

**Code 711 & 712**

*Certificate*  
*in*  
**Computer Applications**



**NATIONAL INSTITUTE OF OPEN SCHOOLING**

(An autonomous organisation under MoE, Govt. of India)

A-24-25, Institutional Area, Sector-62, NOIDA-201309 (U.P.)

Website: [www.nios.ac.in](http://www.nios.ac.in), Toll Free No. 18001809393

---

## ADVISORY COMMITTEE

---

**Chairman**  
NIOS, NOIDA

**Director, Vocational**  
NIOS, NOIDA

---

## CURRICULUM COMMITTEE

---

**Prof. Sunil Kumar Khatri**  
Director  
Amity Institute of Information  
Technology, NOIDA

**Dr. G. Mythili**  
Deputy Director  
IGNOU  
New Delhi

**Dr. Athiyaman**  
Scientist – F & Head  
Ministry of Earth Science  
New Delhi

**Ms. Radhika. B**  
Assistant Director (Vocational)  
NIOS, NOIDA

---

## LESSON WRITERS

---

**Ms. Ramya Sriram**  
Head of the Department  
Computer Science  
Meridian School  
Telangana

**Ms. Vineeta Garg**  
Head of the Department  
Computer Science  
SRDAV Public School  
New Delhi

**Ms. Sangeeta Rawal**  
Head of the Department  
Delhi Public School  
Vasant Kunj  
New Delhi

---

## EDITORS

---

**Prof. Deepti Mehrotra**  
Amity school of Engineering  
and Technology  
Uttar Pradesh

**Mr. Parikshit Pushkarna**  
Vice President  
Accenture Technologies  
Gurgaon

**Dr. Harita Ahuja**  
Assistant Professor  
Acharya Narendra Dev College  
University of Delhi

**Mr. Arun Kumar Dubey**  
Assistant Professor  
Bharati Vidyapeeth college of  
Engineering, New Delhi

**Ms. Mohini Arora**  
HOD (Computer Science)  
AirForce Golden Jubilee Institute  
New Delhi

**Ms. Purvi Kumar**  
HOD (Computer Science)  
Ganga International School  
New Delhi

**Ms. Anju Gupta**  
Former HOD Computer Science  
Rukhmini Devi Public School  
New Delhi

**Ms. Chetna Khanna**  
Former PGT Computer Science  
Shadley Public School  
New Delhi

**Ms. Shilpreet Kaur**  
PGT Computer Science  
National Victor Public School  
New Delhi

**Ms. Radhika. B**  
Assistant Director (Vocational)  
NIOS, NOIDA

---

## COURSE COORDINATOR

---

**Ms. Radhika B**  
Assistant Director (Vocational)  
NIOS, NOIDA

---

## GRAPHICS/DTP

---

M/S Sri Krishna Graphics  
Delhi

---

## *A Word With You*

Dear Learner,

Congratulations for having chosen the course on **Certificate in Computer Applications**.

In an Information Age having Computer knowledge is basic requirement not only to stay connected but to solve the problems efficiently. Nowadays all the services are delivered at our doorsteps because of advancement of technologies and emergence of new digital platforms. As a learner, you should acquire and understand technologies to get the benefit of these services.

The course of Certificate in Computer Applications is specially designed keeping in view of the demand in the IT industry. The curriculum has been mapped with the qualification pack of **Junior Software Developer** at level 4. This curriculum will give you a clear idea about the basic concepts in computing and the programming concepts which you should learn to become programmer. The course contains two modules: The first module will explain about the basic concepts and second module will explicate about Python programming and database concepts.

We have ensured that the course contents are presented in a simple and straightforward manner. We are confident that you will enjoy learning this course through Open and Distance learning system. We have also prepared a practical manual which will help you to do more practice in the programming concepts, database concepts etc. Do more practice for developing the skills which you require to become a developer / Junior programmer etc. We hope you will find this material interesting, informative and useful. We wish you all the very best for a bright and successful future.

**Course Team**  
**National Institute of Open Schooling (NIOS)**

## *How to use the Study Material*

Congratulation! You have accepted the challenge to be a self-learner. NIOS is with you at every step and has developed the material in *Certificate in Computer Applications* with the help of a team of experts, keeping you in mind. A format supporting independent learning has been followed. If you follow the instructions given, then you will be able to get the best out of this material. The relevant icons used in the material will guide you. These icons have been explained below for your convenience.

**Title:** will give a clear indication of the contents within. Do read it.

**Introduction:** This will introduce you to the lesson linking it to the previous one.



**Objectives:** These are statements that explain what you are expected to learn from the lesson. The objectives will also help you to check what you have learnt after you have gone through the lesson. Do read them.



**Notes:** Each page carries empty space in the side margins, for you to write important points or make notes.



**Intext Questions:** Very short answer self check questions are asked after every section, the answers to which are given at the end of the lesson. These will help you to check your progress. Do solve them. Successful completion will allow you to decide whether to proceed further or go back and learn again.



**What You Have Learnt:** This is the summary of the main points of the lesson. It will help in recapitulation and revision. You are welcome to add your own points to it also.



**Terminal Exercise:** These are long and short questions that provide an opportunity to practice for a clear understanding of the whole topic.



**Answers to Intext Questions:** These will help you to know how correctly you have answered the questions.

# CONTENTS

## MODULE–1: Basic Computing Skills (711)

|   |    |
|---|----|
| 1. Computer Concepts and Networking ..... | 1  |
| 2. Computer Software .....                | 18 |
| 3. Data Organization .....                | 32 |
| 4. Securing the System .....              | 45 |
| 5. E-Governance .....                     | 60 |
| 6. Social Networking .....                | 73 |
| 7. Office Productivity Tools .....        | 82 |

## MODULE–2: Computer Applications (712)

|   |     |
|---|-----|
| 8. Introduction to Python .....                   | 115 |
| 9. Control Statements .....                       | 140 |
| 10. Python Lists .....                            | 163 |
| 11. Functions in Python .....                     | 181 |
| 12. Stacks and Queues .....                       | 202 |
| 13. File Handling in Python .....                 | 219 |
| 14. Errors and Exceptions .....                   | 236 |
| 15. Database Management System .....              | 254 |
| 16. MySQL .....                                   | 266 |
| 17. Workplace Practices and Work Management ..... | 296 |



## **MODULE–1: Basic Computing Skills (711)**

1. Computer Concepts and Networking
2. Computer Software
3. Data Organization
4. Securing the System
5. E-Governance
6. Social Networking
7. Office Productivity Tools





**1****COMPUTER CONCEPTS AND  
NETWORKING**

---

**1.1 INTRODUCTION**

---

Computers play a significant role in our day-to-day life and you must be familiar with computers. Have you ever thought as to how a computer would have looked like when it was first invented or how it has evolved over the years? In this lesson, you will see how computers have transformed over a period of time. The term ‘**Computer**’ was derived from the word ‘**Compute**’ which means to calculate. But the usage of computers is not just limited to calculations. It performs many functions at a very high speed and makes our work easier. In this lesson, you will learn about the major milestones in the evolution of computers.

**OBJECTIVES**

After reading this lesson, you will be able to:

- classify the different generations of computers;
- explain data communication;
- identify the networking topology and
- discuss the basics of Internet.

---

**1.2 GENERATIONS OF COMPUTER**

---

We all know that a computer is an electronic device, which transforms data into meaningful information. Data can be anything such as names of the persons, their

# MODULE-1

Basic Computing  
Skills (711)



Notes

age, height, weight, educational qualifications, salary, increments, taxes paid etc. From this data, the computer can provide meaningful information. For example, you can ask the computer to provide a list of all the persons in the age group of 20 to 30 years with a salary above Rs. 30000. A computer can accept the data, store data, process data as desired, retrieve the stored data, as and when required, and print the result in the desired format.

Let us discuss about the different generations of computer.

The evolution of computers can be classified into five generations. Each generation of computer is characterized by the advancement of technology used to build the computer as compared to the previous generations. Whenever there is an advancement in the internal structure of the computer, then we can say that a new generation of computers has evolved, resulting in computer systems to work in a much efficient manner than before.

Have you ever thought about how computers looked when they were first invented? What was their size, how was their performance, what was the hardware parts used? In order to get answers for these questions, you need to understand the evolution of computers. The evolution of computers is classified into five generations. You will learn about each generation of computer in detail now.

## First Generation Computer (1940-1956)

The first generation of computers was built by using vacuum tubes (refer Fig. 1.1). However, this generation of computers was big in size and occupied a lot of space.



Fig. 1.1: Vacuum Tube

Characteristics of the first generation computers:

- Unlike today's computers, these computers were huge in size and build.



- The computers were very expensive so only very few organizations like scientific research institutes could afford it.
- This generation of computers consumed a lot of electricity.
- The performance of these computers was very slow.
- They generated a lot of heat so these computers had to be kept in air-conditioned rooms.
- They required high maintenance.

### Second Generation Computers (1956-1963)

Since the first generation of computers had a lot of issues, the next generation of computers were invented which replaced the vacuum tubes with transistors. A transistor (refer Fig. 1.2) is a small device made up of semiconductor materials such as Germanium and Silicon. The transistors were highly reliable and easier to handle than vacuum tubes.



Fig. 1.2: Transistors

Characteristics of the second generation:

- These computers were smaller in size than the first generation of computers.
- They required less electricity consumption.
- Their performance was comparatively faster than the first-generation computers.
- Inputs to these computers were given using punch cards (punch cards are paper cards in which holes may be punched by hand or machine to represent data and instructions).



## Third Generation Computer (1964-1971)

The third generation of computers emerged due to the development of '**Integrated Circuits**' (IC) (refer Fig. 1.3). Many computer programming languages such as COBOL, Pascal, ForTran gained widespread use during this generation.

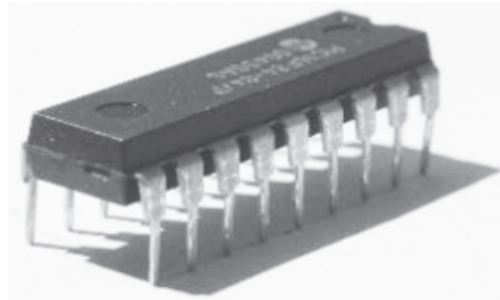


Fig. 1.3: Integrated Circuits

Characteristics of the third generation:

- These computers were more reliable.
- The size of these computers was much smaller than the previous generation of computers.
- They generated less heat.
- They required less maintenance.
- This generation of computers consumed less electricity as compared to the first generation computers.
- From this generation of computers onwards, the keyboard and monitor were used for input and output respectively.

## Fourth Generation Computer (1971-1980)

The period of fourth generation computers started when **Very Large-Scale Integration (VLSI)** circuits were used. The VLSI circuits had about 5000 transistors and other circuit elements with their associated circuits on a single chip which made it possible to have microcomputers in fourth generation. These devices are called microprocessors.

Characteristics of the fourth generation:

- This generation of computers was less expensive.
- They were small in size so they could be carried from one place to another easily.
- They were easily affordable by anyone.
- Mouse and GUI made these computers more user-friendly.

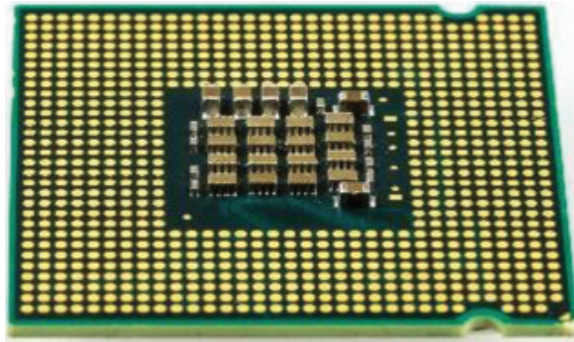


Fig. 1.4: Microprocessor

### Fifth Generation of Computers (1980-till date)

In the fifth generation, **Ultra Large-Scale Integration (ULSI)** technology resulted in the production of microprocessor chips having millions of electronic components embedded on it. This generation of computers can solve highly complex problems including decision making and logical reasoning.

Characteristics of the fifth generation:

- They are good in parallel computing.
- High-level computer programming languages are used.
- This generation of computers leads to Artificial Intelligence (AI) development.
- They work very efficiently with high performance and speed.

Now that you have learnt about the generation of computers, you would have noticed how new technology in every generation brought changes in the evolution of computers. With every generation, the performance and efficiency of the computers were also improved.



### INTEXT QUESTIONS 1.1

Fill in the blanks.

1. Programming languages such as COBOL, Pascal, For Tran gained widespread use during the ..... generation of computers.
2. The period of fourth generation computers used ..... circuits.
3. .... unit was used during the fourth generation of computers.
4. The large number of ..... was used in first generation computers which generated a lot of heat.



Notes

## 1.3 DATA COMMUNICATION

Though the computers were initially invented for calculations and computing, their current usage is not restricted to these purposes. Think of the ways you use computers in your day-to-day life. You can definitely think of using computers beyond calculation purpose. You use it for learning, playing, communicating, sharing and many more.

The rapid evolution of computers has generated the need to share data from one computer to another so that the data can be used in different locations by different people. To share the data between computers, a medium/channel is required. The medium which is used to send or receive data between computers is called '**communication medium**'.

The data communication process has five main components which include message, sender, receiver, medium, encoder and decoder (refer Fig. 1.5).

- **Message** is the data or information to be communicated. The message can consist of text, numbers, images, audio or video.
- **Sender** is a device that sends the message. It is also called as '**transmitter**'. Generally, a computer is used as a sender.
- **Receiver** is a device that receives messages. The receiver can be a computer, printer or another computer related device.
- **Medium** is the physical pathway which connects the sender and receiver. It is used to transmit data. It is also called as '**communication channel**'.
- The '**encoder**' is a device that converts digital signals in a form that can pass through a transmission medium. The '**decoder**' is a device that converts the encoded signals into digital form. The sender and receiver can communicate successfully with the help of encoder and decoder.

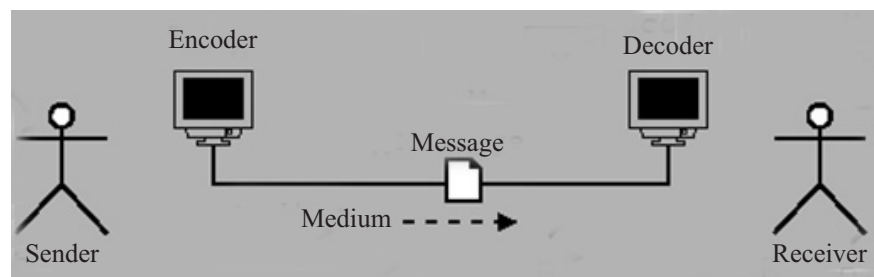


Fig. 1.5: Data Communication



Notes

In order to attain effective data communication, three fundamental characteristics are required. Those are

- **Delivery:** Sending the data to correct destination computer.
- **Accuracy:** Delivering the data accurately. If the data is not delivered accurately, it becomes worthless.
- **Timeliness:** Delivering the data within a certain time interval.

Let us discuss communication media now.

### 1.4 COMMUNICATION MEDIA

A medium is required to transfer the data between computers. The communication media refers to the means of delivering and receiving data or information (refer Fig. 1.6). It is classified into two categories.

- **Wired/Guided media**
- **Unguided/Wireless media**

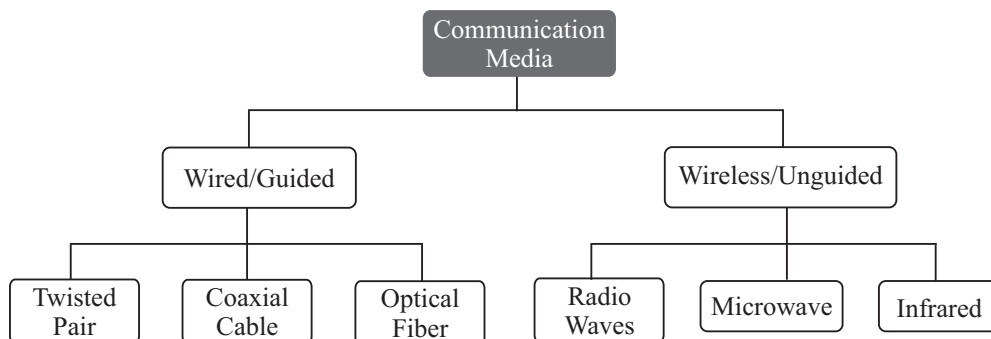


Fig. 1.6: Communication Media

Let us learn about the two categories of Communication media in detail.

#### Wired/Guided Media

The wired/guided media uses physical cables to transfer data between different devices and computer systems. There are three types of wired media like twisted pair, Coaxial cable and fiber optic cable. Let us learn about all of these.

**Twisted Pair:** It is the most widely used medium for telecommunication (refer Fig. 1.7). Twisted-pair cabling consists of copper wires that are twisted into pairs. The use of two wires twisted together helps to reduce crosstalk.

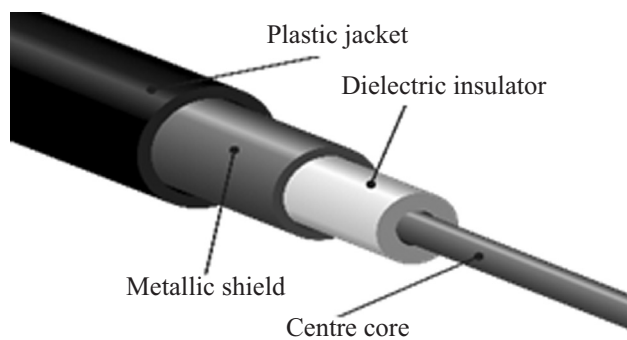


Notes



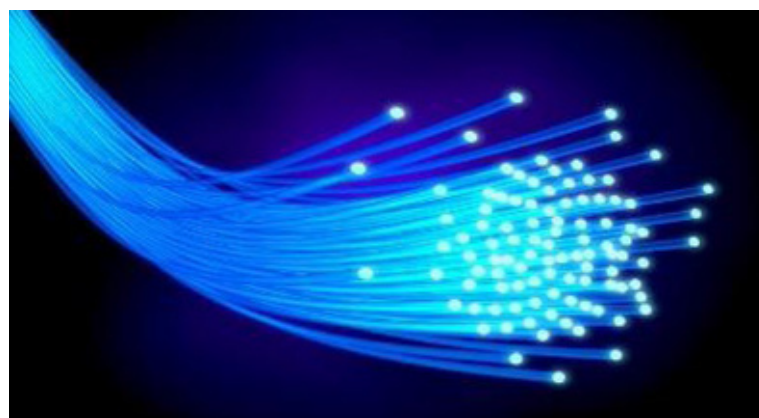
**Fig. 1.7: Twisted Wire**

**Coaxial Cable:** It is a type of copper cable that is specially built with a metal shield and other components to block signal interference (refer Fig. 1.8). It is primarily used by cable television companies to connect their satellite antenna to customer homes and businesses.



**Fig. 1.8: Coaxial Cable**

**Fiber Optic Cable:** It is a high-speed data transmission medium. It contains tiny glass or plastic filaments that carry light beams (refer Fig. 1.9). Digital cable television is delivered through a network of high-speed fiber optic cables.



**Fig. 1.9: Fiber Optic Cable**





Notes

### Wireless/Unguided Media

The wireless/unguided media communicate over the air.

**Radio Wave:** A radio wave is an electromagnetic wave circulated by an antenna. The radio waves have different frequencies and have very long wavelengths. The radio, television, mobile phones use radio wave transmission in one form or the other. Depending upon the frequency, it can travel long or short distances.

**Microwave:** It uses high frequency beams of radio waves to provide high speed wireless connections that can send and receive voice, video, and data.

**Infrared (IR):** Infrared is used for communication over short ranges. It is inexpensive and easy to use a wireless communication medium. IR light is very similar to visible light, except that it has a slightly longer wavelength.



### INTEXT QUESTIONS 1.2

Fill in the blanks.

1. The ..... is a device that converts digital signals in a form that can pass through a transmission medium.
2. .... media uses physical cables to transfer data between different devices and computer systems.
3. ...., ....., ..... uses radio wave transmission in one form or other.
4. Cable TV operators use ..... to connect the television with their satellite antenna.

## 1.5 NETWORK TOPOLOGIES

Before discussing network topology, let us understand the term 'Network'. Networking means grouping of computers and other devices that are linked together for the purpose of sharing data and hardware resources. The network topology defines the way the computers are placed and how they are interconnected with each other in a network. It also describes how the data is transferred among these nodes (Each computer/device in a network is called a node). The topology can be physical or logical. The physical topology refers to how the devices are interconnected using wires and cables whereas the logical topology refers to how the devices in the network communicate among each other.



The network topologies are categorized into the following types:

- Point-to-Point topology
- Bus topology
- Star topology
- Ring topology
- Tree topology
- Mesh topology

Let us discuss each one in detail.

### Point-to-Point

It is the simplest form of network structure where two nodes are directly connected to each other (refer Fig. 1.10). This topology is suitable for small areas and it provides a faster and a more reliable connection.

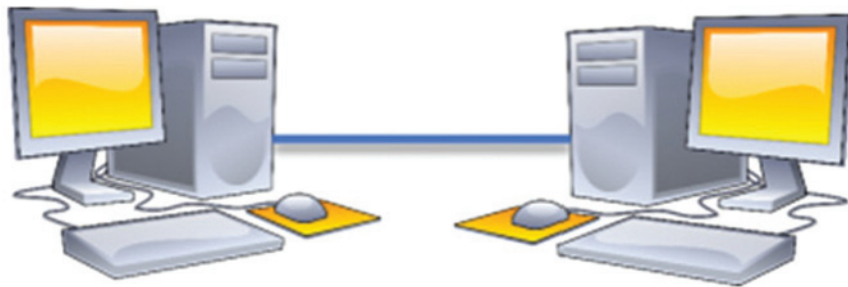


Fig. 1.10: Point to Point Topology

### Bus Topology

This topology uses a single communication line or one main cable in which all the other nodes are connected (refer Fig. 1.11). The main cable acts as a backbone for the network. In this topology, one of the computers acts as a server (a computer or a program that manages the access to a centralized resource in a network) and other computers act as a client (a computer which is capable of obtaining information and applications from a server). The server provides information to all clients in the network. This topology is used in small networks where cable requirement is relatively small. The disadvantage of this topology is that, if there is a malfunction in the main cable, the entire network goes down.

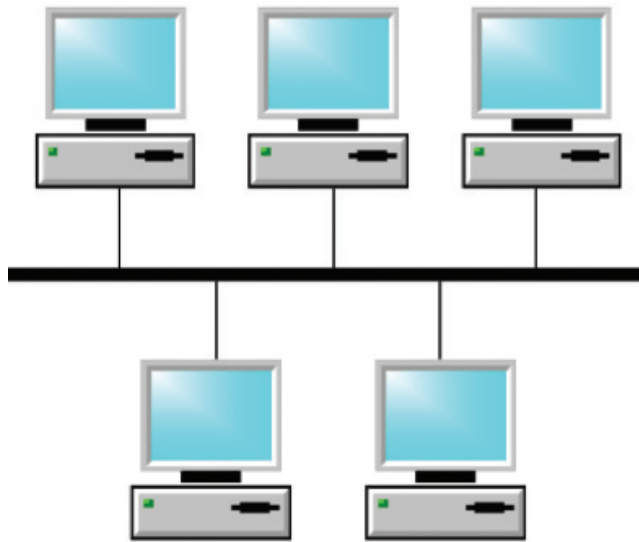


Fig. 1.11: Bus Topology

### Star Topology

In this topology, each computer in the network is connected to a central computer using a point-to-point connection (refer Fig. 1.12). The central computer is called **'Server'** and other computers in the network may be termed as **'Clients'**. The data is transmitted across the network through the networking device named **'hub'** or **'switch'**. The star topology is popular because of simple configuration and low cost. If any one of the computers in the network fails, the other computers continue to do their job but if the hub itself fails, the entire network becomes unusable.

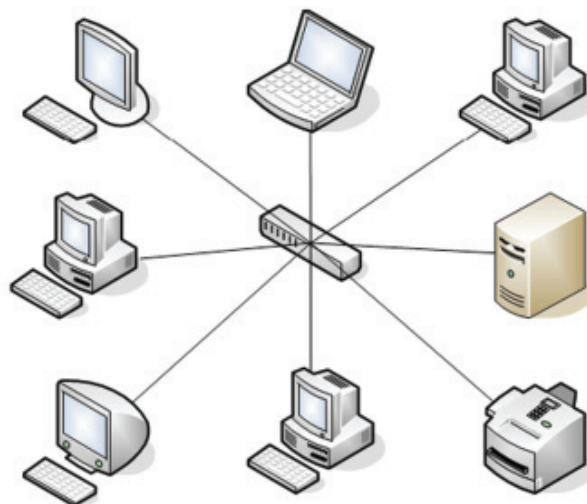


Fig. 1.12: Star Topology



## Notes

### Ring Topology

In this topology, all the nodes are connected in a circular manner. Each node is connected to exactly two adjacent nodes. In this topology, the data will be transmitted through all the intermediate nodes until it reaches to the destination node. Since the data is transferred through each node, the network gets slower than star topology. The disadvantage of this topology is that, if one node fails, then the entire network gets affected.

### Tree Topology

It is a special type of structure in which many connected nodes/components are arranged like branches of a tree. This topology is used to organize the computers in a corporate network. A tree network is a combination of star networks which are interconnected via a bus network. The expansion of network, error detection and correction are also possible in this topology. If one segment is damaged, other segments are not affected. In this type of topology, if the main bus cable breaks, the whole network is crippled. The disadvantage of this topology is that as more and more nodes and segments are added, the maintenance will become difficult.

### Mesh Topology

In this topology, each node is connected to every other node in the network (refer Fig. 1.13). This topology can handle large volume of traffic and it is widely used in wireless networks. An advantage of this topology is that if any one of the component/nodes fails, the data transfer can happen through other alternative node. The expansion and modifications can be done without affecting the other nodes, but the overall cost is high as compared to other topologies.

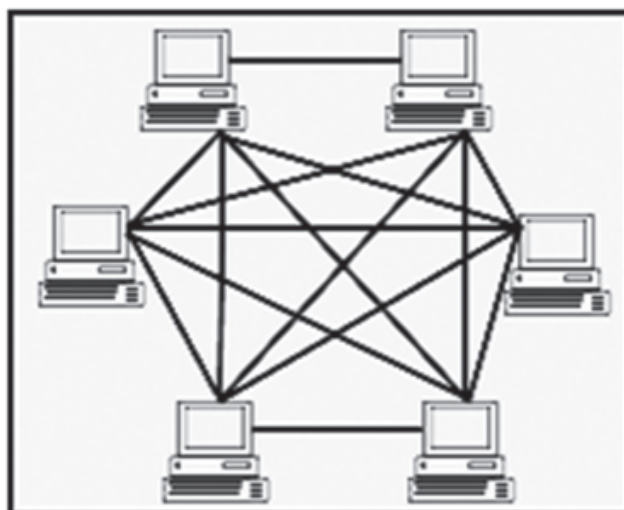


Fig. 1.13: Mesh Topology



## INTEXT QUESTIONS 1.3

Fill in the blanks.

1. Each computer in a network is called .....
2. In ..... topology the nodes are connected in a circular manner.
3. A tree network is a combination of ..... network which are interconnected via ..... network.
4. .... refers to the way information flows between different computers/components.

---

## 1.6 INTERNET

---

Internet is a network of millions of computers connected to each other to share information and resources. It is a global network with millions of private, public, academic, government and business networks. With the help of the Internet, you can do online shopping, get the latest news update, read books and magazines, do online booking of tickets and send e-mails. It also allows you to get connected with people around the world for personal and professional benefits. It is also a major source of entertainment as you can watch movies or play games using a computer. It plays a vital role in each individual's personal and professional life. Internet is widely used by students, scientists, educational institutions, professionals, business organizations for research and general purposes.

Let us now learn about some of the essential terms that are frequently used while talking about the internet.

### Internet Related Terms

**Web Page:** It is a collection of hypertext documents which contains information in the form of text, images, links, video, audio and graphics. HyperText Markup Language (HTML) is used to create web pages. The web pages can be accessed through Internet by using a web browser.

**Website:** It is a collection of related web pages on Internet. There are many websites available on the Internet such as entertainment websites, career websites, commercial websites and educational websites.

**Home Page:** The first page which is displayed when you access a website is called a '**home page**'. It is the main page of a website which provides information about the website.




---

Notes



**Hyperlink:** It is a word or an image in a webpage which is used to navigate to another webpage/website or another section in the same webpage. By clicking on these links, you can navigate from one web page to another. The hyperlink can be identified on a web page as it appears with an underlined text and it is usually in blue colour. When you hover the mouse on the hyperlink, the cursor changes from an arrow to hand-pointing as shown below.



Fig. 1.14: Example for Hyperlinked Text

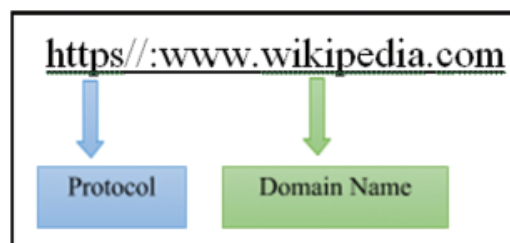
**World Wide Web (WWW):** It is a collection of documents stored on computers around the world. These documents are available on the Internet and contain a wide variety of resources such as text, graphics, video, audio, pictures etc.

**Web Server:** It is a type of computer that stores websites and their related files for viewing on the internet and serves them when they are accessed from anywhere in the world. So, to make a website available on the internet, it has to be uploaded on a server.

**Web Browser:** It is a software used to view web pages. It acts as an interface between the user and the World Wide Web. A browser enables the users to request for web pages from the web server and view the pages. Some of the examples of web browsers are Google Chrome, Mozilla Firefox, Internet Explorer, Opera, and Netscape Navigator.

**Uniform Resource Locator (URL):** URL is the web address of a web page. It is used for identifying a resource on the web. The first part of the URL is called a '**protocol**' (A protocol is a set of rules to be followed for communication between network devices) which indicates what protocol is used and the second part specifies the domain name (A domain contains a group of pages that can be accessed with a common set of rules and procedures) where the document is located. The URL of a web page/website is typed in the address bar of the web browser.

Example of a website URL is *https://www.wikipedia.com*





Notes

**Internet Protocol (IP) Address:** Each resource on the web (computer or a website) on the web is identified by a unique address called ‘IP address’. The IP address is a numeric address which specifies the exact location of a computer on the Web. You can access a computer on the web by using an IP address, such as 198.162.1.2. All devices in the network sharing a common part of the IP address are said to be in the same domain.



**INTEXT QUESTIONS 1.4**

Fill in the blanks.

1. .... is a language used to create web pages.
2. A ..... is a set of rules to be followed for communication between network devices.
3. Each computer on the web is identified by a unique address called ..... address.
4. A ..... is a collection of web pages on the Internet.



**WHAT YOU HAVE LEARNT**

- The first-generation computer is generally characterized by the widely used vacuum tubes.
- The second generation of computers used transistors and the third-generation computers used Integrated Circuits.
- The fourth generation computers used microprocessors and fifth generation computers used ULSI.
- The data communication process has five main components which include message, sender, receiver, medium, encoder and decoder.
- There are different types of medium used for data communication over the network.
- The physical topology is the physical layout of nodes, workstations and cables in the network whereas logical topology is the way information flows between different computers/ components.
- Point to Point topology is the simplest form of topology where two nodes are directly connected to each other and it is suitable for small areas.

## MODULE-1

Basic Computing  
Skills (711)



Notes

### Computer Concepts and Networking

- In Bus topology, one of the computers acts as a server and other computers act as a client.
- In Star topology, each computer in the network is connected to a central computer through a networking device called 'Hub' using a point-to-point connection.
- In Ring topology, each node is connected to exactly two adjacent nodes. Here, the data will be transmitted through all the intermediate nodes until it reaches the destination node.
- The tree topology is used to organize the computers in a corporate network.
- Mesh topology can handle large volume of traffic and it is widely used in wireless networks.
- Internet is a network of millions of computers connected to each other to share information and resources.
- Web page is a collection of hypertext documents and can be accessed on the Internet using a web browser.
- Web Server is a computer that stores websites and their related files for viewing on the Internet.
- Web Browser is a software used to view web pages. It acts as an interface between user and the World Wide Web.
- URL is used for identifying a resource on the web.
- The IP address is a numeric address which specifies the exact location of a computer on the Web.



#### TERMINAL EXERCISE

1. Write any two characteristics of fifth generation computer.
2. What is Infrared used for?
3. Define network.
4. Write one advantage and one disadvantage of Mesh topology.
5. What do you mean by IP address?
6. Write about five main components of data communication.





7. Explain the ring topology with the diagram.
8. Write one sentence about the following Internet terms.
  - (i) Home page      (ii) Hyperlink      (iii) Browser
  - (iv) Web server      (v) WWW



**ANSWERS TO INTEXT QUESTIONS**

**1.1**

- |                   |  |
|-------------------|--|
| 1. Third          | 2. Very Large-Scale Integration (VLSI) |
| 3. Microprocessor | 4. Vacuum Tubes                        |

**1.2**

- |                             |                  |
|-----------------------------|------------------|
| 1. Encoder                  | 2. Wired/guided  |
| 3. Radio, TV, Mobile phones | 4. Coaxial Cable |

**1.3**

- |                   |                     |
|-------------------|---------------------|
| 1. Node           | 2. Ring             |
| 3. Star ..... bus | 4. Logical Topology |

**1.4**

- |                           |             |
|---------------------------|-------------|
| 1. HTML                   | 2. Protocol |
| 3. IP (Internet Protocol) | 4. Website  |

**Key Learning Outcomes**

- Be proficient in choosing the communication media.
- Be able to arrange computers in different topologies (i.e., BUS, STAR, TREE etc.).

## MODULE-1

Basic Computing  
Skills (711)



Notes

# 2

## COMPUTER SOFTWARE

---

### 2.1 INTRODUCTION

---

In the previous lesson, you have learnt about the evolution of computers. The evolution was possible to achieve, because of the advancement in hardware and software. The computer hardware is the collection of physical parts of the system i.e., the devices which you can touch and feel, e.g., keyboard, mouse, printer, monitor etc. The software is nothing but a set of instructions given to the computer to perform tasks, for example: word processing software, presentation software etc. Have you ever tried to identify the hardware and software components in your computer? If not, then you can try now as in this lesson, you will learn about computer software as well as its types and applications in detail.



### OBJECTIVES

After reading this lesson, you will be able to:

- classify software;
- differentiate between system software and application software;
- list the types of operating systems;
- explain application software and its types; and
- use open source software.

---

### 2.2 SOFTWARE

---

As you are aware that, computer cannot do anything on its own. It is the user who instructs the computer, what to do, how to do and when to do. In order to perform any task, you have to give a set of instructions to the computer. These set of



instructions are called 'Programs'. Software refers to a set of programs that makes the hardware perform a particular set of tasks in a particular order. You can broadly classify softwares into two categories, i.e., system software and application software (refer Fig. 2.1). Let us learn about system software and application software in detail.

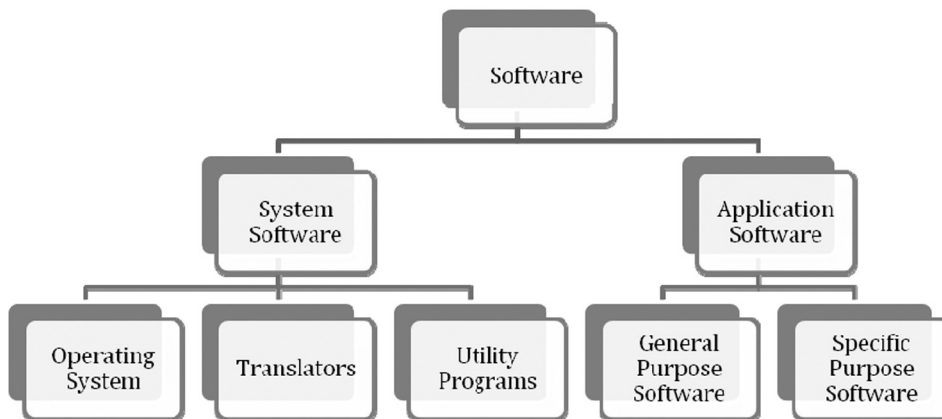


Fig. 2.1: Software and its Classification

### 2.2.1 System Software

When you switch on the computer, the programs stored in Read Only Memory (ROM) are executed which activates different units of your computer and makes it ready for you to work on it. This set of programs can be called system software. This software controls various operations of the computer systems and manages the computer resources. It is responsible for accepting the data from the input devices, transmitting the information to the output devices, conversion of data/instructions into computer understandable language, storing the data and so on.

The system software is further classified into the following three categories:

- (a) Operating System
- (b) Translators
- (c) Utility Programs

#### (a) Operating System

In order to make the computer to perform tasks, you need an operating system. Operating system (OS) is a program, which acts as an interface between the user and the computer hardware. Every computer needs an operating system to run other programs. OS manages computer hardware and software resources and



makes it possible for the user to use the computer system. It is also responsible for program execution, managing input/output operations, memory/storage allocation of files, providing user interface etc.

Now let us discuss about the different types of operating systems.

## Types of Operating Systems

There are different types of operating systems that run on different types of devices. The operating systems are mainly classified into:

- **Multi-tasking/Time Sharing OS:** It enables many people located at different places to share and use single computer resources at the same time. Here, each task is given some time for execution.
- **Real Time OS (RTOS):** This operating system interacts within a specified time and responds to the request almost instantaneously. The time interval required for processing and responding to the given inputs is very less in this operating system. The RTOS are used in robots, air traffic control systems, defense application systems like RADAR etc.
- **Distributed OS:** It uses many processors located in different computers to serve more users. The main goal is to make the users access and share the computer resources available in other computers.
- **Multi-Processing OS:** This operating system supports the execution of multiple processes at the same time. It uses multiple numbers of CPUs. It is expensive but the processing speed is faster. Examples are Linux, Unix.

Operating systems can also be classified on the basis of types of interfaces used. Those are as follows:

### Command Line Interface (CLI)

It is a text-based interface, used to view and manage the computer files. In this interface, user types the commands to operate the software or hardware of the computer and receives the reply in the same way. Here the user has to remember the commands. Some of the examples of OS with CLI are linux, unix etc.

### Graphical User Interface (GUI)

In GUI, the user interacts with the computer by pointing or touching the visual interfaces. The user can use the mouse to point or click the objects on the screen. It uses icons, menus to carry out the commands for opening, deleting files etc. Here the user does not have to remember/write any command for doing the tasks. Some

of the examples of OS with GUI are Windows 7, Windows 8, Windows 10, Android etc. These operating systems provide user friendly environment.

Fig. 2.2 depicts the difference between a CLI and GUI.

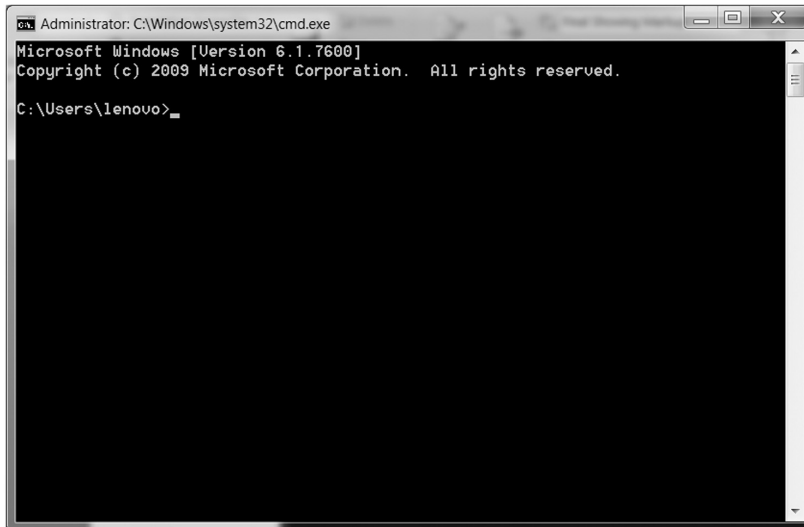


Fig. 2.2: Command Line Interface vs Graphical User Interface



Notes

### (b) Translators

We write computer programs in English language, which is easy for others to understand. These programming languages are called **High Level Language (HLL)**. But computers cannot understand the high level language. Do you know what computer can understand? It can only understand machine language that consists of 0s and 1s. So, we need a program that converts the high level language into machine language. There are system programs called translators/language processors, which perform the job of converting high level language into machine understandable form.

There are three different types of translators available. Let us discuss them one by one.

- **Assembler:** It converts the program written in assembly language (i.e. low level programming language that consists of instructions called mnemonics) into machine language.
- **Interpreter:** It converts high level language into machine language. It translates the program statements line by line. If there is any error in any line of the program, the program execution will be stopped until the error is rectified.



- **Compiler:** It also converts the high level language program into machine language, but the conversion method is different from that of an interpreter. The compiler converts the entire program in one shot and reports all the errors of the program with line numbers. The compiler is faster than interpreter, but compiled programs take more memory.

## Utility Programs

Utility programs are designed to analyze, configure, optimize or maintain a computer. These programs ensure the smooth functioning of a computer. It also helps to backup the data, remove unused files, recover the data that are accidentally deleted, and prevent the virus entering into our computer. In simple words, these programs perform housekeeping functions and assist our computer for better use. These programs are also known as ‘**service programs**’. Some of the utility programs are disk defragmenters, network managers, application launchers, file managers, disk cleaners, disk repair, etc. You try to identify some other utility programs that are available in your computer. Generally, utility programs are divided into:

- **Disk/Folder/File Management Programs:** These programs are also called as ‘**File Management Software**’ or ‘**File Manager**’. It is used to organize and keep track of the files. Also, it helps you to create new files/folders, edit an existing file, save a file and organize the files in a folder. The Windows Explorer is the default file management software in MS Windows. Some of the examples of other file management software are Explorer++, Pathfinder, Free Commander etc.
- **Antivirus Software:** This utility program prevents the virus to enter into your computer. A computer virus is a malicious program, which is created to destroy the data stored in the computer. It can make our computer system behave abnormally. Antivirus software scans the computer hard disk and removes the virus from the system. Also, it detects the virus as soon as the viruses enter our system and disables it right away. It is important to install antivirus software in your computer and frequently scan the computer for any viruses. If any of the viruses are detected, then remove them immediately. Some of the examples of antivirus software are Norton Antivirus, McAfee Antivirus, Kaspersky Antivirus, Quick heal antivirus etc.
- **Encryption/Decryption Software:** Encryption refers to the process of encoding data or file(s). It uses algorithms for encrypting the data. The purpose of encryption is to protect the data, when it is transmitted through the internet. Decryption refers to the process of converting the encrypted data to its original form using algorithms. An authorized user can decode/decipher

**Notes**

the encrypted data into a readable form. The various software products provide the facility for encrypting and decrypting the data. Few examples are BitLocker, LastPass and Symantec endpoint encryption.

Till now, you have learnt about the system software which refers to the programs that coordinate the internal operations of the computer. On the other hand, the application software refers to the programs, which assist a user to carry out a specific task. But application software cannot run without the presence of system software. Let us learn about the application software in detail.

**2.2.2 Application Software**

It is a program, or a set of programs designed for the need of the end users. The application software includes, the applications for medical and financial accounting, inventory control, word processing, handling calculation and figures, reservations, billing etc. You might have used some of the application software in your day-to-day life. Open Office Writer is an example of application software.

Application software can be further classified as:

- General purpose software.
- Specific purpose software.

Let us discuss about them in detail.

**General Purpose Software**

These are user friendly software written to cater to the general needs of a user, such as preparing documents, drawing pictures, database to manage data/information, playing games etc. Such general-purpose software is also called a '**package**'. Some of the general-purpose software are:

- Word processing software.
- Spreadsheets.
- Presentation software.
- Database management systems.
- Graphics Software.

Let us discuss about these software in brief.

**(a) Word Processing Software**

Nowadays, processing and organizing a textual content is the most common task done by the users. This is possible with the help of a word processor. Do you know

# MODULE-1

Basic Computing Skills (711)



Notes

what a Word Processor is? Let's understand it first. Simply you can say that a word processor processes words. It is an application program that allows users to compose, edit, format and print the documents. It is also used to create resume, worksheet, circular, report, book etc. Some of the commonly used word processors are AbiWord, Google Docs, OpenOffice Writer. The Fig. 2.3 shows the document window of OpenOffice Writer.

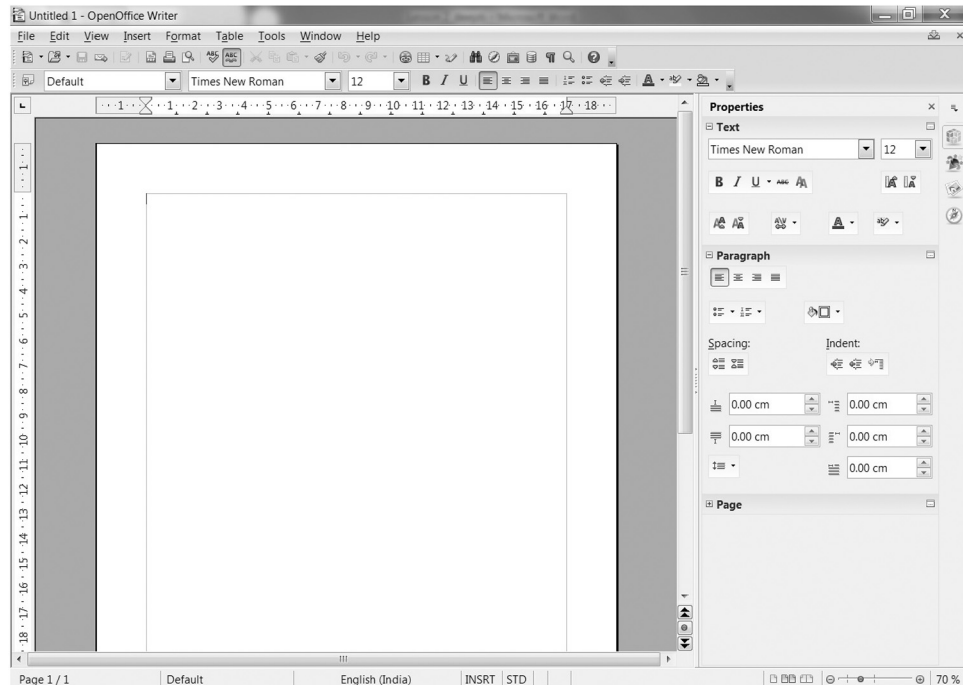


Fig. 2.3: Word Processing Software Screenshot

## (b) Spreadsheets

It is an application software for organizing, analyzing and storing the data in a tabular form (refer Fig. 2.4). The data entered in this application program can be manipulated, calculated and analyzed in the desired manner. This software operates on the data entered in the cells of a table, rows and columns. Do you know that by using spreadsheets, calculations can be made in a much easier way than any other data processing applications? Yes. It is. You can see this vividly when you calculate the average/grades of the students, billing, receipts, etc., on your own through this software. Some of the popular spreadsheet programs are Microsoft Excel, Open Office Calc, Google Sheets. The Fig. 2.4 depicts Open Office Calc worksheet.





Notes

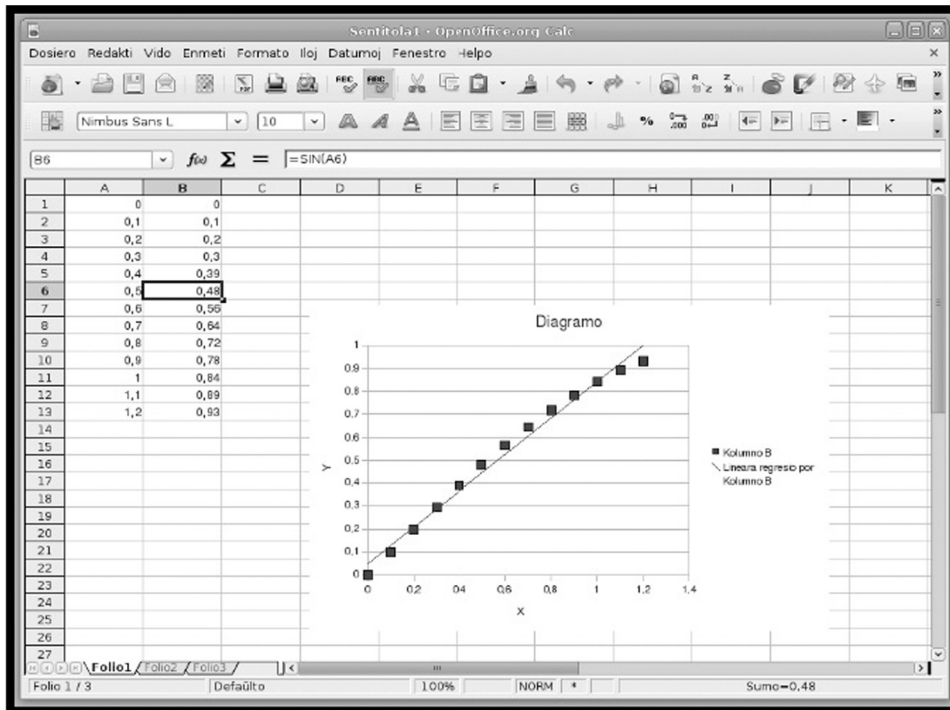


Fig. 2.4: Spreadsheet Screenshot

**(c) Presentation Software**

The presentation software is an application software used to present the information, normally in the form of a slide show. These programs are used for creating slide shows that display information in an attractive way. The Presentation software is broadly divided into:

- (i) Business presentation software;
- (ii) Multimedia authoring software.

Business presentation software enables you to include texts, images and multimedia authoring software allows to include text, images, animations, audio and video etc. Do you know that video and audio included in a presentation attract the audience? Yes, it's the fact that multimedia presentation normally captures the attention of the audience more than the normal presentation. Some of the popular examples of the presentation software are Microsoft PowerPoint, OpenOffice Impress, GoogleSlides etc.

# MODULE-1

## Basic Computing Skills (711)



Notes

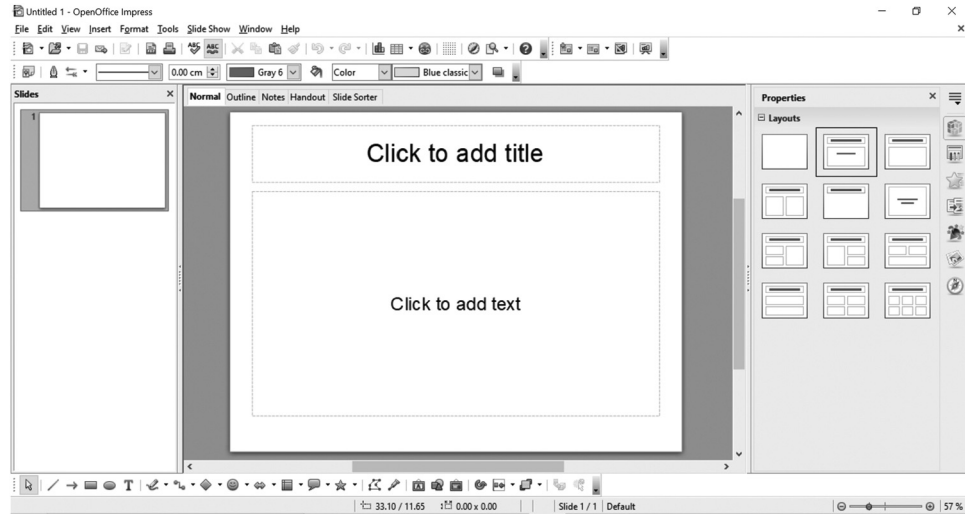


Fig. 2.5: Open Office Impress Screenshot

### (d) Database Management System Software (DBMS)

DBMS is a set of programs, which creates and manages a database (collection of logically related data items). It provides the facilities of storing, organizing, retrieving, and updating the data. It accepts large volumes of data with multiple concurrent users. You will be learning about the lesson on DBMS in module 2. Some of the examples of DBMS include Oracle, MySQL, OpenOffice Base etc. OpenOffice Base screenshot can be referred at Fig. 2.6.

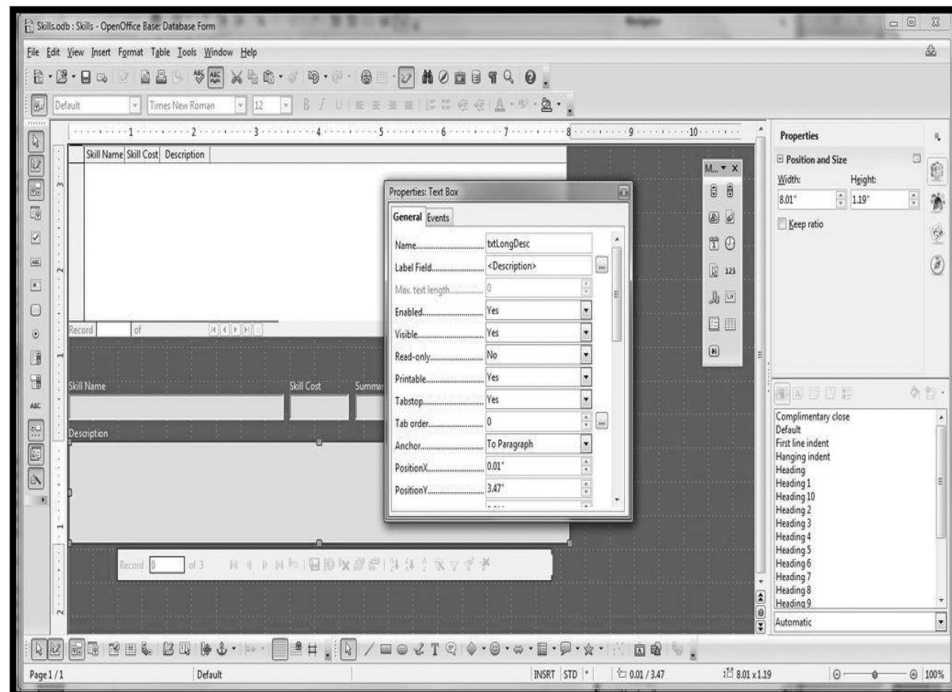


Fig. 2.6: OpenOffice Base Screenshot



Notes

### (e) Graphics Software

A graphics software is an application software that can be used to create and manipulate images on a computer. The computer graphics can be cliparts, logos, headings, backgrounds, digital photos, etc. You may have come across a situation in your daily life when you may want to change any picture. The graphics software enables you to edit or draw pictures, then store them in files, embed them in documents, and finally print them. Some of the examples of popular graphics software are Adobe Illustrator, CorelDRAW and GIMP etc. In Fig. 2.7, you can see the GIMP software screenshot.

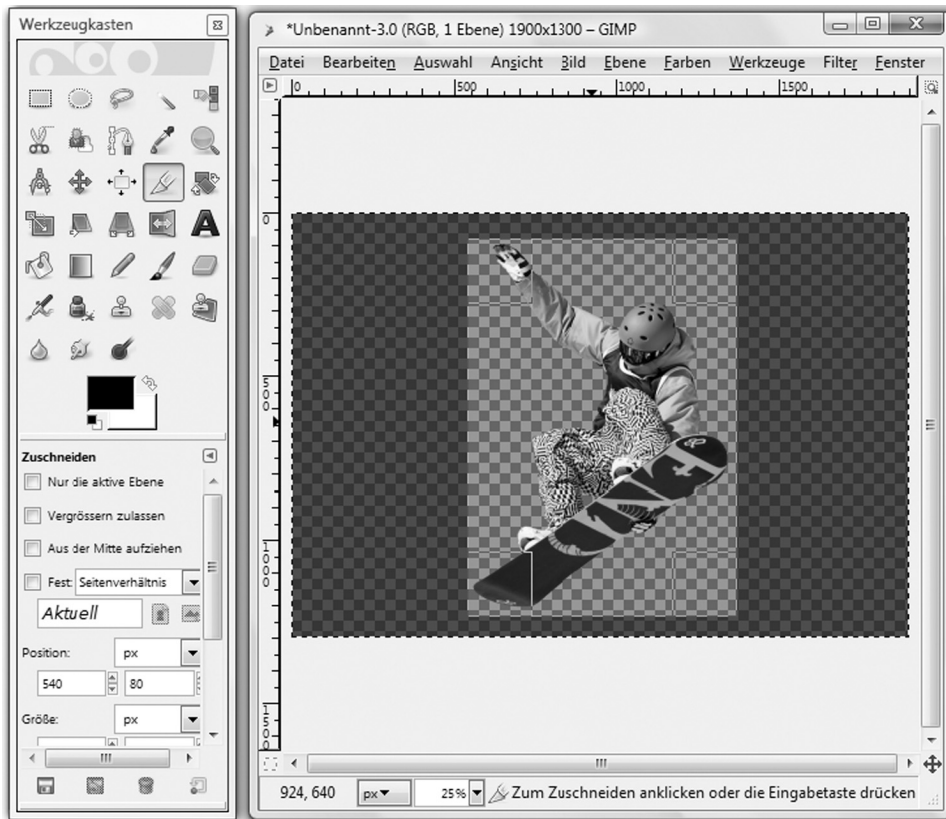


Fig. 2.7: GIMP Screenshot

### Specific Purpose Software

These are the applications that are customized to meet specific requirements of an organization/institution/individual. This software can do only a specific task. For example, student information detail contain only the students details. It may not have any other information except related to the specific task i.e., details of the students. In the same way, payroll package software will have only the payroll information. You can think of some more examples of specific purpose application software.



Notes



## INTEXT QUESTIONS 2.1

Fill in the blanks.

1. .... is the default file management software in MSWindows.
2. .... converts the entire high level language program in one shot and reports all the errors of the program with line numbers.
3. .... is a term used to refer to encoded data so as to make the data unreadable and unusable to unauthorized users.
4. A ..... is a software that handles the storing, retrieving and updating of data.
5. A ..... is an application software that can be used to create and manipulate images on a computer.
6. .... OS interacts with in a specified time and responds to the user almost instantaneously.
7. In ..... Interface, the user types in the commands to operate the software or hardware of the computer.

### 2.2.3 Open Source Software

In the previous section of this lesson, you have learnt about system software and application software in detail. Most software, which are installed in your computer might be proprietary software. Proprietary software is software which is distributed under license agreement. Did you hear about the license free software? The license free software is a free software, which can be downloaded and used by all. You can simply search the name of the software in Internet and download it. Thereafter, you can install it in your device and start using it.

The open source software is available with its source code. Source code is the collection of statements written by a programmer in some programming language. The source code of the software can be inspected, modified and enhanced by anyone. It is released under a license in which the copyright holder grants the rights to the user to use, modify and distribute the software to everyone. Some examples of open source software are Linux, Ubuntu, Apache, Mozilla Firefox and GIMP etc.



Notes

Open source software licenses promote collaboration and sharing of knowledge because it permits others to make modifications of the source code and incorporate those changes into their own projects. It encourages computer programmers to access, view and modify this software, whenever they are ready to do it.

**Open Educational Resources (OER)**

OERs are free and openly licensed educational material that can be used for teaching, learning, assessing, research and other purposes. It's an open platform where everyone can share, use and learn. OER often have a creative commons license that states how the materials can be used, reused, adapted and shared. It saves the cost for the learners and also helps them in learning.

Some of the advantages of OERs are:

- It is useful in improving the education across the globe.
- It offers free access to some of the world's best resources.
- It offers equal access to knowledge and re-usability.
- It helps in enhancement of content knowledge.
- It is accessible and affordable for all.
- It is adaptable and allows others to modify them for their use.

Open source licenses affect the way people can use, study, modify, and distribute the software. This license software stipulates that anyone who releases a modified/ updated open source program must also release the source code for that program alongside it.



**INTEXT QUESTIONS 2.2**

Fill in the blanks.

1. .... is a Linux operating system developed to increase usability and ease of use.
2. Full form of OER is .....
3. .... is a part of the Open-Source software, which allows the computer programmers to analyze how the software works.

# MODULE-1

Basic Computing  
Skills (711)



Notes



## WHAT YOU HAVE LEARNT

- Software refers to a set of programs that makes the hardware perform a particular set of tasks in a particular order.
- Software can be classified into System Software and Application Software.
- System software controls the various operations of the computer systems and manages the computer resources.
- The system software is further classified into operating system, translators and utility programs.
- Operating system (OS) is a program which acts as an interface between the user and the computer hardware.
- Translators/language processors convert the high level language into machine understandable form.
- Utility programs/service programs are designed to analyze, configure, optimize or maintain a computer.
- Application software can be classified into General purpose application software and Specific purpose application software.
- A word processor is an application software used for composing, editing, formatting saving and printing documents.
- Spreadsheet software allows one to store, calculate, manipulate and analyze the data.
- A presentation is a combination of multiple slides.
- Some of the examples of open source software are Linux, Ubuntu, Apache, Mozilla Firefox and GIMP etc.
- OERs are free and openly licensed educational material that can be used for teaching, learning, assessing, research and other purposes.



## TERMINAL EXERCISES

1. Differentiate between encryption and decryption.
2. What is presentation software? Give examples.



3. What is the importance of OER?
4. Classify the operating system based on the interface used.
5. Explain operating system.
6. Describe about the types and role of translators.
7. What do you mean by DBMS?
8. What is the purpose of an operating system?
9. Explain the different types of operating systems.



**ANSWERS TO INTEXT QUESTIONS**

**2.1**

1. Windows Explorer
2. Compilers
3. Ciphertext
4. Database Management System (DBMS)
5. Graphics package
6. Real Time
7. CommandLine

**2.2**

1. Ubuntu
2. Open Educational Resources
3. Source code

**Key Learning Outcomes**

- Be proficient in identifying the system and application software.
- Be able to use the open source software and Open Educational Resources effectively.

## MODULE-1

Basic Computing  
Skills (711)



Notes

# 3

## DATA ORGANIZATION

---

### 3.1 INTRODUCTION

---

In the beginning of the academic year, you must have shared your personal details such as your name, admission number, date-of-birth, parent's name, residential address, phone number, passport size photograph etc., with your instructor. The details provided by each student to their instructor/teacher is data. This clearly indicates that the facts and figures of the concern entities (in this case the student) are called **data**. Data can be in the form of text, numbers, dates, symbols, images, audios, videos etc. For storing and managing the details provided by the students in computer, we need a database. Database is a collection of data stored in an organised form. Can you think of an example where data and database is used in real life? Yes, you are right when you give an example like student's data, customer's data will be managed by cellular companies in customer's database. Another similar examples can also be explored and put forward. In this lesson, you will learn about data, information, data warehousing, data mining and big data in detail.



#### OBJECTIVES

After reading this lesson, you will be able to:

- differentiate between data and information;
- analyse the data and learn the stages of data analysis;
- discuss data warehousing and data mining; and
- explain big data.





Notes

### 3.2 DATA AND INFORMATION

In lesson-1, you have already learnt that data can be anything such as names of the persons, their age, height, weight, educational qualifications, salary, increments, taxes paid etc., and from this data, the computer can provide meaningful information about that person. At this point you may be thinking that **data** and **information** are interchangeable and mean the same thing. But there is a difference between data and information. Data is simply some facts or figures or bits of information, but not the information itself. When data is processed, interpreted, organized, structured or presented then it becomes meaningful or useful. This is called **information**.

In the classroom scenario, after collecting the details from all the students, your instructor/teacher might store all the students' data in table format in a notebook (as shown at Table 3.1), which then becomes meaningful information of the entire class.

**Table 3.1: Student's Details**

| Name    | Admission No | Date of Birth | Father's Name | Address               |
|---------|--------------|---------------|---------------|-----------------------|
| Ravi    | 1001         | 12/12/2004    | Shankar       | 103, Vertex,<br>Delhi |
| Dheepa  | 1085         | 22/01/2005    | Vinoth        | 21, Prajay,<br>Delhi  |
| Lalitha | 1264         | 05/05/2004    | Sundar        | 12/B, Vista,<br>Delhi |

The table 3.2 will give a clear picture about the key differences between the data and information.

**Table 3.2: Differences Between Data and Information**

| Data   | Information  |
|--|--|
| Data means raw facts gathered about someone or something, which is random. | Facts, concerning a particular event or subject, which are refined after processing is called information. |
| It can be text and numbers.  | It is refined data.  |
| It is in unorganized form.   | It is in organized form.   |
| The data may or may not be useful.   | The information is always useful.  |
| It does not depend on information.   | The information cannot be generated, without data.   |



When the amount of data is small, it is easier to organize and find meaning/discover information from it. But when a large amount of data is collected, first you need to analyse the data to find out/discover information from it as per your requirement. Thus, data analysis becomes an important process to generate useful information. Now let us discuss data analysis and the various stages involved in data analysis.

### 3.3 DATA ANALYSIS

It is a process of inspecting, transforming, and displaying the data with the goal of finding out/discovering some useful information. This information will help the persons who are involved in the decision-making process. First, the data from various sources is gathered/collected, reviewed, and then analysed so as to come to some sort of a conclusion. As you know that, the data is merely facts and figures, Data analysis processes, organises, interprets, structures and presents data into useful information that provides context for the data.

The software products are often used to perform an efficient and optimum data analysis. Industries/companies use this analysis in areas such as strategic management, marketing and sales, business development and maintenance of records of human resources etc.

There are various stages involved in data analysis. Let us discuss each stage in detail.

#### (a) Data Collection

Data is the necessary input for analysis. It depends on what to analyse and how to measure it. The data can be collected from a variety of sources such as traffic cameras, satellites, recording devices, interviews, downloads from online sources etc. For example, the instructor/teacher collected the data from all the students. Here the instructor manually collected the data from the students. Once the data is collected, then the data needs to be processed, which is known as data processing.

#### (b) Data Processing

The data which is collected needs to be processed or organised for analysis. This may involve placing the data into rows and columns in a tabular format. You can use spreadsheet software for placing the data and for further analysis. For example, the instructor/teacher entered the student's data in the spreadsheet. Now by using this software he/she may organise the given data to obtain meaningful information. Once the data is organised, it goes to the data cleaning stage.



---

**Notes****(c) Data Cleaning**

Now at this stage, the data collected may not be useful for the analysis. Hence, it needs to be cleaned. The data may contain errors, duplicate records etc. For example, in the student's details, the data which was entered may contain some errors or inaccuracies; there might be some duplicate data. Hence, it needs to be cleaned to get error free data. Data cleaning is the process of preventing and correcting these errors. The cleaning tasks which include record matching, identifying inaccuracy of the data, evaluating overall quality of the existing data or eliminating duplication of the data. This can also be carried out through a variety of analytical techniques. You need to think about how to clean the data. There are several types of data cleaning techniques depending on the type of data. For example, for textual data, spell checkers can be used to lessen the amount of mistyped or wrongly spelled words. After cleaning, the data goes to the Exploratory Data Analysis stage.

**(d) Exploratory Data Analysis**

Once the data is cleaned, it has to be analysed. Data analysts may apply a variety of techniques referred to as **Exploratory Data Analysis (EDA)** to understand the data. **EDA** is a systematic way of approach to analyse the data. The process of exploration may result in additional data cleaning or requests for getting additional data, so these activities may be iterative in nature. For example, in the student's details, getting additional information like phone number, email id or entering the marks scored by the student after every exam would be useful in future.

These techniques are used to encourage the data to suggest models that might be appropriate. The purpose of EDA is to:

- check the missing data and other errors;
- identify the most influential variables, etc.,

It focuses on discovering new features. After this analysis, it goes to the modelling and algorithm stage.

**(e) Modeling and Algorithms**

Modeling and Algorithms is an analytical model that is designed to perform a specific task. Mathematical formulas or algorithms may be applied to the data to identify the relationships among the variables. (A variable is a data item containing values in the form of numbers or characters). Data analysts build models that



correlate the data and predict business outcomes. They need to understand the business well enough and predict whether the results obtained from the mathematical models are meaningful and relevant according to the given situation. For example, in the student's details, identifying the relationship between the variables - Father's name and Phone number would be useful for communication purposes. The next stage is the data product stage.

### (f) Data Product

A data product is a computer application that takes data inputs and generates outputs. In other words, it's a production output from an analysis. For example, in the student's details, marks scored in every exam would help the instructor/teacher to gain insights and make predictions on the performance of the student in the upcoming exam.

An application that analyses the data about a customer's recent purchases and recommends some other purchase that the customer might enjoy based on the choice of products that he/she has purchased is also an example of Data Product stage. After this stage, the data goes to the last stage in data analysis, which is the communication stage.

### (g) Communication

Once the data is analysed, it may be reported in various formats. The users may give their feedback, which results in additional analysis of the data. When determining how to communicate the results, the analyst may consider the data visualization techniques to clearly and efficiently communicate the message to the audience. The data visualization uses information, which is displayed in the form of tables and charts to help communicate the key messages contained in the data. Tables are helpful to a user who might analyse specific numbers, while charts (e.g., bar charts or line charts) may explain the quantitative data effectively.

For example, in the student's details, representing the marks scored by the student in the form of a chart would help his/her parents to easily understand the academic performance of their child.

Thus, the data goes into all the stages mentioned above during the data analysis. To understand this process better, look at the Fig. 3.1 which clearly shows data science process in the form of a flowchart.

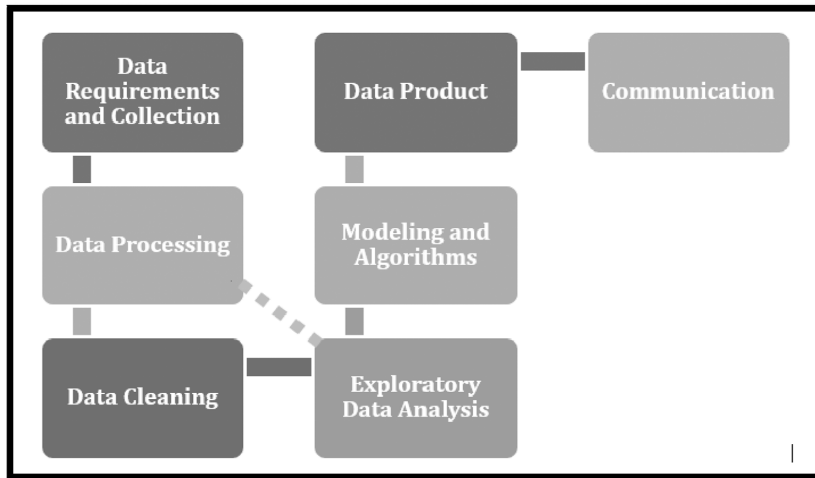


Fig. 3.1: Data Science Process Flowchart

Sometimes the data from one source might not be sufficient to take decisions. So, there may be a situation n, where you need to integrate data from multiple sources for better decision making. To integrate the required data from multiple sources, a data warehouse is created. Let us now discuss data warehouse, its advantages and disadvantages in detail.

### 3.4 DATA WAREHOUSING

Data Warehousing is the process of constructing and using a data warehouse. A data warehouse collects and manages data from multiple sources to provide meaningful insights. It mainly combines data from several sources into one comprehensive database. It acts as a central repository where input is received from different sources. It involves data cleaning, data integration and data consolidations. Since data is pooled from different sources, the amount of data in data warehouse will be very large.

Using this technology is quite effective. For example, in a warehouse, it helps the executives to collect the data and analyse it effectively. Based on the collected data, the executives can take quick decisions after analysis. Warehousing allows us to process large amount of complex data in an efficient manner. The areas where a data warehouse is used are airline, banking, health care, telecommunication, hospitality industry etc.

There are three characteristics of data warehouse. They are integrated, time-variant and non-volatile. Let us discuss about each of these characteristics in brief.

- **Integrated:** The way the data is extracted and transformed is uniform, regardless of the original source.



- **Time-variant:** Data is organized via time-periods (weekly, monthly, annually, etc.).
- **Non-volatile:** A data warehouse is not updated in real-time.

You may have understood the various uses of data warehousing. Let us discuss some of its advantages.

### Advantages of Data Warehouse

- It allows the users to quickly access the critical data from sources, which are available in one place.
- Since data warehouse allows users to access the critical data from various sources at a single place, it saves the user's time of retrieving the data from multiple sources.
- Data warehouse stores a large amount of historical data. This helps the users to analyse different time periods and thereafter generate trends to make future predictions.

Now, let us discuss about some of the disadvantages of data warehouse.

### Disadvantages of Data Warehouse

- Creation and implementation of data warehouse is a time-consuming process.
- The data warehouse may seem easy, but actually, it is too complex for the average users.
- Organisations need to spend a lot of their resources for training and implementation of data warehouse.

When you keep collecting data for a long period of time, the database becomes very large. Extracting the data from a large database, based on your requirement, then becomes very difficult. In order to overcome this obstacle, data mining is required. Let us learn about data mining now.

---

## 3.5 DATA MINING

---

It is the process of analysing very large sets of data to extract useful information from it. It looks for hidden patterns and trends within the data set. Data Mining is primarily used to discover and indicate the relationship amongst the different variables within the large database. Let us now discuss the advantages of the Data Mining:



## Notes

**Advantages**

- **Marketing:** It assists the marketing companies to understand the patterns of purchase transactions by the customers. This will help them to take better decisions for earning more profit. Also, it is beneficial to all the industries as it helps in developing smarter marketing campaigns.
- **Telecommunication:** In the communication field, it helps to identify the call record analysis. For example, monitoring the details of a call and providing the following data such as the date of call, the number from which the call was originated/received and identifying the **International Mobile Equipment Identity (IMEI)** number etc.
- **Financing:** It helps the government agencies to analyse the records of the financial transactions to build the patterns that can detect money laundering or criminal activities. It also helps financial institutions to deduct fraudulent transactions.

Besides these advantages, data mining has some disadvantages too. Let us see some of them.

**Disadvantages of Data Mining**

- **Skilled Persons Required for Data Mining:** The tools used for data mining are very complex, hence a specialised person is required to analyse the data and understand the result.
- **Privacy & Security:** The privacy issues have been increasing in recent times due to the social networks, e-commerce etc. Due to privacy issues, people are afraid to share their personal information, which can easily be manipulated in an unethical way by posing severe threats to the individual or organisation. As large amount of data is collected in data mining systems, confidential data might be hacked by the hackers.
- **Selection of correct Data mining tool:** Data mining tools work in different manners due to different algorithms employed in their design. So, selecting a correct data mining tool is a difficult task. Also there is no guarantee that it would provide accurate results every time which may lead to serious consequences.

These are the disadvantages of data mining. Now, let us discuss about Big Data.

There can be scenarios where the data needs to be collected from the users frequently. If the data is collected from every user frequently, the amount of data



becomes extremely large. We call this as big data. Let us understand the concept of big data and its usage in real-life scenarios in detail now.

## 3.6 BIG DATA

Big data is a term used to describe a collection of very large, highly complex data sets, which are continuously growing exponentially with time. It is so voluminous that traditional data management tools and software techniques cannot store and process it quickly. In the current scenario, this technology is advancing at a rapid pace. It is useful in addressing a range of business activities from customer experience to analytics.

There are many real-time situations, where big data is being used. Let us discuss some of the examples.

### Examples

#### Internet and Social Networking sites

According to a recent survey, data in TeraBytes and PetaBytes gets consumed on the internet and in the databases of social media sites such as Facebook, Twitter etc., on a daily basis. This data is mainly generated in the form of images and video uploads, message exchanges, uploading comments etc.

#### Airways

An aeroplane engine generates huge volume of data in a short flight time. With as many as thousands of flights flying daily, the generation of data reaches up to many PetaBytes (PB;  $1 \text{ PB} = 10^{15}$  bytes) per day.

Now you must have understood the concept of big data and its usage in real-time scenarios. The big data can be classified into two types i.e., structured data and unstructured data. Let us discuss about each one of them in detail.

#### 3.6.1 Types of Big Data

- **Structured Data:** In the structured data, the data can be processed, stored, and retrieved in a fixed format. It refers to the data that has a defined length and format. Examples of structured data are numbers, dates and strings (group of words). The structured data is usually stored in a database. For example, a '**Student**' table in a database is considered as Structured Data as shown at Table 3.3.





Notes

Table 3.3: Structured Data

| Admission No | Student Name | Date of Birth | Class | Gender |
|--------------|--------------|---------------|-------|--------|
| 1001         | Ravi         | 12/12/2004    | 10    | Male   |
| 1132         | Radha        | 02/04/2006    | 8     | Female |
| 1503         | Rajesh       | 01/06/2009    | 5     | Male   |
| 1425         | Priya        | 05/05/2007    | 3     | Female |

- **Unstructured Data:** It refers to the data that lacks any specific form or structure. Unstructured data includes bitmap images/objects, text and other data types that are not part of a database. For example, an email or the search result returned by ‘Google Search’ is considered as an unstructured data as shown at Fig 3.2.

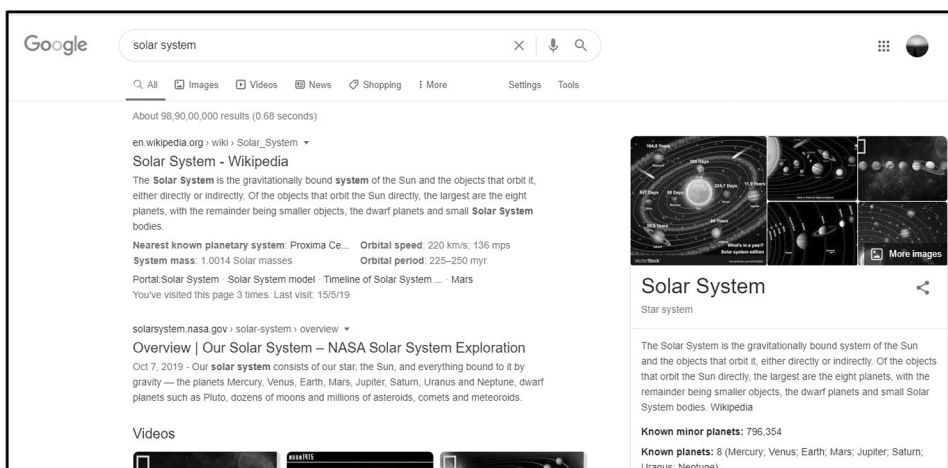


Fig. 3.2: Google Search Result

You can try identifying some of the examples from your day-to-day life and classify the data as structured and unstructured data.

Now, let us discuss some of the characteristics of Big Data.

### 3.6.2 Characteristics of Big Data

- **Volume:** It is one of the most important characteristics, which needs to be considered while dealing with big data as the term big data itself point out towards data which is enormous in size. Also, whether a particular data can actually be considered as a big data or not, that depends upon the amount or volume of the data.



Notes

- **Variety:** It refers to heterogeneous sources and the nature of the data (both structured and unstructured). During earlier days, spreadsheets and databases were the only sources of the data which were considered by most applications. Nowadays, data in the form of emails, photos, videos, PDFs, audio files, etc., are also being considered for analysis purposes. This variety of unstructured data poses certain issues related to storage, mining and correct analysis of data.
- **Velocity:** The term ‘**velocity**’ refers to the speed of generation of data. How fast the data is getting generated and processed to meet the user’s demand, which determines the real potential of data. Big data velocity deals with the speed at which the data flows in from sources such as, business processes, application logs, networks, social media sites, mobile devices, etc. The flow of data is large, fast and continuous.

Ability to process the big data brings in multiple benefits. Let us discuss some of these now.

### 3.6.3 Benefits of Big Data Processing

- The sources such as search engine and social media sites can also be used for businesses, while defining business strategies and taking decisions for earning profits/selling or marketing the products etc.
- You might have seen that company people using traditional customer feedback forms after a customer purchases any products from the company. Now these customer feedback systems are getting replaced by new systems designed with Big Data technologies. In these new systems, big data and Natural Language Processing (NLP) technologies are being used to read and evaluate consumer responses, which leads to improved customer service.
- With the help of big data, the identification of the risk whilst launching the products/services, can be detected at an early stage itself. This leads to better operational efficiency of the product/services.



### INTEXT QUESTIONS 3.1

Fill in the blanks

1. .... strategy is the process of creating plans for handling the data that has been created, stored, managed and processed by an organization.
2. .... is the process of constructing and using data warehouse.



3. The Google Search result is an example of ..... data.
4. .... involves digging through information to identify predictable patterns, interpret results and take business decisions.
5. The characteristics of big data are ....., ..... and .....



**WHAT YOU HAVE LEARNT**

- Data means raw facts gathered about someone or something, which is random.
- Facts, concerning a particular event or subject, which are refined after processing is called information.
- Data analysis is a process of inspecting, transforming and displaying data with the goal of discovering useful information and supporting decision- making.
- Data mining is a process, which is used to extract usable data from a larger set of raw data.
- A data warehouse is constructed by integrating data from multiple diverse sources that support reporting and decision making.
- Big data is a term used to describe a collection of voluminous amount of data that is very large in size and is continuously growing exponentially with time.
- Big data can be classified as structured and unstructured data.



**TERMINAL EXERCISE**

1. Define Data mining.
2. What are the three characteristics of data warehouse?
3. What do you mean by data cleaning?
4. Differentiate between data and information.
5. What are the characteristics of Big Data?
6. Mention the various stages of data analysis.

# MODULE-1

Basic Computing  
Skills (711)



Notes



## ANSWERS TO INTEXT QUESTIONS

### 3.1

1. Data management.
2. Data warehousing.
3. Unstructured.
4. Data analysis.
5. Volume, Variety and Velocity

#### Key Learning Outcomes

- Be proficient in differentiating data from information, identifying the various stages of data analysis, data warehousing and data mining as well as the characteristics of big data.
- Be able to differentiate between structured and un-structured data.

**4****SECURING THE SYSTEM**

---

**4.1 INTRODUCTION**

---

We store a lot of important personal and professional information in our computers in the form of data. You must have heard about the misuse of data happening very commonly now-a-days, isn't it? This shows that the information stored in the computer can be accessed and misused by someone who is not authorised to access it. It is extremely important to protect the information in the computer to keep the information secure. The process of protecting the information stored in your computer is called '**information security**'. In this lesson, you will learn about information security, cyber security and network security. It also gives us an opportunity to learn about how we can protect our data that is stored in the computer.

**OBJECTIVES**

After reading this lesson, you will be able to:

- define information security;
- discuss about cyber security;
- explain network security and
- protect and secure your computer system by following the prevention measures.

---

**4.2 INFORMATION SECURITY**

---

The information stored in the computer and other electronic devices needs to be protected from any theft or damage. Otherwise, the same can lead to loss or leak



of information. The threat which leads to exploitation of information can be caused by accident or it can be intentional. Some of the damages can be permanent, which means once the information is lost, it cannot be recovered. Measures must be taken to ensure that your information is not viewed or accessed without your permission. Also, measures should be taken to protect the information from getting damaged. Information security deals with the measures that one should adopt to avoid damage to the computer and to its data. In this lesson, you will learn about what are the steps to be followed for preventing loss of data and also the measures that need to be adopted to keep the computer safe from natural and man-made threats. Let us discuss about cyber security now.

### 4.3 CYBER SECURITY

Cyber security is the process of protecting and recovering programs, network and data from attacks on the internet that are intended for exploitation. Cyber security covers threats that use internet. The main objective of cyber security is to take the measures against such attacks over the internet and protect the programs, network and data by adopting a few safety measures. You need to ensure that your username, password, credit card or online banking information are secure as these details are prone to attack and could be used by criminals or unauthorized users. Nowadays, many of us are using internet for online learning, online bank transactions, online ticket booking, online shopping etc. But you have to be very careful while accessing internet because some websites can also install malware on the computer without the knowledge of the user. The risk of such actions can be reduced by installing software such as antivirus, anti-spyware, firewalls etc. Strong passwords can also be used to protect data. Let us learn about some of the best practices that should be followed to ensure cyber security.

#### Best Practices for Cyber Security

- **Keep Strong Passwords-** A password should be a combination of alphanumeric (alphabets and numbers) and special characters so that it would be tough for others to find out or guess the password. Most websites check for the effectiveness of the password while creating an account and inform you of the strength of the password (For example, strong, fair, weak etc.) as shown at Fig. 4.1.

The guidelines for managing strong passwords are:

- (i) The length of the password should be at least 08-14 characters.
- (ii) Avoid keeping passwords as sequential numbers, usernames, pet names, your likes and dislikes; which are easy to guess.



Notes

- (iii) It should be a combination of alphanumeric characters and symbols such as ~ !, @ # \$ % ^ & \* ( ) { } [ ] < > , . : " ' ; ' | \
- (iv) Make sure every account of yours has a different password.
- (v) If possible, use random password generators to create new passwords.

|                    |  |                    |        |
|--------------------|--|--------------------|--------|
| Choose a password: | <input type="password" value="123456789"/> | Password strength: | Weak   |
|                    | Minimum of 8 characters in length.         |                    |        |
| Re-enter password: | <input type="password"/>                   |                    |        |
| Choose a password: | <input type="password" value="98765432"/>  | Password strength: | Fair   |
|                    | Minimum of 8 characters in length.         |                    |        |
| Choose a password: | <input type="password" value="987654321"/> | Password strength: | Weak   |
|                    | Minimum of 8 characters in length.         |                    |        |
| Choose a password: | <input type="password" value="98765432A"/> | Password strength: | Strong |
|                    | Minimum of 8 characters in length.         |                    |        |

Fig. 4.1: Password Strength

Are you using a strong password for your online account? If not, please start using a strong password from now on.

### Clear Browser Cookies

You might have noticed that while opening some of the websites, the system usually asks your permission to allow the use of cookies. Cookies are small files that are created on the local computer when you visit these websites. They provide a way for the website to keep track of the users' preferences. Though the cookies store data based on the activity performed during the visit to the website, such as login details, visited web pages in a website, etc., the same information could also be tracked by unauthorized users who try to gain access to your personal information stored in your computer. But now-a-days, popular browsers give a feature to implement do-not-track option, which allows the users to block the cookies that may track your browsing activity.

Now we will discuss about how to clear cookies data and block cookies. The process is explained for Mozilla Firefox, however, it is similar for all the browsers.

- (i) Open the Mozilla Firefox (any other browser) and click **Open Menu** icon on the top right corner of the browser.

# MODULE-1

Basic Computing Skills (711)



Notes

(ii) Click **Options** from the drop-down menu (refer Fig. 4.2).

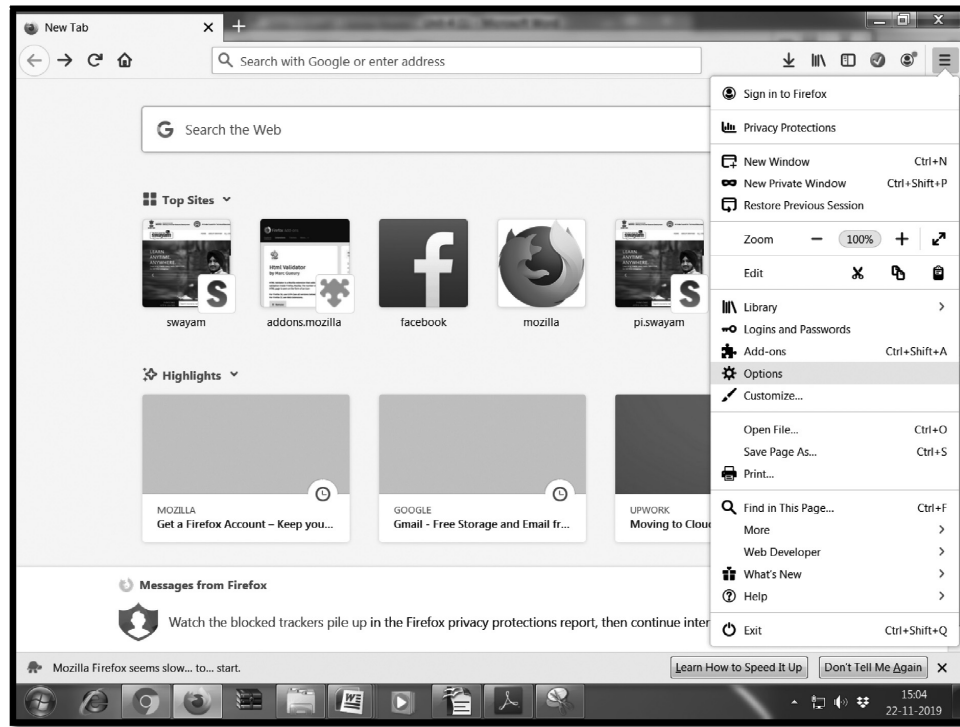


Fig. 4.2: Mozilla Firefox Options Menu

(iii) Click 'Privacy & Security' option.

(iv) From 'Cookies and Site Data', click 'Clear Data' button (refer Fig. 4.3).

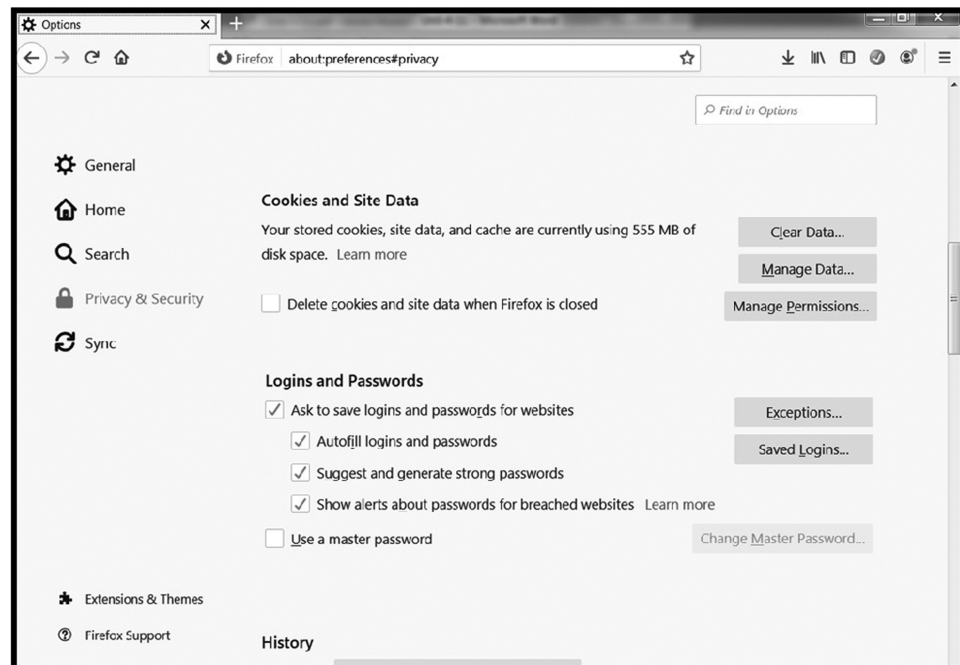


Fig. 4.3: Privacy and Security Option



- (v) Clear Data dialog box will be displayed. Check the box **Cookies and Site Data** and click on **Clear** button (refer Fig. 4.4). The **Clear all cookies and site data** confirmation box will be displayed as shown at Fig. 4.5.

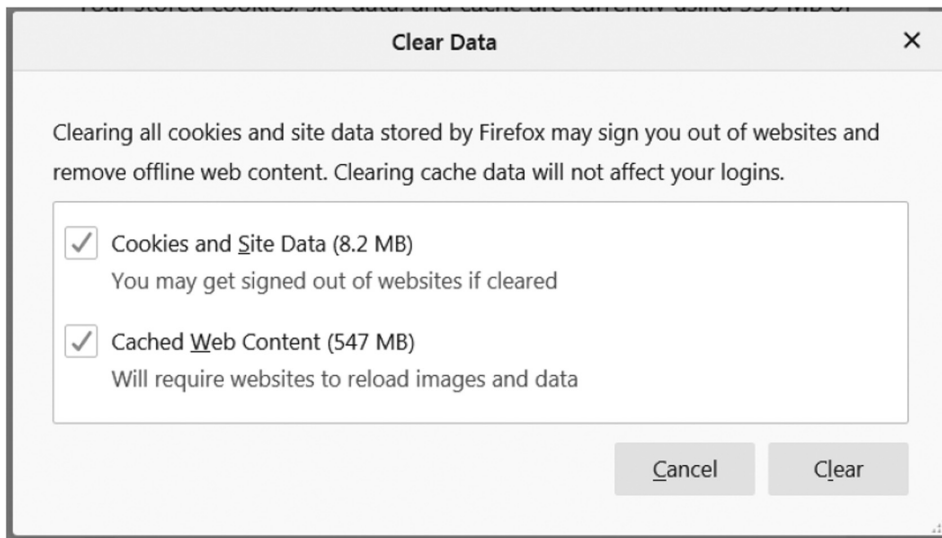


Fig. 4.4: Clear Data Dialog Box

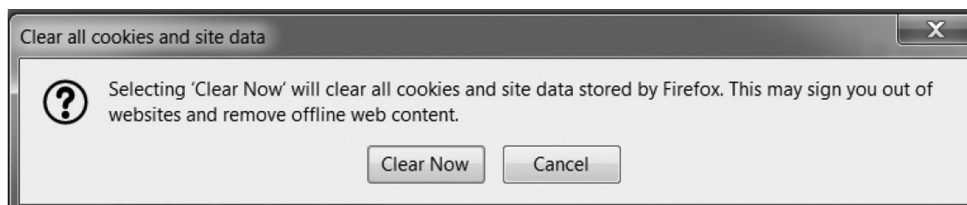


Fig. 4.5: Clear All Cookies and Site Data

- (vi) Click **Clear Now** button to clear all the cookies.

The cookies stored in your browser will get deleted. Once you have cleared the cookies, now you can block them by choosing **Privacy & Security** option (refer Fig. 4.3). Choose **Block cookies and site data** → **Third-Party trackers** option from the drop-down menu (refer Fig. 4.6). Now the cookies for third-party trackers will be blocked in your computer.



Notes



Notes

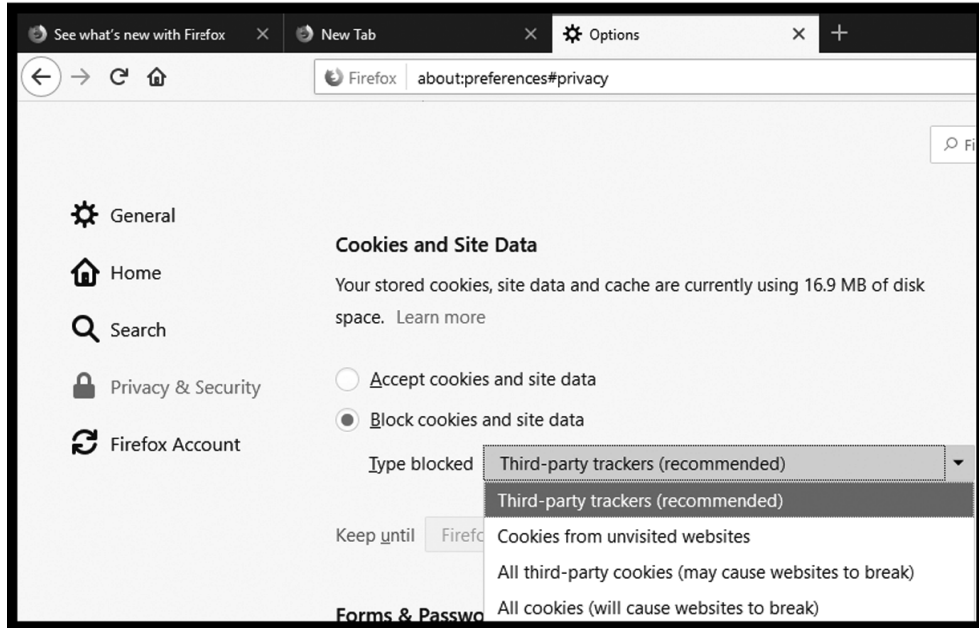


Fig. 4.6: Block Cookies and Site Data

You can try to delete or block cookies in your browser to keep your personal information safe.

**Install Anti-Virus and Anti-Spyware Software**

You have already learnt about anti-virus software in lesson 2 that it will block and detect unwanted software program installations and remove those programs, if installed. Similarly, anti-spyware software will prevent, detect and remove any spyware. It also provides real-time protection for the computer. Some of the popular examples of anti-virus software are Norton, McAfee, Kaspersky etc.

Keep your antivirus and antispyware programs always up-to-date in your computer. This helps in protecting your computer from threats. In order to protect your computer from virus, do not download any attachment, which is sent by an unknown sender through emails. Even if it is from a known sender, scan the attached file for viruses and ensure that the attachment is not having any virus, only then download the attached file. It might be a fake e-mail trying to gather your personal information such as bank account details, home address, debit/credit card details and so on. Some e-mails urge you to click on the links saying you have won a lottery and pretend to transfer a large amount of money to your bank account. Do not get trapped by these mails. These are fraudulent mails, which could compromise the security of the information and data stored in your computer. Ensure that your computer has an anti-virus software installed in it.

**Take Precautions While Registering with Websites**

Read the terms and conditions, privacy statement or policies, whenever you create an account on a website. This will help you to understand how the website uses your personal data, which you have provided while registering with that particular website.

**Be Careful While Sharing Personal Information**

You have to be very careful, while accessing online platform. Before browsing the website, you have to ensure that the website is trust-worthy as there are chances that the information given by you can be misused by unauthorized users.

If you are using social networking sites such as Facebook, Instagram etc., then do not give permission to the site to access your personal details. You can change the settings of these networking sites and give the personal detail access to only your friends and family members and that too with due diligence. Do not accept any friend request, which is sent by unknown persons.

You might have used WhatsApp in your mobile. You can change the privacy settings of your account and do not give the permission to everyone to view your profile picture. People can download the picture and misuse it. Even though we are having lot of applications which keep us connected with friends and family members. But these apps require our permission for accessing our personal information, which means safety is in your hands.

**Secure Transactions**

You might have purchased some products through online shopping. While doing online shopping or transactions, the websites save your credit card or debit card or online banking information such as your credit card number, debit card details account details, etc. This information can be tracked and used by unauthorized users called '**hackers**' to misuse it. Ensure that the website is safe while entering banking information by checking the URL of the website. If the website uses the term '**https**' in the website address, then it is a secure one. Also, you can notice a lock symbol in the address bar of the web browser, which ensures that the website you are using is secured.

**Backup Your Data**

Sometimes you may not be able to access the files in your computer or any other devices because it may be corrupted or misplaced. It is recommended to always



keep copies of the data in other devices such as CDs, DVDs, pen drives or other drives (D:, E: etc..) of your computer system which could be helpful in situation when data is lost. Information can also be backed upon various secure cloud services such as Google Drive, DropBox, etc.

### Install Firewall

Firewall is network security system comprising of software, hardware or both that prevents unauthorized access to the network by monitoring and controlling incoming and outgoing network traffic, based on the security rules. Firewall enforces a set of rules and determines whether a third-party can send any information to your computer thus keeping your computer and the network secure.

### Install Software from Trusted Websites

Always install software from trustworthy websites. Go-through the review/ comments given by the other users who had already downloaded the software from this website. Also, understand the purpose and nature of the software before downloading and installing it. Once you have downloaded any software, then it will be having the rights to access your device.

### Clear Browsing History

Web browsers have built-in password management, which is designed to store the passwords and some other information entered in the online forms while accessing a website. Browsers save usernames and passwords when users attempt to log in the website. This facility is helpful because frequently used login details are stored by the web browsers and the users need not type or remember the username and password every time they login to the website. However, this feature is not safe on public or shared computers. It is always a good practice not to save usernames and passwords while you are using public or shared computers. Also, delete the browsing history after the session is completed. To clear the personal data from a web browser such as Mozilla Firefox, you need to open the browser and follow the steps given below:

- Click **Open** menu icon on the top-right corner of the Mozilla web browser.
- Choose **Options** from the drop-down menu and click **Privacy & Security** option.
- Then click **Clear History** button in the History section (refer Fig. 4.7).



Notes

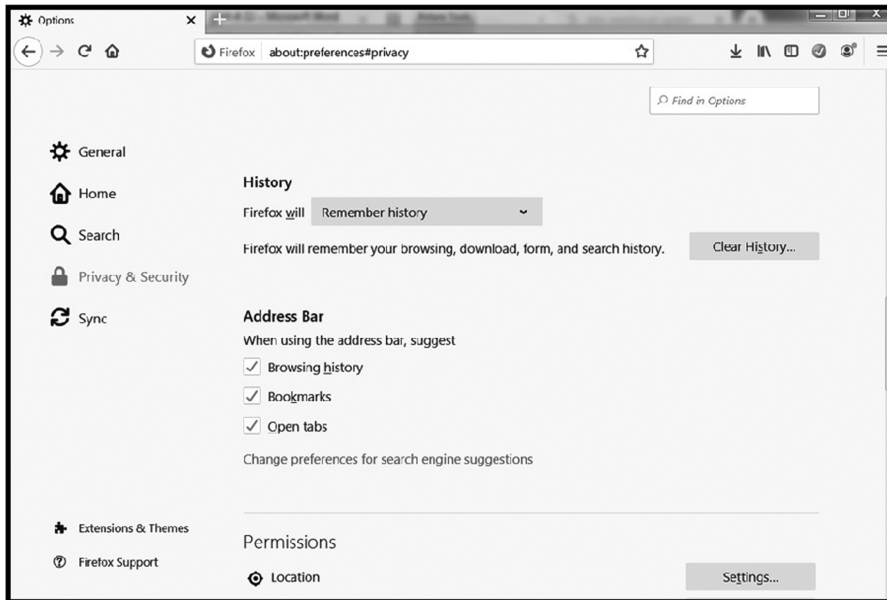


Fig. 4.7: Privacy and Security Option

- In the **Clear All History** dialog box, select the time range to clear and click **Clear Now** button (refer Fig. 4.8). All the saved login details, and recently visited webpages will get deleted.

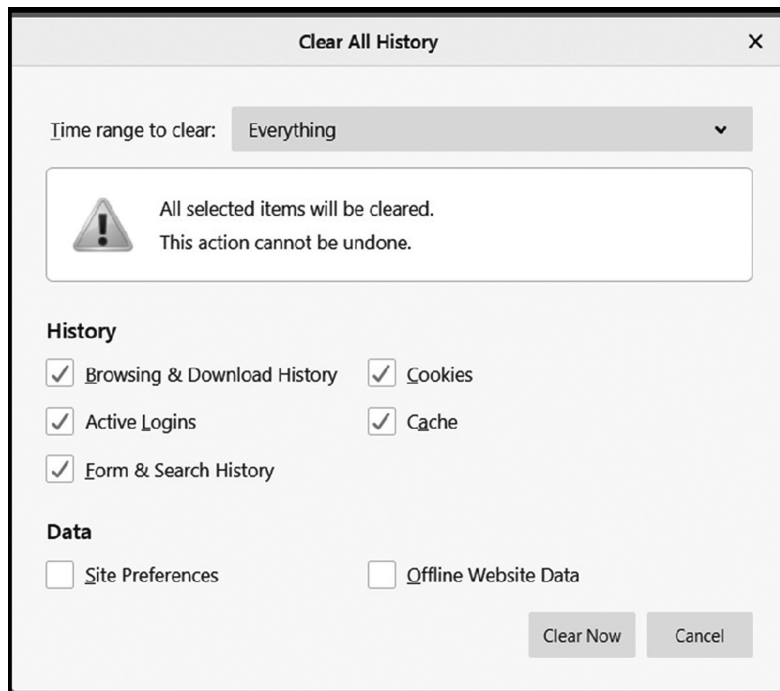


Fig. 4.8: Clear All History Dialog Box



Notes

You can try to clear the browsing history in your browser frequently. After having learnt cyber security, now let us discuss about network security.

### 4.4 NETWORK SECURITY

In this digital world, there are many organizations, which rely on computer networks to share information in an efficient, productive and safe manner. Computer networks are becoming larger and more complex every day. If each member in an organization has a dedicated workstation, a large organization would have many workstations and servers on the network. It is necessary that the information stored in the network should be maintained safely from unauthorized access. There is a possibility that each computer in the network have different operating systems, hardware and software. Also, it is used by users with different proficiency level.

Most users use internet for learning, doing transactions, sending/receiving emails etc. The computers in the network are directly connected to the internet may be unsecured. This unsecured computer network will become a target for the users who want to gain unauthorized access to the information in the network. So, it is very important to keep the network secure. Network security, refers to the measures taken to protect the data in the network. It includes both hardware and software resources. It ensures confidentiality (prevents unauthorized use of information), integrity (ensures the accuracy of information) and availability (authorized users have timely access to information). It also provides access control, which checks the authenticity of the user before the data is used by the user in the network.

The personal data which we save in the computer needs to be secured. Therefore, it is important to take measures to protect your computer from damage. There is a possibility of losing data, due to natural or man-made threats.

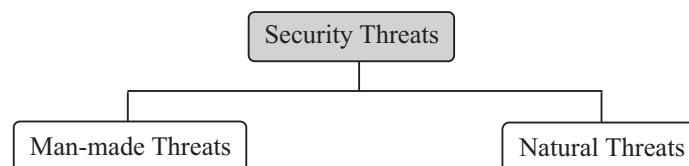


Fig. 4.9: Threats

You will now learn about the natural threats and man-made threats and the preventive measures you should take to protect your computer from these threats.



## Notes

#### 4.4.1 Natural Threats to Computers

- Lightning that strikes with high electrical charge can cause a surge. Many parts of the computer such as CPU, memory unit, motherboard, modems and routers can also be affected.
- Temperature is a major factor which affects the performance of the computer. Most electronic devices are designed to function within a specific temperature. Extreme heat or extreme cold will result in causing a malfunction in the computer.
- The major damage can be caused by fire which is irreplaceable. The heat from the fire makes the sensitive components of the computer to melt. The smoke from the fire can cause the CPU to overheat and damage it.

#### Preventive Measures to Protect Computers from Natural Threats

The preventive measures that can be adopted to keep the computer safe from natural threats are:

- **Backup Data:** It helps to restore the data if it is lost. You should take the backup in any external storage device such as CD, DVD, pen drive and hard disk at regular intervals. With cloud storage, the process of backup data has become simpler and easier.
- **Placing Computer in Secure Location:** You should install the computer in a secure place, where it will not be affected by natural threats. For example, you can avoid placing the computer in a room where it receives direct sunlight, dust or moisture.
- **Maintaining Ambient Conditions:** Make sure the room's temperature and humidity is optimum for the computer to work without any malfunction. Air conditioners and humidity controllers help maintain a proper working ambient environment.
- **Preventing Loss of Data:** Install Uninterrupted Power Supply (UPS) to provide power supply in case a power failure takes place. Since UPS prevents the computer from shutting down suddenly, the data may not be lost.
- **Fire Insulation:** Insulate computers from fire by placing them in fireproof surroundings. You can also install fire safety equipment and procedures for data damage control.

Now let us discuss about the man-made threats and the preventive measures you can take to protect your computer from these threats.



## 4.4.2 Man-Made Threats

Computers receive threats from humans in the form of spies, crackers, hackers who try to breach network security. They illegally access confidential information such as email passwords and bank account details etc. They are also called '**white-hats**'. Once they gain unauthorized access, they destroy important information and cause problems for their targets.

### Human Threats to Computers

- **Device Theft:** Electronic devices such as computers, laptops or smartphones can be stolen by thieves when they get physical access to these devices.
- **Identity Theft:** Hackers can use other person's identity to steal bank account number, credit card/debit card number, ATM pin etc., for making online transactions or purchases. This would result in major financial losses to the victim.
- **Virus, Worms and Trojan Horses:** You should understand that the virus is a computer program, which can damage the data stored in the computer. The virus spreads through CDs, DVDs, pen drives or while downloading the files from the internet. Worms are the programs that replicate fast and occupy memory space of a computer system. Trojan horse is a destructive program which claims to be useful but harms the computer.
- **Spyware:** It is a malicious program that gets installed in a computer without the knowledge of the user. These programs scan our online browsing activities and leak the information to third parties.
- **Internet Scams:** As you already know that internet scams are taking place when people receive e-mail messages that are related to offering attractive prizes, but the real purpose of the mail is to deceive the users and take their money or personal information. Such e-mails contain messages, which make the users believe that it is from a trusted source/organization asking the users to visit a website and give personal information such as password, bank account details, credit card/debit card or pin number etc. However, these websites are fake and they are used to gather user's personal information. With the details provided by the user, the author of the fake website steals user's identity and commits crimes with that identity.
- **Online Predators:** When children use the internet, they may get in contact with strangers, which put them at great risk. Children who chat with strangers online are easy targets for online predators. It is not advisable for anyone to communicate with strangers online.





Notes

- **Accidental Damage to the Computer:** Carelessness can cause damage to the computer. Spilling tea, coffee or water, can cause accidental damage to the computer. Dropping any device, which was attached with the computer also causes the device to malfunction. If any of these things happen, it is suggested to take the device to an expert and get it repaired or recover the data stored in it.
- **Accidental Deletion of Data:** There may be a chance that you may accidentally delete files. The deletion of files moves those files into the recycle bin folder, but the permanent deletion happens when **Shift+Del** keys are pressed together or when the items are deleted from the recycle bin also. You have to take regular backups of your files for avoiding deletion of data.

There are many ways that you can follow to prevent your computer from man-made threats.

### Preventive Measures to Protect Computers from Man-Made Threats

The measures that can be adopted to keep your computer safe from man-made threats are as follows:

- **Backup Data:** One can take a backup of data periodically and store it in a secure location. Now-a-days, a lot of cloud storage options are available, which can be used to ensure data security.
- **Data Encryption:** Data can be encrypted using the encryption software. You are already familiar with encryption and decryption. The encrypted data will not be in the readable form. Therefore, it becomes very difficult for unauthorized users to read and interpret the data. For example, bitLocker feature in Windows 7 and Windows 10 enables disk encryption.
- **Installing Firewalls:** In a networked environment, firewall (acts a virtual wall) prevents human threats to the computers, which are connected in the same network. Installing firewall is a safe way to protect your computer as it stops malicious users to access your computer.



### INTEXT QUESTIONS 4.1

Fill in the blanks:

1. .... is the method of protecting programs, network and data from unauthorized access or attacks that are intended for exploitation.

## MODULE-1

Basic Computing  
Skills (711)



Notes

### Securing the System

2. A ..... is a malicious program which destroys and damages the data in the computer.
3. .... provides battery backup in case of power failure.
4. .... refers to creating multiple copies of the data.
5. Spyware programs scan our ..... activities and leak the information to third parties.



#### WHAT YOU HAVE LEARNT

- The process of protecting information stored in your computer is called '**Information Security**'.
- Cyber security is the method of protecting programs, network and data from unauthorized access or attacks that are intended for exploitation.
- Password should be a combination of upper case letters, lower case letters, numbers and special characters.
- The cookies are small files that are created on the local computer when we visit websites.
- Keep your antivirus and antispyware software always up-to-date in your computer.
- Ensure that the website is safe while entering banking information by checking the URL of the website. If the website uses the term '**https**' in the website address, then it is a secure one.
- Firewall is a network security system comprising of software, hardware or both that enforces a set of rules and determines whether a third-party can send their information to your computer thus keeping your computer and the network secure.
- Always install software from trustworthy websites. Before downloading and installing any software, first understand the purpose and nature of the software.
- It is always a good practice not to save usernames and passwords while you are using public or shared computers.
- Network security refers to the measures taken to protect the data in the network.

- Spyware are malicious programs that get installed in the computer without the knowledge of the user. These programs scan your online browsing activities and leak the information to the third parties.



### TERMINAL EXERCISE

1. Define cookies.
2. What are the benefits of clearing browsing history from the web browser?
3. What are the three primary features of network security?
4. Mention two man made threats and three preventive measures to protect computers from human threat.
5. Write any three best practices to be followed to ensure cyber security.



### ANSWERS TO INTEXT QUESTIONS

#### 4.1

1. Cyber security.
2. Computer virus.
3. Uninterruptible Power Supply (UPS).
4. Backup data.
5. Online browsing.

#### Key Learning Outcomes

- Be able to follow the best practices for ensuring information security.
- Take preventive measures for protecting computer system from human threats/natural threats.



## MODULE-1

Basic Computing  
Skills (711)



Notes

# 5

## E-GOVERNANCE

---

### 5.1 INTRODUCTION

---

We all know that government periodically introduces many policies and provides many services for the benefit of its citizens. With the advancement of technology, interaction between government and the citizens have become simpler and easier. Now-a-days government uses electronic medium for delivering new initiatives to citizens. The electronic medium or digital medium provides transparency in the process and delivers the information to the citizens in a very fast and effective manner. People can really benefit by the government's initiatives. In this lesson, you will learn about e-Governance, digital payment methods, e-commerce and m-commerce.



### OBJECTIVES

After reading this lesson, you will be able to:

- define e-Governance;
- explore digital payment methods;
- identify mobile apps for e-Governance;
- differentiate between e-commerce and m-commerce.

Let us begin the lesson with the definition of e-Governance, its types and advantages.

---

### 5.2 E-GOVERNANCE

---

Electronic Governance or e-governance is the application of Information and Communication Technology (ICT) to deliver government services, do transactions



with the government and exchange of information and services between government and the public. It provides an easy access to all the information pertaining to government, thus improving transparency. E-Governance provides speedy information to all citizens. It improves administration efficiency of public services like health, water, power, security, transportation and municipal services.

The e-governance is classified into four types according to the services it provides. Let's discuss about the types of e-governance.

The four types of E-Governance are:

- (a) **Government to Citizen (G2C):** It refers to the government services that can be accessed by common people to reduce the time and cost to conduct a transaction. A citizen can have access to the services anytime anywhere. Online services like license renewal, payment of tax come under G2C category.
- (b) **Government to Business (G2B):** It is the exchange of services between government and business organizations. A business organization can have easy and convenient online access to government agencies which helps in the smooth functioning of the business. It plays a vital role in business development. Online services such as electronic submission of tenders, electronic GST, payments etc., are the examples of G2B category.
- (c) **Government to Employee (G2E):** It aims to bring the employees together to improvise knowledge sharing. It provides online facilities to employees such as applying for leave, checking salary payment records, checking the balance of leaves etc. It also provides human resource training and development which can be taken online. It acts as an intermediate between employees, government institutions and their management.
- (d) **Government to Government (G2G):** It refers to the interactions between different departments, organizations and agencies within the government. This increases the efficiency and quality of government processes. It helps different government departments to establish relations and work together which leads to enhanced service. It can communicate with local government as well as global government.

Can you think of any of the e-governance services that you have used for your learning purpose? If so, under which type of e-governance does it belong to?

Now let us discuss some of the advantages of e-governance.



Notes

## Advantages of e-governance

- (a) It provides a wide range of services to the citizens in an efficient and cost-effective manner.
- (b) It facilitates better communication between government and businesses.
- (c) It enables sustained availability of information and transparency in the process.
- (d) It supports and simplifies governance for government, citizens and businesses by connecting all three parties together.
- (e) It offers better delivery of services to citizens, improved interaction with businesses and industries, citizen empowerment, better management, greater convenience, time saving and cost reductions when compared to paper based system.

Can you think of one advantage of the e-governance service that you are using for learning purpose?



## INTEXT QUESTIONS 5.1

Fill in the blanks.

1. .... is the application of Information and Communication Technology (ICT) to deliver government services, communication transactions, exchange of information and services between government and the public.
2. .... refers to the interactions between different departments, organizations and agencies within the Government.
3. .... provides online facilities to employees like applying for leave, checking the balance of leaves, checking salary payment records etc.

## 5.3 EXPLORING DIGITAL PAYMENT METHODS

You would have studied in your history subject that in the ancient times, people exchanged goods without the need for money. At that time, there was no currency involved in business trading. However, people could only exchange goods when both the parties were in need of each other's products. This resulted in increased inconvenience and difficulties. That is when the concept of money came into existence. Money, being the most basic form of credit allows people to accept the payment in a belief that they will be able to use that money to purchase something



which they want later. But as the transactions increased and amount involved in the transaction became more, handling hard cash money was difficult. In order to handle situations like these, digital payment methods were adopted.

### Digital Payments

Digital payment is a method of paying for products/goods or services electronically using electronic devices instead of using cash or cheque. In digital payment, the payer and payee, both use electronic modes to send and receive money. No hard cash is used for the transaction.

The Government has been taking several measures to promote and encourage digital payment methods. There are various types and modes of digital payments. Some of these include the use of debit/credit cards, internet banking, mobile wallets etc. Digital payment methods are easy to use and provide customers the flexibility to make payments from anywhere anytime. These are good alternative to traditional methods of payment as it speeds up the transaction cycles.

Can you think of places where you have noticed people using digital payment methods?

There are many digital payment methods. Let's see some of the popular digital payment methods now.

#### The popular digital payment methods are:

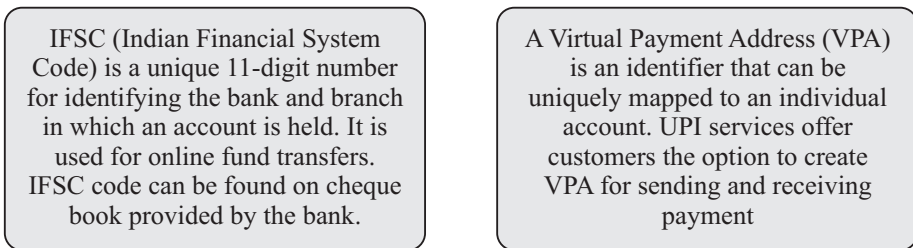
- (a) **Banking cards:** Cards are among the most widely used digital payment methods. These cards come with various features and benefits such as security of payments and convenience. Moreover, they can also be used to make payments for any online purchases. Examples of banking cards are debit/credit cards, prepaid cards etc.
- (b) **USSD (Unstructured Supplementary Service Data):** It is another type of digital payment method. It can be used to carry out mobile transactions without downloading any app. The main aim of this type of digital payment is to create an environment of inclusion among the underserved sections of the society and integrate them into mainstream banking. It is used for fund transfer, to have a look at the bank statement and make balance queries.
- (c) **AEPS (Aadhar Enabled Payment System):** AEPS can be used for all banking transactions like balance enquiry, cash withdrawal, cash deposit, fund transfers etc. All transactions are carried out through a banking correspondent based on Aadhar verification. There is no need to visit the bank



**Notes**

physically. This service can be availed only if the Aadhar number is registered with the customer’s bank account.

- (d) **UPI (Unified Payment Interface):** It is a type of payment system through which any customer holding a bank account can send and receive money through a UPI based app. This service allows the users to link any number of bank accounts on a UPI app on their smart phone to easily initiate fund transfers. The main advantage of UPI is that it enables money transfers without a bank account or IFSC code and it is instant. The user has to create a **Virtual Payment Address (VPA)** and payments will happen with this VPA. To use this service, one should have a valid bank account and a registered mobile number which is linked to the same bank account.



**Fig. 5.1**

- (e) **Mobile Wallets:** It is a type of virtual wallet service that can be used by downloading the app. The digital or mobile wallet stores the bank account information in an encoded format to allow secure payments. The user can add money to the mobile wallet and use that amount for payments to merchants, and for sending and receiving money from other users. Some mobile wallets may charge a certain transaction fee for the services offered. Examples of mobile wallets are Mobikwik, PayTM.
- (f) **Bank prepaid cards:** A prepaid card is an alternative banking card that lets the user spend the money which is loaded onto the card. Like a debit card, a prepaid card works at any merchant that accepts its payment network such as Visa or Master card. It is safer and more convenient than using cash.
- (g) **Bharat Interface for Money (BHIM) app:** It is a payment app that lets the user to make simple, easy and quick transactions using **Unified Payments Interface (UPI)**. Users can make direct bank payments to anyone on UPI using their UPI ID or scanning their QR code with the BHIM app. BHIM app can be used by anyone who has a mobile number, debit card and a valid bank account. These digital payment methods are very useful as they have many benefits as compared to cash transactions.



Due to these popular digital payment methods, money transactions have become much simpler to handle. Let us discuss about the advantages of digital payment methods.

### Benefits of Digital Payments

Some of the benefits of digital payments are:

- (a) **Faster, Easier and Convenient:** Cashless payments speed up the payment process as there is no need to fill in lengthy forms. You would have noticed, while shopping, digital payment methods help to avoid people waiting in long queue in the cash counter.  
There is no need to stand in the long queue to withdraw money from ATM. Digital payments are very convenient as the services are available 24×7 all 365 days.
- (b) **Economical:** Most of the payment apps do not charge a transaction fee or processing fee for the services. For example, UPI services can be availed by the users free of cost.
- (c) **Discounts and Rewards:** There are many discounts, rewards and cash backs offered by banks to the customers who use digital payment apps and mobile wallets.
- (d) **Easy Bill Payment Process:** Digital payment platforms are very convenient for paying utility bills. Whether it is electricity bill, internet bill, phone bill, all such utility bills can be paid using single app.
- (e) **Maintenance Record of Digital Transactions:** All the transaction records can be maintained properly. Unlike cash transactions, customers can keep track of each and every digital transaction that is made no matter how small the transaction amount is.

With all these advantages, digital payment methods are getting popular nowadays as it makes our transactions secure and hassle-free. Not only for online payments, there are many apps launched by the government for e-governance to make lives better for the citizens of the country.



### INTEXT QUESTIONS 5.2

Fill in the blanks.

1. .... refers to the use of smartphone or other cellular devices to perform online banking transactions such as monitoring account balances, transferring funds between bank accounts, bill payments.



Notes



2. .... is a payment app that lets the user to make simple, easy and quick transactions using Unified Payments Interface (UPI)
3. .... is a type of virtual wallet service that can be used by downloading the app.

5.4 MOBILE APPS FOR E-GOVERNANCE

There are many mobile apps that ensure electronic services by the government are available to people. With increase in the accessibility of mobile phones even in rural areas, the government is promoting the delivery of e-governance services through mobile apps. The mobile apps provide fast and easy access of services and information to citizens, thus establishing a two-way communication between government and citizens and promoting empowerment of citizens. Let us discuss about some of the mobile apps which enable easy and round-the-clock access to public services.

- (a) **DigiLocker:** It is a key initiative under Digital India that enables citizens to store official documents on the cloud-based platform which is safe and secure. This service is aimed towards reducing the need to carry physical documents. Aimed at the idea of paperless governance, DigiLocker is a platform for issuance and verification of documents and certificates in a digitally (refer Fig. 5.2).

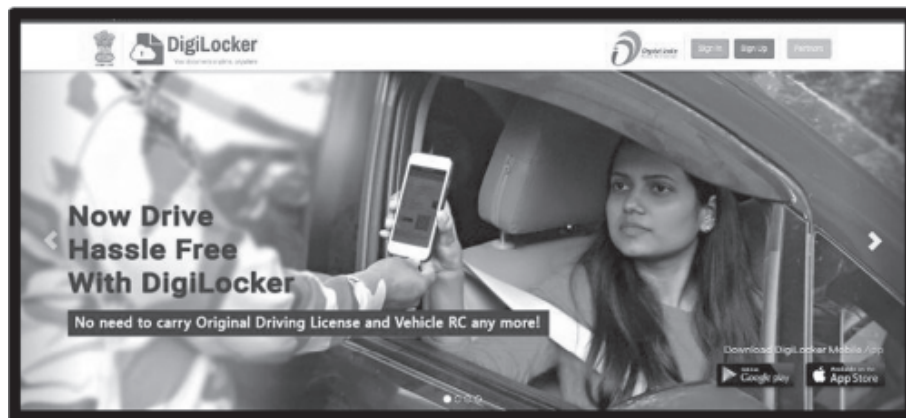


Fig. 5.2: DigiLocker

- (b) **Mobile-Swachh Bharat Mission (mSBM):** mSBM has been launched by Ministry of Drinking water and sanitation for monitoring the progress under Swachh Bharat Mission at the central level. Citizens can click pictures related to civic or sanitation issues and send to concerned municipal authority which

would help authorities to take quick action and resolve the issue (refer Fig. 5.3).

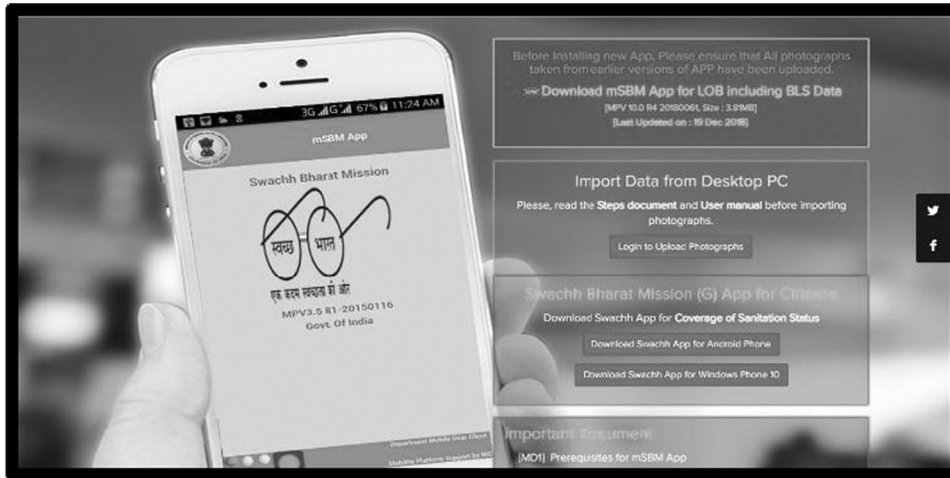


Fig. 5.3: Swachh Bharat Mission

- (c) **mAadhaar:** mAadhaar app is an official app developed by Unique Identification Authority of India (UIDAI). It aims to let users carry their Aadhaar Identity on their smart phones. The biggest advantage of mAadhaar app is that it enables users to share their eKYC (Know Your Customer) information with any service provider, anywhere. Users can also view and share their Aadhaar profile using QR code. Fig. 5.4 shows the interface of mAadhaar app.

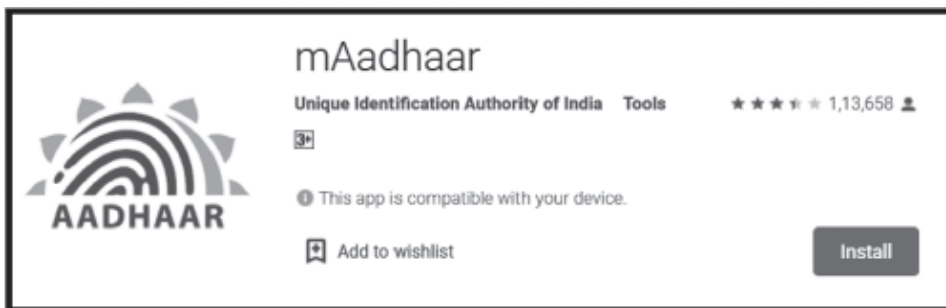


Fig. 5.4: Aadhar app

- (d) **IRCTC:** IRCTC (Indian Railways Catering and Tourism Corporation) is one of the most popular apps by the government. It enables online booking of railway tickets. It comes with an integrated IRCTC e-wallet for faster and easier transactions. Users can book, cancel, enquire and check the status of their tickets using this app.



Notes



Notes

- (e) **MyGov:** MyGov app acts as a platform for citizen partnership in governance. Citizens can provide ideas, comments, suggestions and feedback to ministries and associated organisations. Users can also participate in policy formulation and programme implementation. It is an innovative citizen engagement platform by the Indian government.
- (f) **UMANG:** UMANG (Unified Mobile Application for New-age Governance) app brings together all government departments and their services on a single platform to provide better and easier services to citizens (refer Fig. 5.5). It is an evolving platform that provides Digital India services like Aadhaar, DigiLocker, PayGov and many more on a single platform.

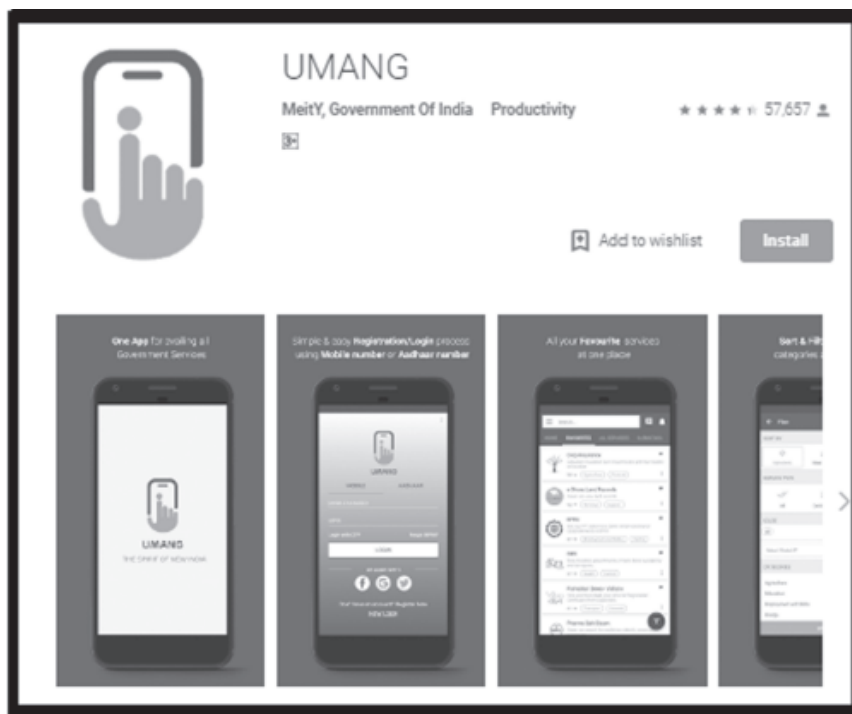


Fig. 5.5: UMANG App



INTEXT QUESTIONS 5.3

1. .... is a key initiative under Digital India that enables citizens to store official documents on the cloud-based platform which is safe and secure.
2. .... app acts as a platform for citizen partnership in governance. Citizens can provide ideas, comments, suggestions and feedback to ministries and associated organisations.
3. The mAadhaar app is an official app developed by .....



---

**Notes**

---

**5.5 E-COMMERCE**

---

In today's competitive and convenience focussed world, customers no longer prefer to venture into the street to buy items. Instead, they want to shop from the comfort of their own home. With the advancement of technology, many websites have features that make it easy for the customers to browse for products and buy them online. Buying and selling of goods or services through the internet and the transfer of money to execute these transactions is called E-commerce, also known as electronic commerce. Whenever you buy or sell a product using the internet, it means that you are involved in e-commerce. E-commerce often refers to the sale of physical products online, but it can also include any kind of commercial transaction that is facilitated using internet.

Independent freelancers, small businesses and large corporations have benefited from e-commerce which enables them to sell their goods and services at a large scale which was not possible with traditional offline retail.

Due to the accessibility of internet to millions of people across the world, e-commerce has deep-rooted its way into the business world. It helps organization to provide better customer services. It simplifies the business transactions and makes them faster and more efficient. E-commerce websites allow customers to post their comments/feedback regarding products and services. This feedback can be used by the business organization to improve its services and deliver better quality products. E-commerce development is highly important in today's competitive world. The major benefit of e-commerce development is its lower start-up cost. The cost involved in setting a physical store is much more than the cost involved in setting it up online.

---

**5.6 M-COMMERCE**

---

With the growth in smartphones and the increase in the number of mobile internet users, buying and selling of goods or services using a mobile device has become very common now-a-days. It is an advancement of e-commerce which enables people to do business transactions from anywhere, anytime by simply using a mobile phone or a tablet device.

Using wireless handheld devices like mobile phones or tablet to conduct commercial transactions online is called Mobile Commerce or M-Commerce. It also includes the purchase and selling of products, online banking and bill payments using mobile phones. Mobile commerce has a very wide reach as a huge part of the world's population has a mobile phone available with them all the time. It provides the customers very convenient and easy way to do business transactions.

Can you think of some examples of m-commerce apps that can be used for online shopping?



Notes

Since m-commerce enables the user to buy/sell products or use any services anytime anywhere, it has many benefits. Let us discuss about the benefits of m-commerce.

**Some of the benefits of M-Commerce are:**

- M-Commerce plays a vital role in business. Location-based services have allowed customers to browse using their mobile devices in-order to find service that are in close proximity to their current location.
- M-Commerce offers a lot of opportunities, both for connecting with new audience and also allowing brands to expand market as they gain insight on the behavioural pattern of customer’s transaction.
- With the help of M-Commerce, brands are able to scale everything from inventory to marketing. For example, they can increase their mobile app marketing when consumer interest spikes and downscale when the interest declines.
- Compared to shopping physically at any store where customers have to walk across a number of products to shortlist a few of them, in m-commerce they can easily navigate and get desired products in a few clicks using the mobile apps.
- Customers can get alerts in the form of push notifications in their mobile phone about latest product releases, promotions and discounts which gives the customers a better engagement and a more personalized experience.

Though e-commerce and m-commerce functions in a same manner, there are differences between them. Let us see the differences between e-commerce and m-commerce using the table given below (Refer Table 5.1).

**Table 5.1: Difference Between E-Commerce and M-Commerce**

| Basis for comparison | e-commerce   | m-commerce   |
|----------------------|--|--|
| Definition           | Selling and buying of products using electronic system.  | Selling and buying of products via mobile devices.   |
| Devices used         | Computers, Laptops, mobile phone etc.  | Only hand-held devices like Mobiles, Tablet, iPad  |
| Portable             | Limited mobility as the devices such as computer is hard to carry. Though laptop is portable it is not as light as mobile phone. | Yes. Since mobile devices are easy to carry, it offers more mobility, accessibility and convenience. |
| User Interface       | Complicated user interface with more features and functionalities available.   | Simple user interface with all the features simplified.  |
| Reach                | Limited reach as it is available in those places where there is internet along with electricity.                                 | Broad reach due to its portability.  |



**INTEXT QUESTIONS 5.4**

1. Buying and selling of goods or services through the internet and the transfer of money to execute these transactions is called .....
2. Using wireless handheld devices like mobile phones or tablet to conduct commercial transactions online is called .....
3. M-Commerce has ..... user interface with all the features simplified.
4. Services like license renewal, payment of tax come under ..... category.
5. .... code can be found on cheque book provided by the bank for identifying the bank and branch in which an account is held.
6. .... app lets us book, cancel, enquire and check the status of train tickets online.
7. .... app brings together all government departments and their services on a single platform to provide better and easier services to citizens.



**WHAT YOU HAVE LEARNT**

- E-Governance is the application of ICT to deliver government services, communication transactions, exchange of information and services between government and the public.
- Digital payment is a method of paying for products / goods or services electronically using electronic devices instead of using cash or cheque.
- Internet banking, also called as e-banking or virtual banking, refers to the process of carrying out banking transactions online.
- DigiLocker is a key initiative under Digital India that enables citizens to store official documents on the cloud-based platform which is safe and secure.
- Buying and selling of goods or services through internet and the transfer of money to execute these transactions is called e-commerce.
- Using wireless handheld devices like mobile phones or tablet to conduct commercial transactions online is called Mobile Commerce or M-Commerce.



Notes



Notes



### TERMINAL EXERCISE

1. What are the four types of e-governance?
2. Define VPA.
3. How MyGov app is helpful in e-governance?
4. Write the examples of banking cards
5. Discuss any five advantages of e-governance.
6. Compare e-commerce and m-commerce.
7. What are the benefits of digital payments?



### ANSWERS TO INTEXT QUESTIONS

#### 5.1

1. Electronic Governance or e-governance
2. G2G
3. G2E

#### 5.2

1. Mobile banking
2. BHIM
3. Mobile Wallets

#### 5.3

1. DigiLocker
2. MyGov
3. Unique Identification Authority of India (UIDAI)

#### 5.4

1. E-Commerce
2. M-Commerce
3. Simple
4. G2C
5. IFSC
6. IRCTC
7. UMANG

#### Key Learning Outcome

- Be able to avail e-Governance Services and adopt digital payment methods.



**6****SOCIAL NETWORKING**

---

**6.1 INTRODUCTION**

---

We all know that there are many websites that help millions of people around the world to communicate with each other and share information. Websites such as Facebook, YouTube, LinkedIn and Twitter allow people to connect with their friends, relatives and professionals to share ideas and information about their company products and professional services. In this lesson, let us discuss about social networking, its advantages and disadvantages and communication tools.

**OBJECTIVES**

After reading this lesson, you will be able to:

- define social networking;
- list the advantages and disadvantages of social networking;
- explain social media tools;
- discuss communication tools.

---

**6.2 SOCIAL NETWORKING**

---

Social networking site is an online platform which can be used to connect with people or establish relations with other users who share similar personal or career interests, hobbies, backgrounds or real-life connections. The social networks are virtual networks to connect people and organizations. These sites allow the users to share ideas, photos, videos or posts to express their views about any real-world



events. There are different websites that you can use to create personal accounts/profiles and share pictures, videos with your friends, family or with the people from all over the world.

There are many social networking sites available for different purposes such as Facebook, Twitter, LinkedIn, Pinterest, YouTube, Skype (refer Fig. 6.1). Though these are well-known social networking forums, there are many new websites that are coming up regularly that let people connect and interact over the web. With these websites, users can make new friends, establish business connections or extend their base by connecting with people of similar interests. You can see the icons of some of the social networking sites in the image (Fig 6.1) below.

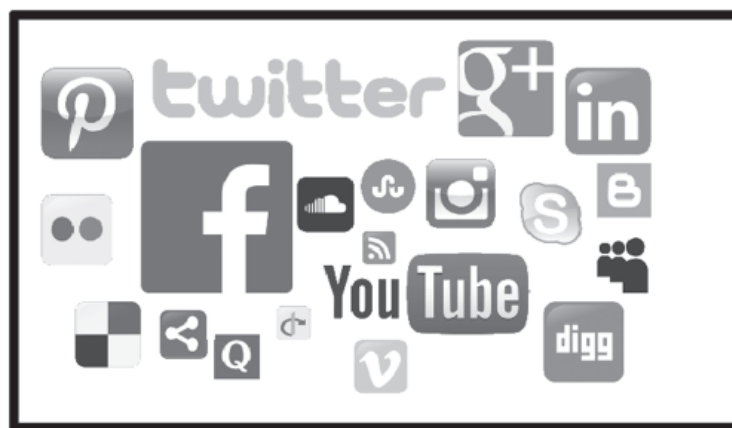


Fig. 6.1: Social Networking Sites

Let us discuss the advantages of Social Networking sites.

### Advantages of Social Networking

Social networking has immense advantages. Some of these are discussed below.

#### Global Connectivity

Social networking sites give the users a mode to get connected with the people world-wide. There is no easier way to get in touch with the people who live in different countries, where it is not possible to meet personally. It gives an opportunity to stay in touch and be connected.

#### Sharing Common Interests

Social networking sites gives us the option of choosing individuals with similar likes and dislikes and builds a network of people around those commonalities. For example, if you are a book lover or a great chef or a person interested in gardening, you can find and interact with those who share the same interests. It would be a

great platform to share tips and ideas from each other. Sites like Pinterest help you to learn and share information about hobbies, crafts, cooking, gardening and other DIY (Do-it-yourself) activities. By pinning and sharing, you can create a network of like-minded people.

### **Real-Time Information Sharing**

Social networking websites have an instant messaging feature which allows you to exchange information in real-time via chat. This feature helps teachers to facilitate classroom discussions, managers to organize meetings, business people to interact with their clients etc.

### **Professional Expansion**

It is very useful for business people to expand and promote their business to a large number of audiences by using social networking. This not only brings more profit to the business but also lowers the expenses towards marketing and advertising. It helps businesses to reach right audience which improves sales and reputation.

### **Noble Cause**

Social networking sites can be used for noble cause like social welfare as well. You can reach out to social media to help the people in need. It is the fastest and the best way to help the needy, share their issues and to get help in terms of money or advice.

Though social networking offers many advantages, it has its drawbacks too. Let's discuss some of the disadvantages of social networking sites.

### **Disadvantages of Social networking**

#### **Addiction**

There are possibilities for a user to get addicted to social networking sites which would have an impact on their personal life. Children and teenagers are more addicted to the social networking sites these days. They tend to get involved extensively and eventually cut themselves off from the real world leading to lack of emotional connection with people around them.

#### **Health Issues**

Spending excess time on social networking sites can also have a negative impact on a person's health. This may happen by being in a sitting position for long hours and continuously looking at the monitor screen. It results to ill health conditions like headaches, back pain, eye-sight problems and laziness.





Notes

### Security Issues

Personal data and privacy can be easily hacked which may lead to financial loss to a person. If the account gets hacked, it might even affect the individual's personal life. Identity theft is also a major concern while using social networking sites. It is the responsibility of the user to protect and secure their personal data.

### Emotional Issues

Children have become the victim of cyber bullying over the past on social networking sites. Since anyone can create a fake account and post anything without being traced, it has become common practice to bully others on social networking platforms. Threats, inappropriate messages and rumours can be sent to disturb someone emotionally. If children are not aware about handling these issues, it leads to emotional anxiety and they become victims of cyber bullying.

### False and Unreliable Information

Anything which is posted on social networking sites can go viral in a very short span of time. This is because people fail to check the authenticity of the information and forward/share it with others without bothering to check whether the information or the news is valid and correct. Sometimes, fake news is being spread which can cause panic among the people in the society.

We have discussed about some of the advantages and disadvantages of social networking. It is the responsibility of each individual to use social networking sites wisely and take required preventive measures to minimise the negative impacts.



### INTEXT QUESTIONS 6.1

Fill in the blanks.

1. Social networking sites allow users to share ..... that expresses their views about online or real-world events with others who are there in their network.
2. Social networking websites have an instant messaging feature which allows people to exchange information in ..... via chat.
3. Most of the time, ..... is being spread as people fail to check the authenticity of the information which can cause panic among the people in the society.
4. A ..... service is an online platform which people use to establish relations with other users who share similar personal or career interests.

Now you have understood about the advantages and disadvantages of social networking sites, let us discuss some of the examples of popular social networking tools that are commonly used by people for various purposes. Most of these sites are available in both mobile app and web version.

**Notes**

---

### 6.3 SOCIAL NETWORKING TOOLS

---

As you know that social networking tool is an online platform which people used to socially connect to each other by sharing information, opinions about an issue facilitating conversations between groups of people online, let us see some of the popular social networking tools.

**(a) Facebook** 

It is one of the largest and most popular social media tools in the world. The unique feature of Facebook is its dynamic and multi-functional interface that brings current and latest news, games and multimedia content in addition to social interactions using text, audio and video.

**(b) YouTube** 

It is the largest video networking tool in the world. There are lot of movies, music, cartoons available on this platform. Anyone with an account can upload video files and share it with their family, friends and even with general public. The videos are free to watch but one must be careful in choosing what you watch on YouTube.

**(c) Twitter** 

It is a free social networking tool for microblogging. It allows the registered users to share posts called *tweets*. A tweet contains text up to 140 characters. The restricted word limit makes the posts catchy. Users can also share photos, videos, hyperlinks and GIFs on Twitter.

**(d) Instagram** 

It is a social networking app dedicated for photo sharing. It allows users to edit and upload photos and short videos. The users allowed to share their life events with friends through a series of pictures captured with a mobile device.



**Notes**

**(e) LinkedIn** 

It is a popular social networking site for professional networking. The employers can post the job opportunities in their organization or institution and job seekers can post their resume and seek for job opportunities. It is useful for the people from similar career backgrounds to connect and showcase business related information and statistics to expand the business.



**INTEXT QUESTIONS 6.2**

Fill in the blanks.

1. .... is the largest video networking tool in the world which contains lot of movies, music and cartoons.
2. Identify the logo given below and mention the name of the logo.



.....

3. Instagram is a social networking app dedicated for ....., which allows users to edit and upload photos and short videos.

**Communication Tools**

There are numerous ways in which people communicate with each other over the internet. Some of these are email, instant messaging, blogs, forums, chat rooms etc. These tools are called as communication tools. Let us discuss some of the communication tools that we use on a daily basis.

**(a) Instant Messenger**

Instant messenger app allows users to share texts, photos, videos, audio and documents in real time. One of the most popular instant messenger apps is WhatsApp. Facebook messenger is another popular example of this category of tools. These instant messengers also provide high quality audio and video calling, which make them more useful. Even variety of groups can be created in these apps where the information can be shared to several people at once. These instant messengers are mobile based apps, but their web version is also available. This is also one of the popular social media tools with over one billion users all over the world.

**(b) Email** 

Electronic mail is the exchange of digital messages between people using electronic devices. An email message is a text consisting of brief message which is sent or received over a network. While email messages are usually simple text messages, we can include attachments (such as image, audio, video, documents etc.). There are many email providers like Gmail, Yahoo, Hotmail etc.

**(c) Video Conferencing**

A video conference is a live, visual connection between two or more people from different locations for the purpose of communication. It provides transmission of high-quality video and audio between multiple locations. The main advantage of video conferencing is that users can see each other while communicating. For video conferencing, users need webcam, microphone and speakers.

**(d) Audio Conferencing**

It is the practice of connecting multiple users on an audio conference call from different locations for the purpose of communication and collaboration. To speak and hear in audio conference calls, users need speakers and microphone. Audio conferencing is a cost-effective communication tool that is, easily accessible and it saves money and time.

**(e) Chat Rooms**

A chat room is a website that provides a platform for community of users with a common interest to communicate in real time. It allows users to exchange messages about a subject in a synchronous manner. Chat room also has the capacity for interactive messaging.

These are some of the communication tools that help us for collaboration and stay connected.

**INTEXT QUESTIONS 6.3**

Fill in the blanks.

1. .... is a mobile-based instant messaging app that allows sharing of text, photos, videos, audio and documents.
2. To speak and hear in audio conference calls, users need ..... and .....



Notes

# MODULE-1

Basic Computing  
Skills (711)



Notes

## Social Networking

3. A ..... is a website that provides a platform for community of users with a common interest to communicate in real time.
4. The use of electronic communication to bully a person by sending inappropriate or threatening messages is called .....
5. .... tool is a collection of online platforms which people use to share information, opinions, experiences, insights and perspective about an issue facilitating conversations and interactions between groups of people online.
6. Twitter allows the registered users to share posts called .....



### WHAT YOU HAVE LEARNT

- Social network site is an online platform which people use to establish social networks or relations with other users who share similar personal or career interests, hobbies, backgrounds or real- life connections.
- Video conferencing, audio conferencing and chat are real- time communications.
- Twitter is a free social networking tool for microblogging. It allows the registered users to share posts called tweets.
- Instagram is social networking app dedicated for photo sharing. It allows users to edit and upload photos and short videos.
- Electronic mail is the exchange of digital messages between people using electronic devices.
- A video conference is a live, visual connection between two or more people from different locations for purpose of communication.



### TERMINAL EXERCISE

1. What are the devices required for video conferencing?
2. Write the names of email service application.
3. What are some of the popular social networking forums?
4. What is the use of LinkedIn website?



5. What is the difference between audio conferencing and video conferencing?
6. Write any two advantages of social networking.



**ANSWERS TO INTEXT QUESTIONS**



**Notes**

**6.1**

1. Ideas, photos, videos or posts
2. Real-time
3. Fake news
4. Social networking

**6.2**

1. YouTube
2. Twitter
3. Photo sharing.

**6.3**

1. WhatsApp
2. Speakers and microphone
3. Chat room
4. Cyberbullying
5. Social media
6. Tweets.

**Key Learning Outcome**

- Use social networking sites for sharing your views or ideas to the world.

## MODULE-1

Basic Computing  
Skills (711)



Notes

# 7

## OFFICE PRODUCTIVITY TOOLS

### 7.1 INTRODUCTION

Computers have a huge impact on all the aspects of our daily life but one area which can't be imagined without computers are our offices and work place. Computing tools are useful in organising, storing and presenting desired information frequently in an efficient way. A variety of software are available now-a-days to achieve the same. We may specially use office productivity tools like OpenOffice which helps in improving overall efficiency and productivity of the workplace by allowing us to manage office documents efficiently. Let us discuss some of the office productivity tools of OpenOffice in this lesson.



### OBJECTIVES

After reading this lesson, you will be able to:

- discuss OpenOffice;
- use OpenOffice Writer;
- acquaint the skills of Calc;
- acquaint the skills of Impress.

### 7.2 OPENOFFICE

Apache OpenOffice is the leading open-source office software suite, containing packages for word processing, presentations, spreadsheets, databases and more. It is available in many languages. It stores all data in an international open standard format which can also be read and written from other office software packages.



It can be downloaded and used completely free of charge. In this lesson, you will learn about how to use word processor, spreadsheet and presentation packages of OpenOffice named Writer, Calc and Impress respectively. Apache OpenOffice is easy to learn, and can be installed on as many computers as you like. To download Apache OpenOffice, visit the website <https://www.openoffice.org/download> and follow the instructions.

Now we will discuss about the OpenOffice Writer which is one of the popularly used tool for creating documents.

---

### 7.3 OPENOFFICE WRITER

---

Assume your instructor has asked you to submit a soft copy of a report. You might be wondering which software to use to submit it. There are many application programs available for this purpose. One such popular program is Word Processor. A word processor is a type of application software used for composing, editing, formatting, saving and printing documents. It is a package that processes textual content and has many features which help us to create and organize flawless documents. Let's see some of the features of Word processor.

Word processing software provides a general set of tools for creating, editing, formatting and printing documents. We will discuss about word processing skills using OpenOffice.org Writer, also called as OOo Writer or simply Writer.

Let us see some of the features of Word Processors now.

#### Features of Word Processor

- (a) **User-Friendly Interface:** Typing in a word processor is easy and fast.
- (b) **Editing:** Any type of editing like inserting, deleting or modifying the text can be made easily by anyone.
- (c) **Formatting:** The document can be formatted according to the requirement to make it more presentable. Different font styles, colour, size, bold, italic, underline, highlighting features, etc., can be applied to make the document look more attractive.
- (d) **Graphics:** Images, pictures, clip arts can be inserted in the document which enhances the understanding of the concept in the document.
- (e) **Object Linking and Embedding (OLE):** Most of the word processing software allows third-party objects to be embedded or linked in the document. The objects can be of any type like charts, equations, video clips, audio clips, pictures etc.



Notes

- (f) **Spelling and Grammar Check:** Word processors are not only capable of checking the spelling but also suggest alternative spelling if there is a spelling error. Even grammatical mistakes will be checked, and alternatives will be suggested by Word processing programs. This ensures in making error-free documents.

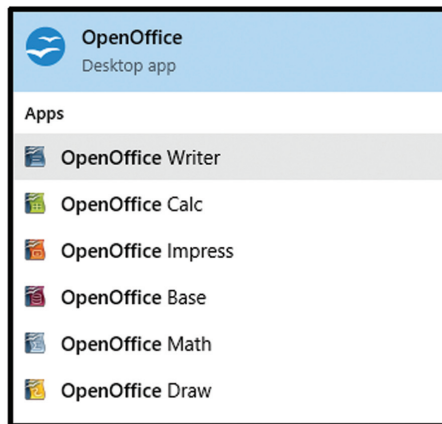
Now, let us see the steps to follow to use Word processor for creating documents.

**7.3.1 Getting Started with Word Processor**

There are five steps involved in working with Word Processor. They are:

- (a) Creating or opening a document
- (b) Typing text
- (c) Editing the document
- (d) Formatting the document
- (e) Saving the document for future use.

Open Writer by clicking on ‘**Start**’ button in the task bar and click ‘**OpenOffice Writer**’.



**Fig. 7.1: OpenOffice Software Package**

The document window with a blinking cursor appears where you can start typing the text. Before working on the document, you must know the basic components of the document window. The document window has many components (refer Fig. 7.2). Let us understand each component now.

- (a) **Title Bar:** It displays the name of the application. By default, the name of the document will be ‘Untitled 1’.



## Notes

- (b) **Menu Bar:** It lists the various menu options File, Edit, View...
- (c) **Formatting Toolbar:** It consists of tools needed for formatting the document.
- (d) **Standard Toolbar:** It consists of most frequently used commands and utilities.
- (e) **Status Bar:** It is present at the bottom of the window and is used to display information about the page numbers, the column and line numbers where your cursor is located at any given point of time.
- (f) **Scroll Bar:** It is used to navigate the document up and down by moving the elevator along the scroll bar.
- (g) **Work Area:** The area in the document window where we enter the text of the document.
- (h) **Ruler:** It is used for setting the margins (top, bottom, right and left margins) and tabs in the document.

Margin is the distance from the text to the paper's edge. The text and the graphics are usually printed within the margin.

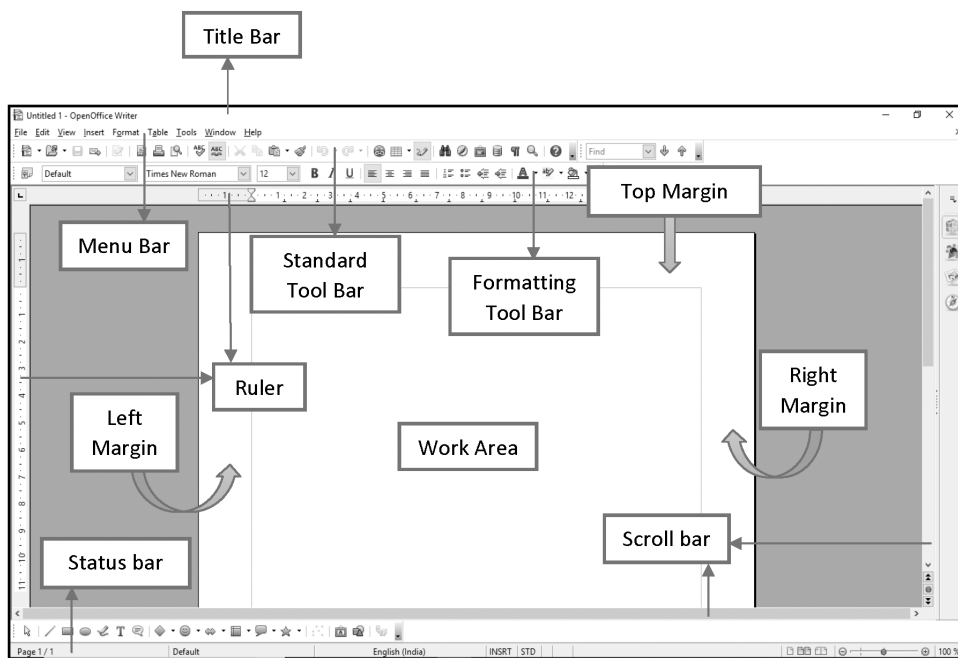


Fig. 7.2: OpenOffice Document Window

# MODULE-1

Basic Computing  
Skills (711)



Notes

Now we will see the steps to create a new document.

## Creating a New Document

To create a new document in the Writer, click **'File'** in the menu bar. Select **'New'** option from the dropdown menu. From the submenu, click **'Text Document'**. A new text document with the name **'Untitled 1'** will be opened.

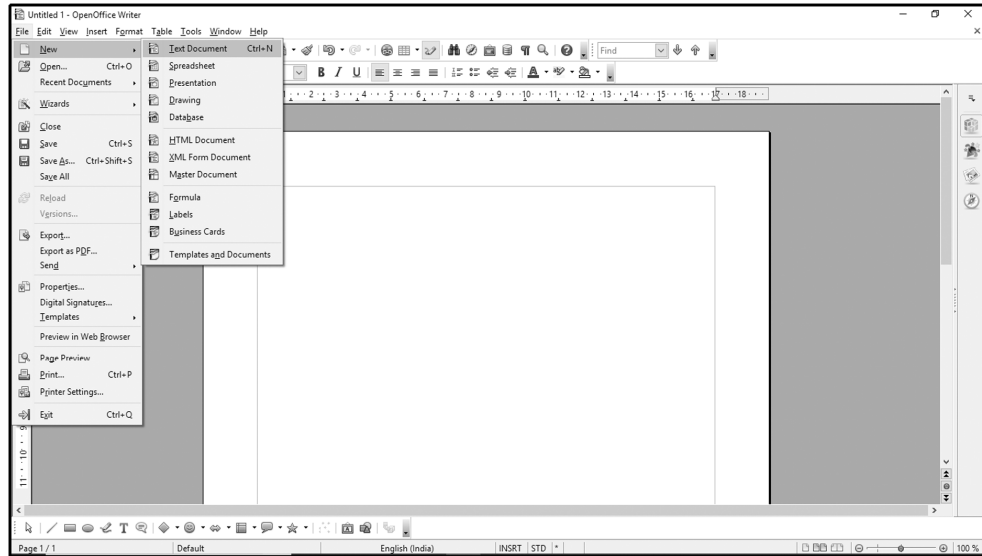



Fig. 7.3: Creating a New Document

Alternatively, you can press **Ctrl+N** shortcut key to open a new text document.

## Typing Text

Once you open a new document, you can start typing in the work area. The area where you type the text in the document is called **work area**. If you observe the mouse pointer, it becomes **I beam pointer** when it is within the work area. When you move your mouse pointer outside the work area, it changes to an arrow .

In Writer, while typing a line, when you reach to the right margin of the page, the text you type further, automatically starts from the beginning of the next line. This feature is called **word wrap**. You do not need to press the **Enter** key to move to the next line. However, you need to press the **Enter** key to:

- Create a blank line,
- End a paragraph,
- End a short line (a sentence that does not reach the end of right margin).

Now let's understand the steps to edit the document.



Notes

## Editing the Document

Editing the document includes making changes in the text after typing. It is one of the most basic Word processing operations. The ability to change the text by adding, deleting and modifying words, sentences and paragraphs is the main operation performed by the users in a word processor software.

Let us see the steps to format the document to make it attractive.

## Formatting the Document

### Font Style

It refers to the appearance or presentation of the text in the document. Writer has many features for formatting which are available in the formatting toolbar.

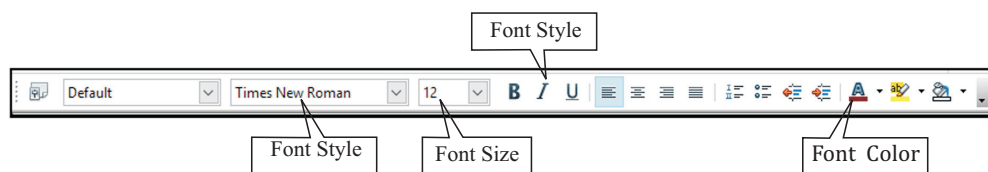


Fig. 7.4: Formatting Toolbar

To change the font style, font color and font size, select the text by dragging the mouse pointer and click on the font style, size and font color options in the formatting toolbar. To make the text bold, italic or underline, select the text, and click on **B**, *I*, U icons respectively.



Fig. 7.5: Formatted Text

After editing and formatting the document, you need to save the document. Let us discuss the steps to save the document.

## Saving a Document

Documents created by using Writer software, exists only in computer's primary memory which is temporary. For permanent storage, we must save the file in the computer's hard disk or in any other secondary storage device. To do so, follow the steps given below-

- (a) Click '**File**' menu in the menu bar.
- (b) From the drop-down menu, select '**Save As**' option. A '**Save As**' dialog box appears.



## Notes

- (c) Choose the folder where you want your file to be saved, type the name of the file in the **'File Name'** text box.
- (d) Click on **'Save'** button to save the document. The document will be saved with the name you mentioned in the **'File name'** text box which will be displayed in the title bar of the document.

You can also save the document by using the shortcut key **Ctrl+S**. By default, the Writer software will save the documents with the extension **.odt**.

Once the editing is completed and the document is saved, you can close it. Now, let us see the steps to close the document.

### Closing the Document

- (a) Click on **'File'** in the menu bar.
- (b) Click **'Close'** option from the drop-down menu.
- (c) In case, you have not saved your document, Writer will show three options as shown below. You have to choose any one of these options.

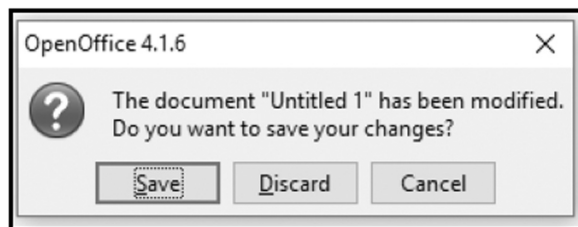


Fig. 7.6: Alert Dialog Box for Saving the Document

- (d) Click on the **Save** button, if you want to save the document before closing.
- (e) Click on **Discard** button, if you want to close the document without saving.
- (f) Click on **Cancel** button, to abort closing and to continue working on the document.

Alternatively, you can also click on the **Close** button in the top-right corner of the window.

### Opening the existing Document

When you want to open the saved document again, you need to follow the below steps.

To open an already existing document,

- (a) Click on **'File'** in the menu bar.





- (b) Click '**Open**' in the drop-down menu.
- (c) The Open dialog box appears.
- (d) Choose the drive and folder where the document is saved as shown in Fig 7.4.
- (e) Select the document which you want to open and click the **Open** button. Your document will be opened.

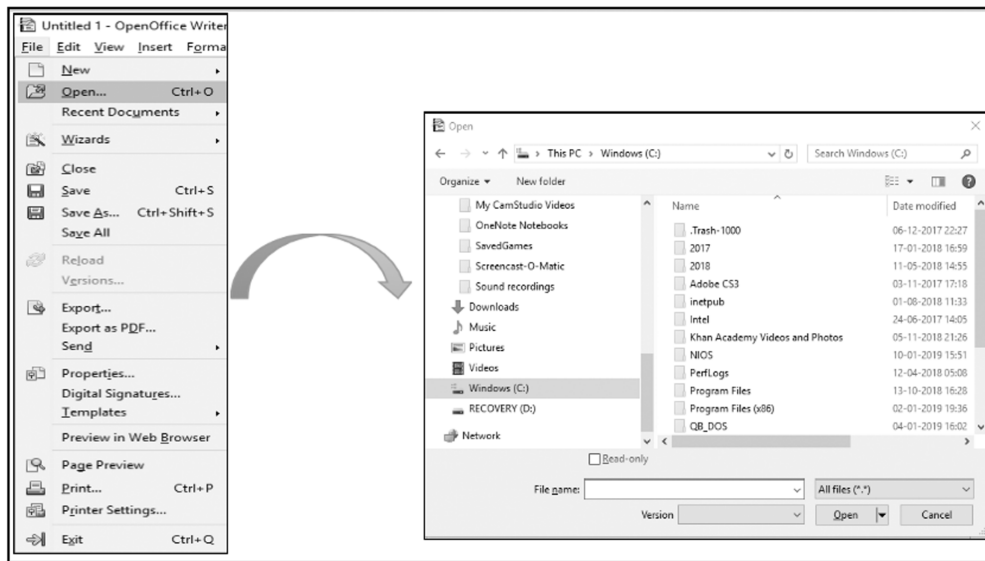


Fig. 7.7: Opening a New Document

After learning how to use OpenOffice Writer, you can create reports, resume, letters, documenting research work and many more.



### INTEXT QUESTIONS 7.1

1. .... is used for setting the margins (top, bottom, right and left margins) and tabs in the document.
2. .... refers to the appearance or presentation of the text in the document.
3. You can press ..... shortcut key combination to open a new text document.
4. What formatting options are applied to the word given below?

Welcome



Assume your teacher has asked you to record the marks of all the students of your class to find out the class percentage. You can use OpenOffice Writer to store the marks of all the students but calculating the class percentage or doing variety of calculation is not easily possible with Writer. Though OpenOffice Writer helps in saving the information, it does not help us in doing variety of calculations. In this scenario, you might need a software which can store data (numbers and figures) and also help you to perform a variety of calculations.

## 7.4 GETTING STARTED WITH CALC

A spreadsheet is a software that lets you to store, calculate, manipulate and analyse set of numbers. One such Spreadsheet application software is OpenOffice Calc which is a part of the OpenOffice.org suite.

Let us see some of the features of Spreadsheet program in brief.

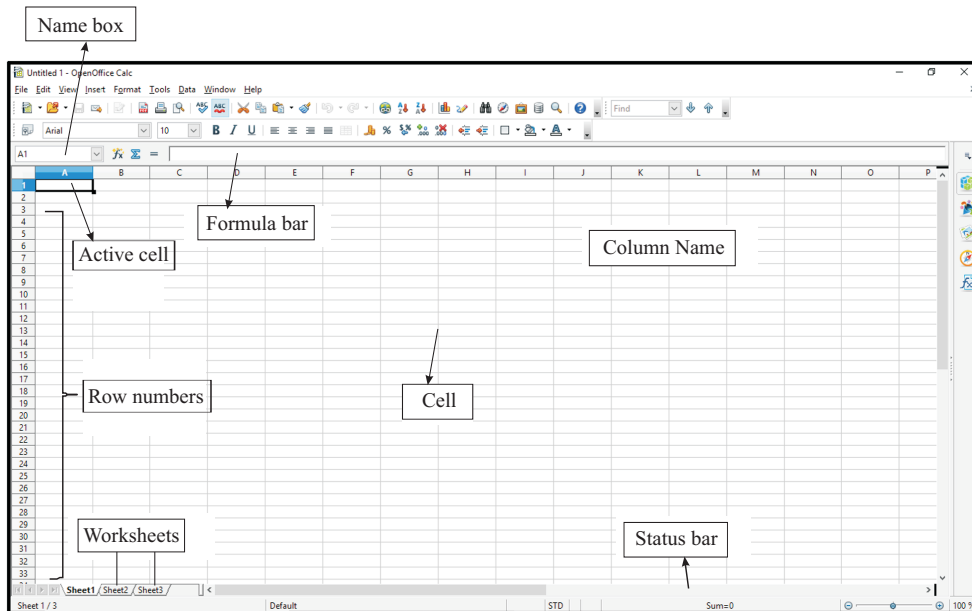
### Features of Spreadsheet Program

- (a) **Performing Calculations** : A spreadsheet software provides powerful support for a variety of mathematical formulas or functions to perform calculations based on numeric data. Spreadsheet formulas are widely used in the financial industry for various types of mathematical modelling.
- (b) **Automatic Recalculation** : Whenever there is any change in the data, it is automatically reflected in the places where formulas, functions or charts are used. Automatic recalculation ensures that the spreadsheet data is always up-to-date.
- (c) **Sorting and Filtering** : You can sort and filter the data in the spreadsheet software.
- (d) **Graphical Representation of Data** : Spreadsheet software can represent data in graphical form using charts which can be useful for quick data examination or analysis.

Let us understand the interface of Calc Window now.

After installing OpenOffice Suite, open Calc by clicking on **'Start'** button in the task bar and click **'OpenOffice Calc'**.

The document window appears as shown in Fig. 7.8 where you can start typing the data. Before working on Calc, you must understand the basic components of the Calc window (refer Fig. 7.8).



**Fig. 7.8: Components of OpenOffice Calc**

Let us understand various components of OpenOffice Calc now.

**Worksheet:** It is a sheet containing grid of cells made up of horizontal rows and vertical columns to store and manipulate the data.

**Workbook:** A spreadsheet program allows you to combine more than one worksheet in a file. Such a file having multiple worksheets is known as workbook.

**Row Number:** As mentioned above, the worksheet is divided into rows and columns. Each row is identified by a row number that uniquely identifies the row. Row number begins from 1.

**Column Name:** Each column is identified by a column name. A column name is a letter and starts from A, B, C.... and so on.

**Cell:** Intersection of a row and column is called a *cell* where data can be stored. Data in a cell can be text, number or formula. Each cell is identified by a unique address which is a combination of Column Name and Row number.

**Name Box:** The address of the active cell is displayed in this box.

**Active Cell:** The cell which is currently selected is called the *active cell*. The cell boundary of the active cell will be highlighted. When data is typed, the data entry takes place in the active cell.

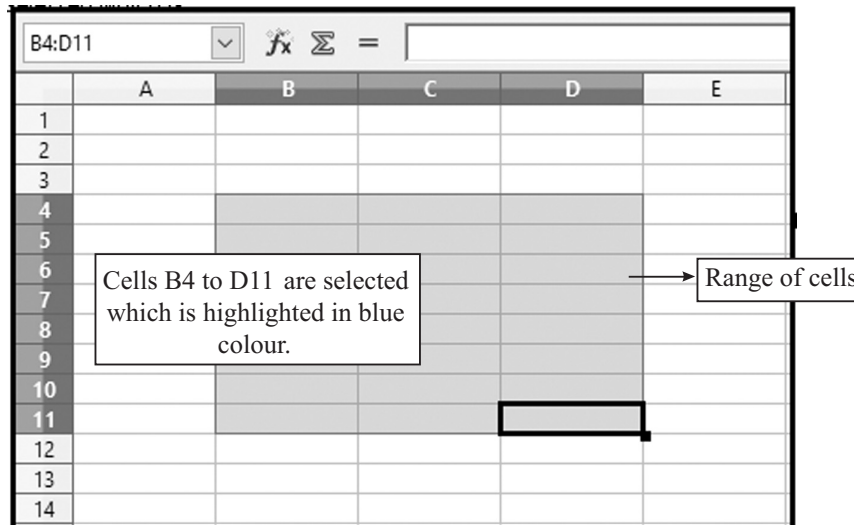


Notes



Notes

**Range of Cells:** A range of cells is a group of contiguous cells that forms a rectangular area in shape. A range may contain a single cell or a group of cells. To select a range of cells, click on a cell and hold the left-mouse button, drag the mouse pointer to the diagonally opposite cell (refer Fig. 7.9). All the cells within that area will get selected. Alternatively, you can also use the navigation keys to select the range of cells.



**Fig. 7.9: Selecting the Range of Cells**

Now you will learn the steps to enter data in worksheet.

**Entering Data in Worksheet**

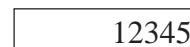
As mentioned before, any type of data – numeric, alphanumeric or formula can be entered in worksheets. To enter data, select the desired cell by placing the cell pointer and then type the data. Three types of data can be entered in Calc worksheet. They are

- (i) Numbers
- (ii) Text
- (iii) Formulas

**(i) Numbers**

These are numeric entries which can be used in calculations. Numbers can contain any of the following characters 0 1 2 3 4 5 6 7 8 9 + - \* ( ) / \$ % , . E e

By default, numbers are always right-aligned in a cell.





Notes

**(ii) Text**

These are text entries in the worksheet. These entries cannot be used in calculations. Text can be any combination of numbers, spaces, and alphabets. All text entries are left aligned.

ABCDE

**(iii) Formula**

These are the entries that describe the relationship among cells. It is an expression that operates on values in a range of cells or a cell. A formula can contain values that can be used for calculations, operators (e.g. + - \* /) and cell address (e.g. C4, B2). A formula should start with = sign. So, the formula to add the values in cells A1 and B1 will be =A1+B1. The formula to multiply the values in cells C3 and D3 will be =C3\*D3.

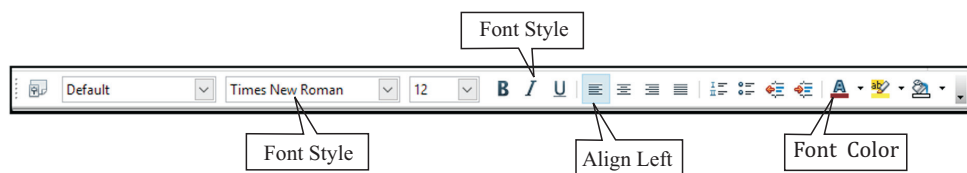
Now we will discuss the steps to format the data in Calc.

**Formatting in Calc**

The general appearance and presentation of data is called formatting. It is the formatting that makes the worksheet presentable. Formatting makes a worksheet presentable and easy to understand.

**Changing Font, Size, Style, Alignment through Formatting Toolbar**

The formatting toolbar contains a set of format buttons that can be used to ease the task of formatting the contents in the worksheet. You can select any cell or a range of cells and click on the desired button on the formatting toolbar and the respective effect will be applied on the data in the cells.



**Fig. 7.10: Formatting Toolbar of OpenOffice Calc**

Now you will learn the steps to perform calculations in Calc.

**Calculations in Calc**

A very important feature of Calc is availability of lots of interesting functions to perform a variety of mathematical and logical operations. Functions are predefined

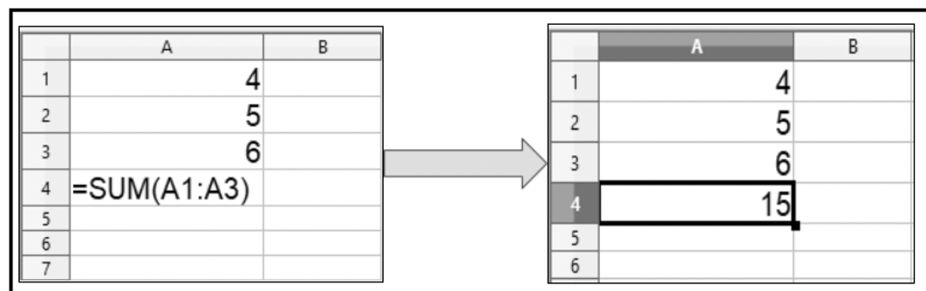


**Notes**

formulas to perform the required operation. Mathematical functions in calc are used to perform various arithmetic operations like sum, average, max, min etc. These functions are used to perform calculations on a range of cells. A function should always start with '=' sign. Let us discuss some of the functions in Calc.

**SUM():** This function is used to add all the values within a cell range. To add the cells from the cells A1 to A3 using SUM function, select the cell where you want to write the function and type:

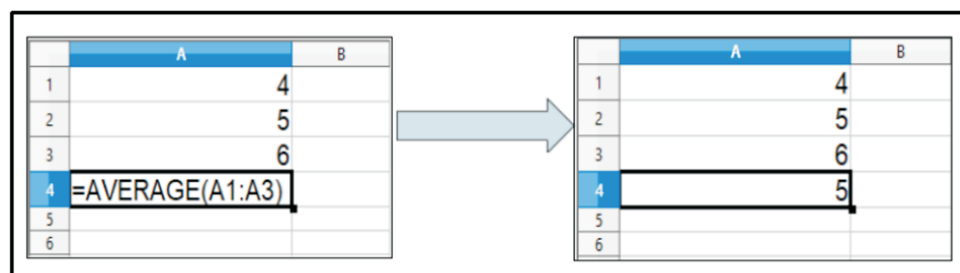
=Sum(A1:A3) and press the Enter key. You will get the answer 15 as shown in the Fig. 7.11.



**Fig. 7.11: Use of SUM( ) function**

**AVERAGE( )**

This function is used to find the average of all the values within a cell range. To find the average of cells in the range A1 to A3, the function is =AVERAGE(A1:A3). The average of values in the cell range will be displayed. In this case it is 5.



**Fig. 7.12: Example for Average ( ) function**

**MAX( )**

This function is used to find the highest value of all the numbers within a cell range. To find the highest value of cells in the range A1 to A3, the function is =MAX

(A1:A3). The highest value in the cell range will be displayed. In this case it will display 6.

|   | A           | B |
|---|-------------|---|
| 1 |             | 4 |
| 2 |             | 5 |
| 3 |             | 6 |
| 4 | =MAX(A1:A3) |   |
| 5 |             |   |

|   | A | B |
|---|---|---|
| 1 |   | 4 |
| 2 |   | 5 |
| 3 |   | 6 |
| 4 |   | 6 |
| 5 |   |   |

Fig. 7.13: Example for Max ( ) function

### MIN ( )

This function is used to find the smallest value of all the numbers within a cell range. To find the smallest value of cells in the range A1 to A3, the function is =MIN(A1:A3). The smallest value in the cell range will be displayed. In this case it will display 4.

|   | A           | B |
|---|-------------|---|
| 1 |             | 4 |
| 2 |             | 5 |
| 3 |             | 6 |
| 4 | =MIN(A1:A3) |   |
| 5 |             |   |

|   | A | B |
|---|---|---|
| 1 |   | 4 |
| 2 |   | 5 |
| 3 |   | 6 |
| 4 |   | 4 |
| 5 |   |   |

Fig. 7.14: Example for Min ( ) function

After entering data, formatting the data and performing calculations in Calc, you need to save the workbook for later use. Let us discuss the steps to save the Workbook.

### Saving a Workbook

To save the work done in Worksheets, perform the following steps:

- Click **'File'** in the menu bar.
- Select **'Save As'** option, from the drop-down menu.
- 'Save As'** dialog box appears.
- Choose the folder where you want your file to be saved and type the name of the file in the **'File Name'** text box.
- Click on **'Save'** button to save the workbook. The workbook will be saved with the name you mentioned in the **'File name'** text box and with the extension.ods.



Notes

# MODULE-1

Basic Computing Skills (711)



Notes

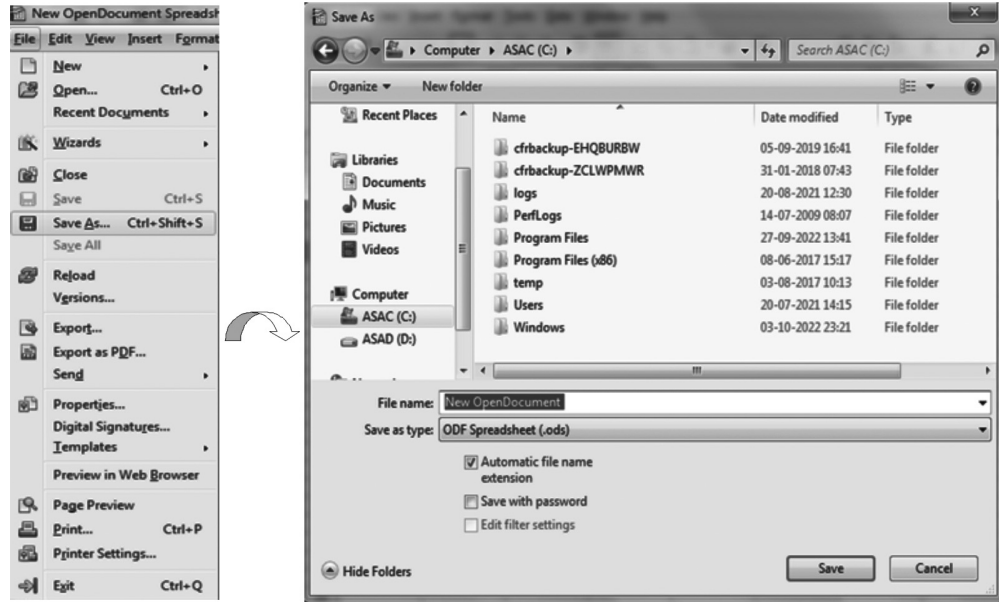


Fig. 7.15: Saving a Workbook



## INTEXT QUESTIONS 7.2

- ..... ensures that the spreadsheet data is always up-to-date in terms of calculations using formulas/functions.
- Spreadsheet software can represent data in graphical form using .....
- ..... refers to the arrangement of data in the cells either in ascending or descending order.
- ..... function is used to find the total of all the values in the cell range.
- Intersection of a row and column is called a ..... where data can be stored.

## 7.5 OPENOFFICE IMPRESS

Assume your teacher has given a project on ‘Solar System’ which you have to present in the classroom in-front of your classmates. In this case, it’s not enough if you just gather and organize data in Word Processor. You need to present your project in a manner which captures audience attention. The traditional method of presenting information was to stand in front of the audiences with a set of notes or prompt cards and to speak clearly and slowly. But with the advancement of



presentation software, now it became the practice of using a presentation show to demonstrate the content. In this section, you will be learning about presentation graphics software called Impress which is a part of the OpenOffice.org suite.

### Introduction to Presentation Software

Presentation software is used to create professional-looking visual aids for the audience. The visual aids can be digital images, charts, graphics, animations etc. Some of the most popular presentation software are PowerPoint, OpenOffice.org Impress etc. Using presentation software, we can create slides which are electronic presentations that can either run on our computer screen or can be projected on a projection screen through a projection device. Slide show is capable of integrating different types of information in the form of text, graphics, sound, audio, video, charts, animation etc.

### What is a Presentation?

The file created by using Impress software is called a presentation. A presentation is a combination of multiple slides. A slide is a combination of different elements such as text box, images, audio, video files etc. When a presentation is played, the slide is shown on the screen one by one. To make our presentation attractive, we can add animation and transition effects to the slide.

You must be very excited to learn how to create a presentation. Isn't it? Yes, so why to wait, lets first learn basic steps to open Impress application.

### Impress Window

After installing OpenOffice Suite, you can open Impress in your computer. To open Impress window, click on the Start button in the Task bar →All Programs →OpenOffice →OpenOffice Impress (refer Fig. 7.16).

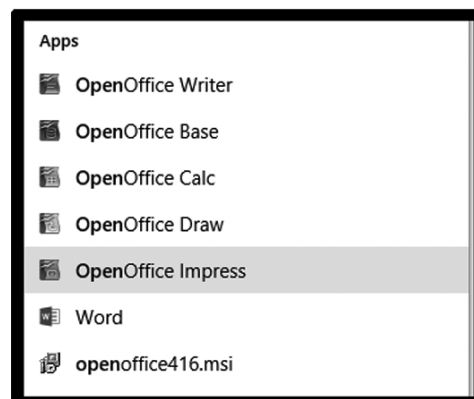


Fig. 7.16: OpenOffice Package



Notes

## MODULE-1

Basic Computing  
Skills (711)



Notes

When you open Impress Window, the screen will look like as shown in Fig. 7.17.

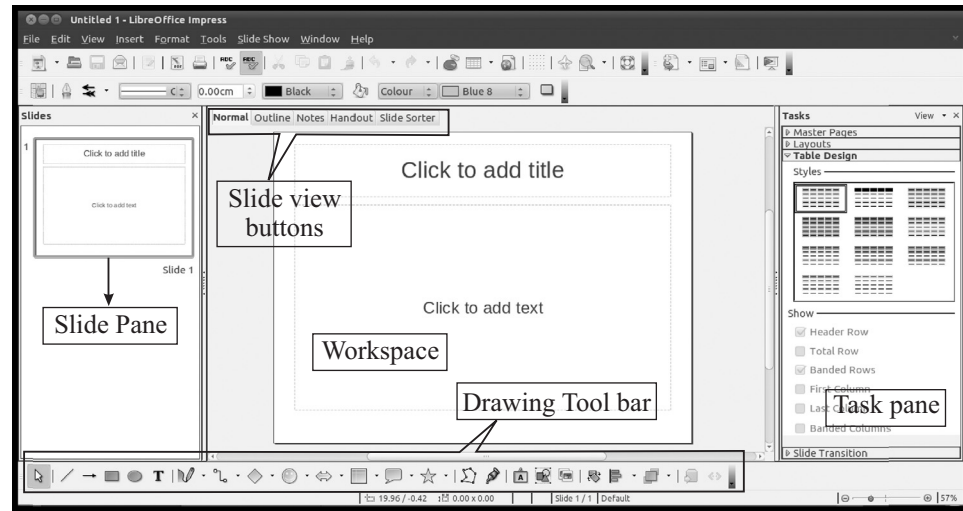


Fig. 7.17: Impress Window and its Components

Now let's understand the basic steps involved in creating a new presentation in Impress.

### Creating New Presentation

When you start a new presentation, Impress provides several options in the 'New Presentation' dialog box. You can create a New Presentation in any of the following ways.

- (a) Empty Presentation
- (b) From Template

When you click File → New → Presentation, Presentation Wizard dialog box appears (refer Fig. 7.18) that provides two ways of creating a presentation. Let us learn about each of these two ways, one by one.

#### Step 1:

When you select 'Empty Presentation' from 'Presentation Wizard', it will allow you to create the presentation from scratch. Click Next button.



Notes

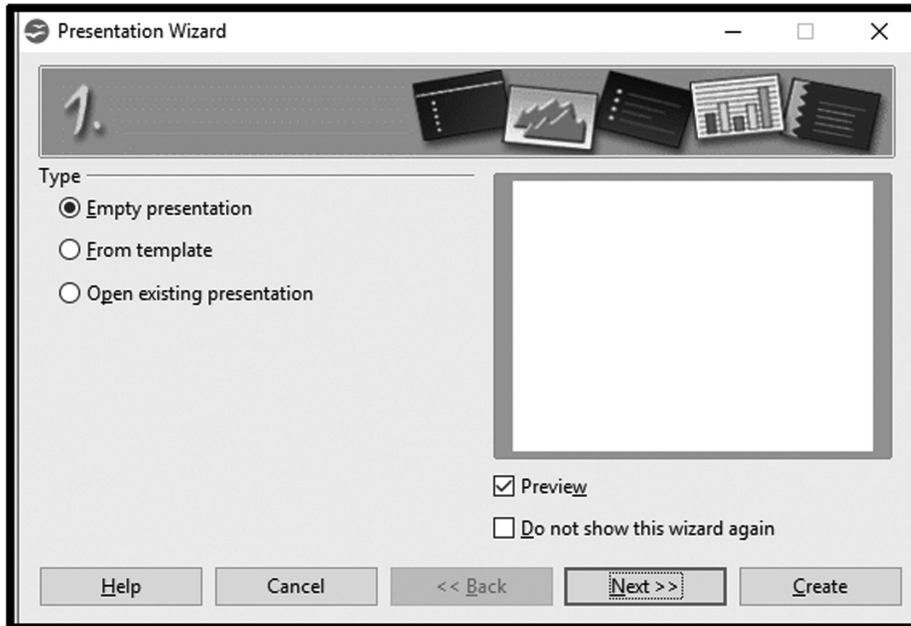


Fig. 7.18: Presentation Wizard

**Step 2:**

In this step, you need to choose a slide design and an output medium for your presentation (refer Fig. 7.19).

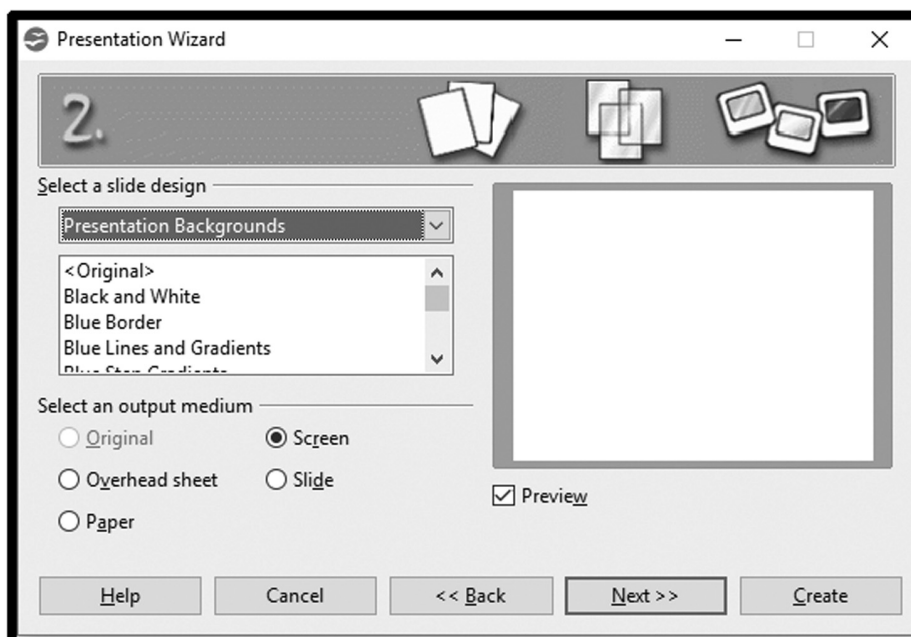


Fig. 7.19: Select Slide Design and Output Medium



Notes

**(a) Selecting Slide design:**

The Slide design section gives you two main choices. Each one has a list of choices for slide design as shown in Fig. 7.20.

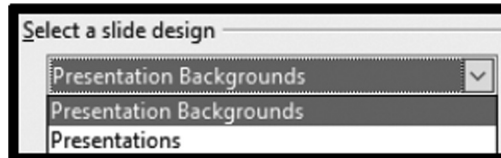


Fig. 7.20: Selecting a Slide Design

**(b) Presentation Backgrounds**

If you choose this option as the slide design, then in the second list box, you will see about 25 types of presentation backgrounds which you can select for your presentation. Out of these, '**Original**' is for an empty background (refer Fig. 7.21).

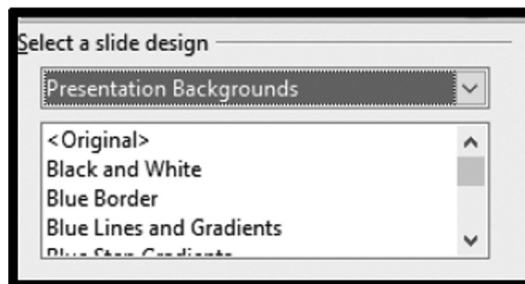


Fig. 7.21: Slide Design – Presentation Backgrounds

**(c) Presentations**

If you choose this option as the slide design, then in the second list box, you can select among three predefined presentations.

- o <Original>
- o Introducing a New Product
- o Recommendation of a Strategy

To start from scratch, you need to choose '**Original**' option.

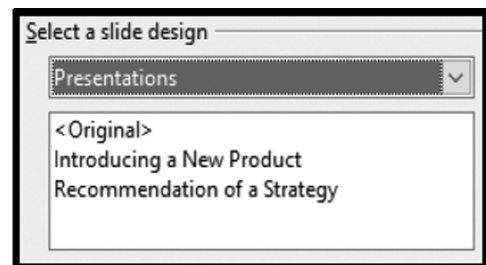


Fig. 7.22: Slide Design - Presentations

**(d) Selecting an Output medium**

You need to select an output medium for the presentation. Usually presentations have been created for computer screen. So, you would select '**Screen**' option. Once done, click '**Next**' button to go to step 3.

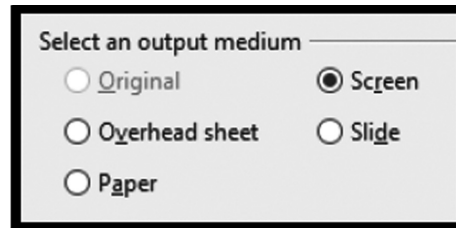


Fig. 7.23: Choose Output Medium

**Step 3:**

In Step 3, you need to decide on two options for your presentation. These are:

- (a) Slide Transition
- (b) Presentation type

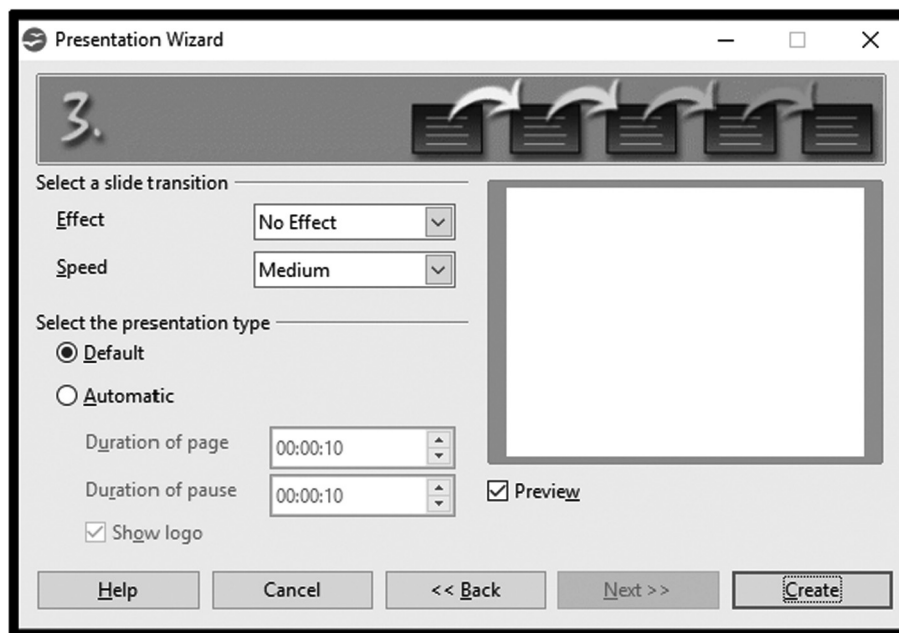


Fig. 7.24: Slide Transition and Type

**(a) Select a Slide transition:**

Under Slide Transition drop-box, there are two options. They are Effects and Speed.

The Effect option allows you to choose the transition effects for your slide.

Transition effects are the motion effects that appear during the presentation when we advance from one slide to another.



Notes

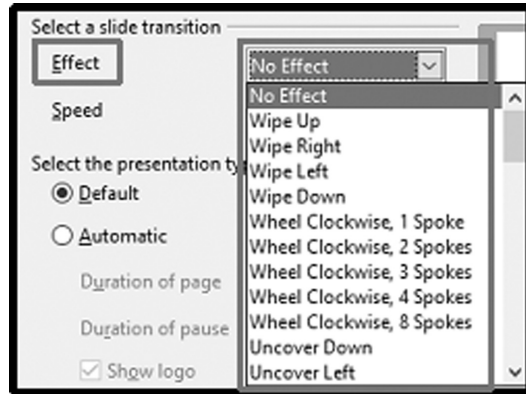


Fig. 7.25: Choose Slide Transition Effect

The Speed Option allows you to control the speed of transition effects during presentation. There are three speeds available – Slow, Medium, Fast.

**(b) Select Presentation Type**

Under ‘**Select the Presentation type**’, choose either ‘**Default**’ or ‘**Automatic**’ to define whether the presenter will manually advance slides in the presentation (Default) or whether the slides will automatically advance based on ‘**Duration of page**’ and ‘**Duration of pause**’ timing definitions.

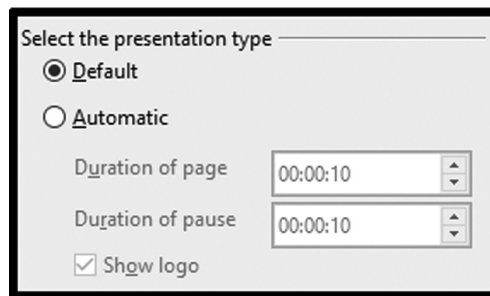


Fig. 7.26: Select Presentation Type

Once all the required options are selected, click ‘**Create**’ button and a new presentation will be created as shown below.

The first slide is normally the title slide which shows the title and subtitle of the presentation. For the specific type of content, you need to choose the correct layout which will best suit your purpose. The slide layouts are available in the Layout section of the task pane as shown in the below Fig. 7.27. To apply a specific layout to your slide, you need to:

A slide layout is an organised set of elements which can be used to create a slide.



## Notes

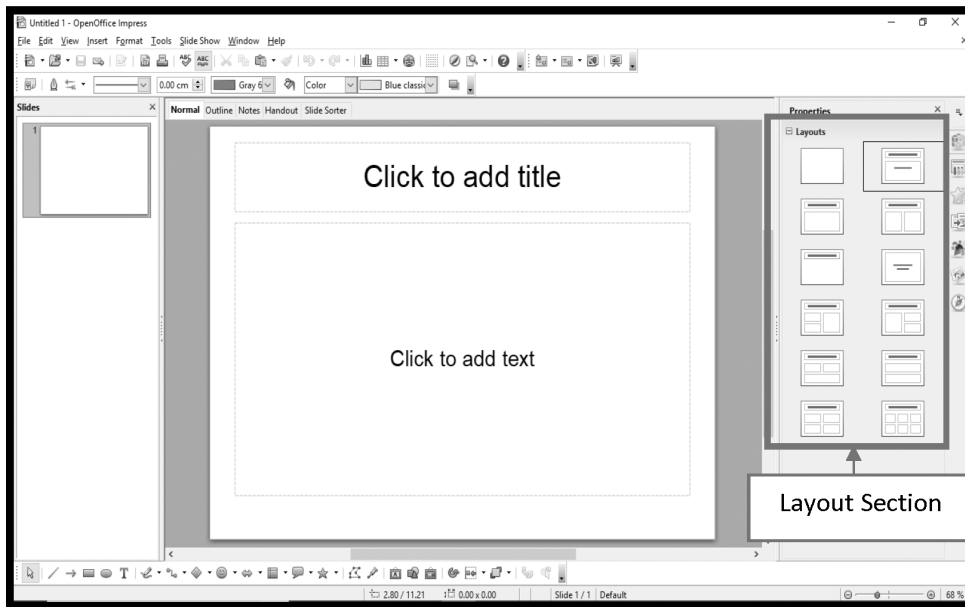


Fig. 7.27: Creating Slides Using OpenOffice Impress

- (i) Select a layout in the Layout Section of the Tasks pane by clicking on it.
- (ii) Once the layout is clicked, it will appear in the workspace.

To add title in the slide, click on the Placeholder **‘Click to add title’** and to add the content, click on the placeholder **‘Click to add text’**.”

Placeholders are boxes with dotted borders that contain content in the slides.

To insert more slides, click **‘Insert’** tab in the menu bar and choose **‘slide’** option. A new slide will be inserted below the selected slide.

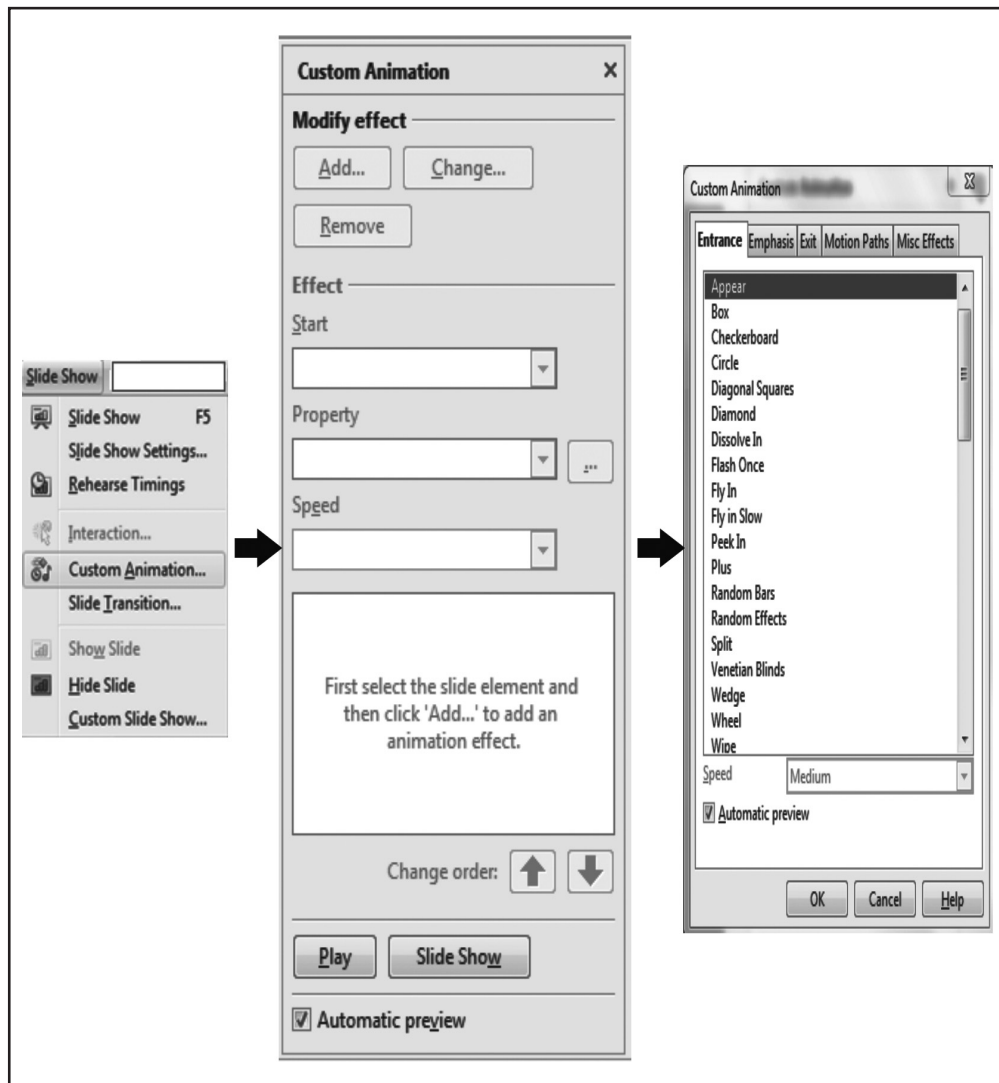
To change the layout of the slide, click on the slide in the Slide pane and click **‘Properties’** icon on the right-side of the window.

To make the presentation more interesting and exciting you need to give animation effects to the objects in the presentation. To give animation effects to the text and images, click **‘Slide Show’** option in the menu bar and click on **‘Custom Animation’**. The **‘Custom Animation’** pane appears on the right-side of the window. Click **‘Add’** button. The **‘Custom Animation’** box appears from where you can select any of the animation effects of your choice and the speed for the animation and then, click **‘OK’** button.

Custom Animation is a set of effects which can be applied to objects which enables the object to animate during presentation show.



Notes



**Fig. 7.28: Applying Custom Animation**

Now you will learn the steps to view the slides in different layouts.

Different views of the slide:

There are six ways you can view your presentation in Impress. Each new presentation opens in Normal view by default. The views can be switched by clicking at the desired view button on the top of the workspace. There are six buttons for six views. Let us discuss about each view button and also see how our slides look in each view.





## Notes

| S. No. | View Name    | Description  |
|--------|--------------|--|
| 1.     | Normal       | This is the default view where we do all the editing in the slide. This view displays three panes – Slide pane, the slide and tasks pane.  |
| 2.     | Outline      | This view displays only the text of the presentation hiding all other objects which allows you to work easily with the content.  |
| 3.     | Notes        | In this view, you can enter speaker notes which you can refer before presenting to the audience. However, these notes will be hidden during presentation.  |
| 4.     | Handout      | In this view, you can choose how many slides you want to print on a single page when you print the presentation. In Layouts pane, you can click on the choice of number of slides per page.  |
| 5.     | Slide Sorter | This view lets you see all the slides in your presentation as thumbnails. This view is used for arranging the order of slides.   |
| 6.     | Slide Show   | In this view, each slide will occupy the entire screen and will be displayed one slide at a time in sequence. To start the Slide Show, you can press 'F5' function key. To navigate from one slide to the next, you need to press 'Enter' key and to stop the Slide Show, you need to press 'Esc' key. |

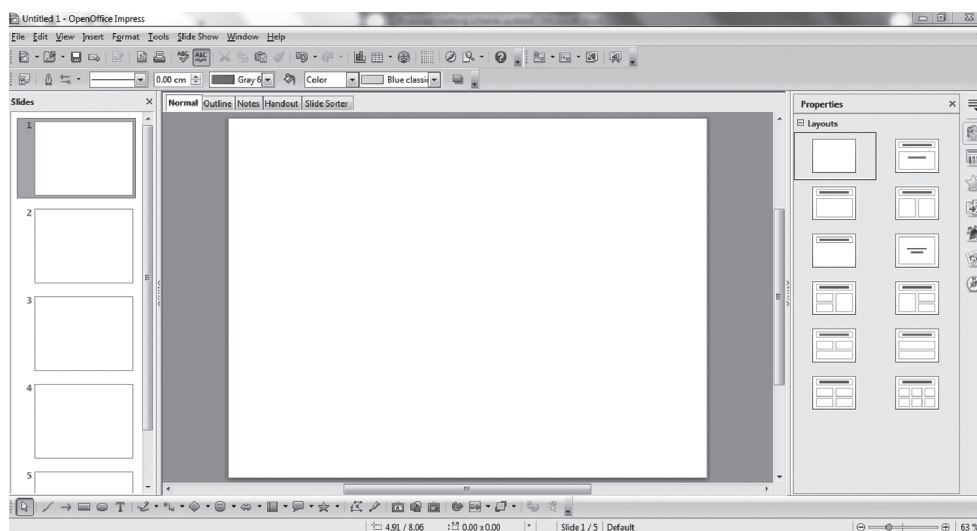


Fig. 7.29: Normal View of Slides

# MODULE-1

## Basic Computing Skills (711)



Notes

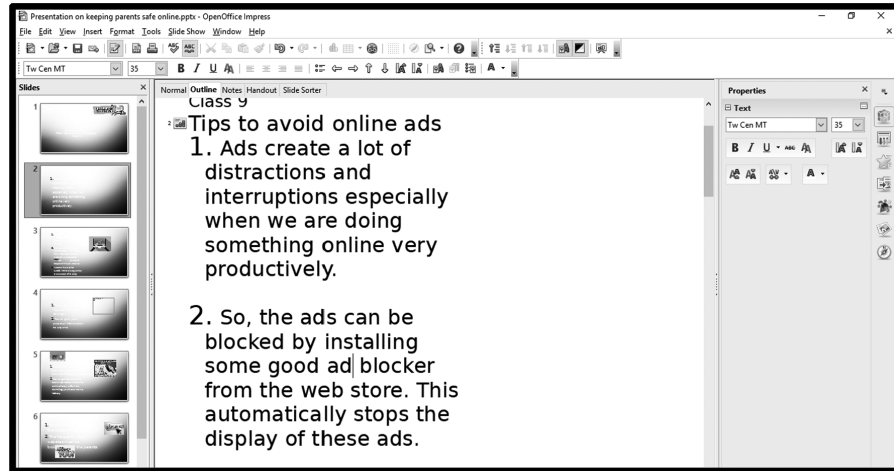


Fig. 7.30: Outline View of Slides

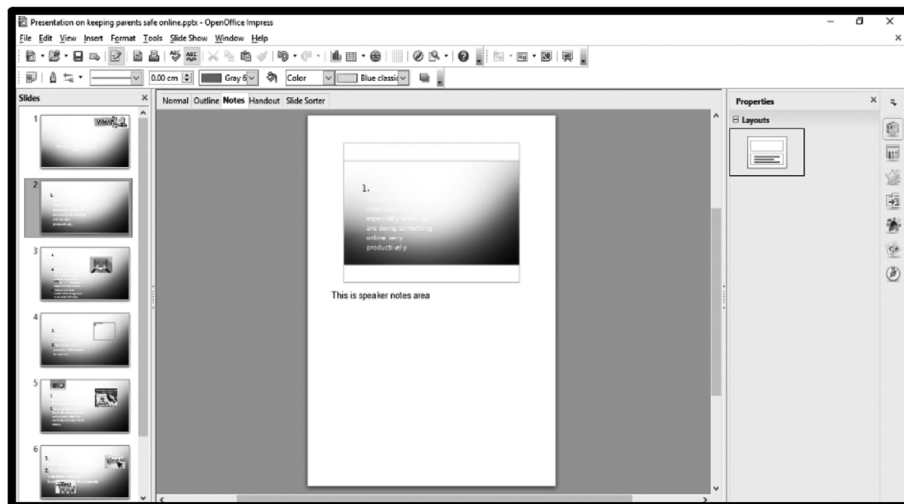


Fig. 7.31: Notes View

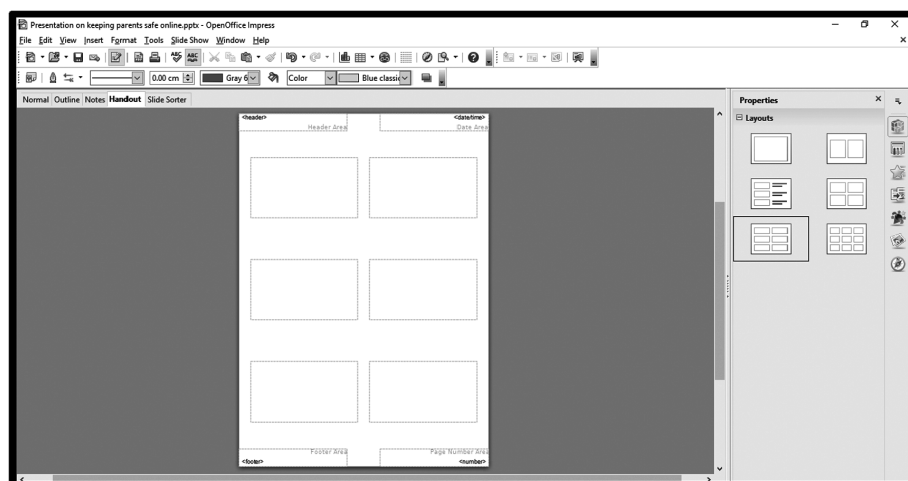


Fig. 7.32: Handout View



## Notes

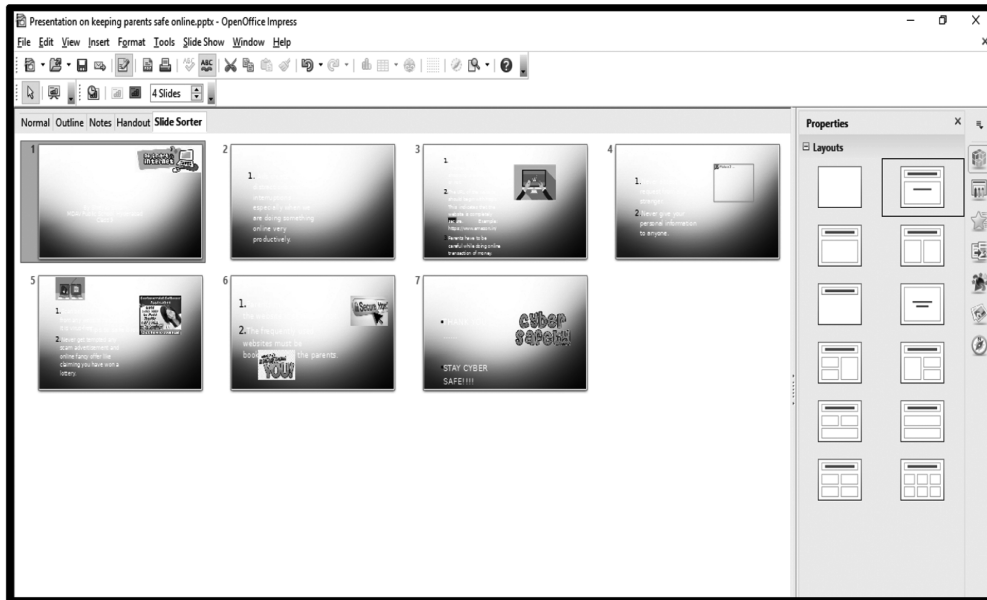


Fig. 7.33: Slide Sorter View



Fig. 7.34: Slide Show

After creating the presentation, you need to save it for later use. Now you will learn the steps to save your presentation.

### Saving the presentation

To save your presentation in the computer, click **File** in the menu bar and click on 'Save As' option. The 'Save As' dialog box appears. Choose the folder where you

# MODULE-1

Basic Computing Skills (711)



Notes

want to save your presentation and enter the name of your presentation and click ‘Save’ button. The file will be saved.

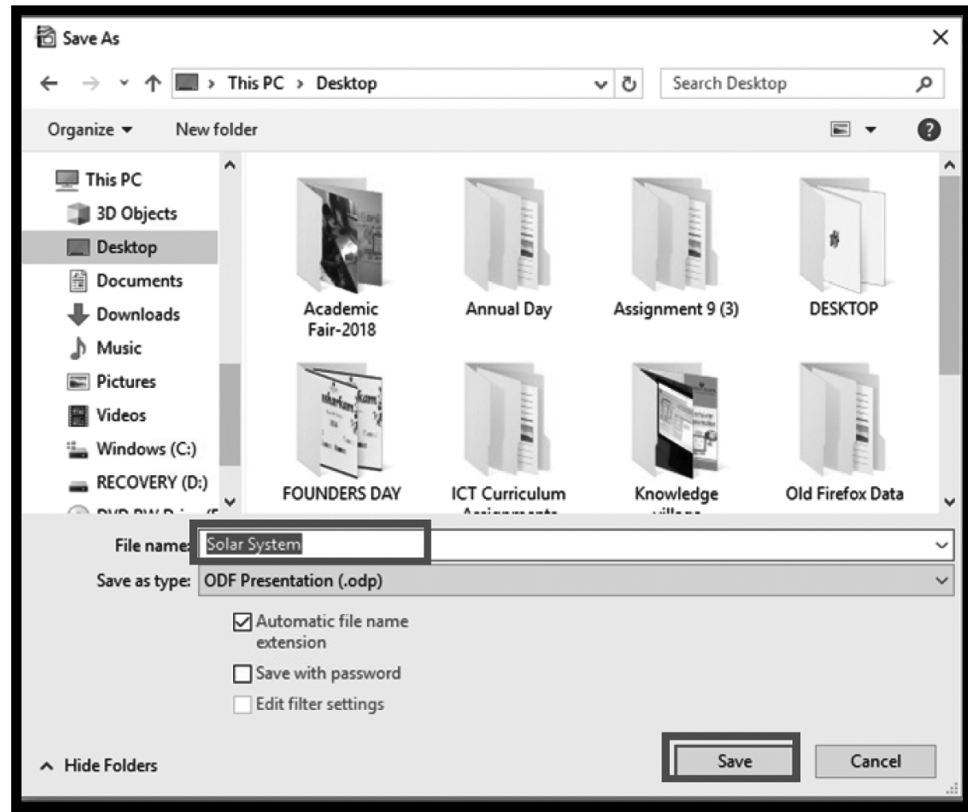


Fig. 7.35: Save The Presentation

Now you will learn few tips to make an effective presentation.

## Tips to create an effective presentation

To create an effective presentation, you need to keep certain points in mind, while preparing a presentation.

### Know your audience

Knowing your audience is very important while preparing a presentation. You must take into consideration the age group of your audience, their level of knowledge about the topic, what they expect from your presentation. Preparing your presentation based on these parameters would help you to design a structured presentation.

**Keep it simple**

Always use simple vocabulary which can be easily understandable by anyone. Be precise in what you want to convey to the audience. Use a simple font which can be easily read and use one or two font styles in your presentation. Too many font styles would cause strain to the eyes of the audience.

**Balance the information**

Too much of text or graphics will be boring for the audience. Try to strike a balance between text and graphics in your slides.

**Use animations effectively**

Use minimum animation and transition effects. Too many fast animation and transition effects would be a distraction for the audience. Focus more on the content and keep the animation and transition effects minimal.

**INTEXT QUESTIONS 7.3**

1. A ..... is a combination of different elements such as text box, images, audio, video.
2. A ..... is an organised set of elements which can be used to create a slide.
3. .... are boxes with dotted borders that contain content in the slides.
4. .... is a set of effects which can be applied to objects which enables the object to animate during presentation show.
5. To start the Slide Show, you can press ..... function key.
6. .... is a program integration technology that can be used to share information between programs through objects.
7. The area in the document window where we enter the text of the document is called .....
8. In spreadsheet, by default, the numbers are always ..... aligned and the text is always ..... aligned.
9. .... is a grid of cells made up of horizontal rows and vertical columns.
10. .... is the shortcut key for saving a file.



**Notes**



**TERMINAL EXERCISE**

1. What is the use of status bar in Open Office Writer?
2. Mention any two features of spreadsheet program.
3. Define Workbook.
4. Define Placeholders.
5. Differentiate between transition and animation effects.
6. Write any four features of Word Processing.
7. Explain five different views of Impress Presentation software.
8. Write the tips you should keep in mind while preparing a presentation.
9. Observe the following snapshot of a spreadsheet and answer the following questions.

|   | A        | B     | C | D | E |
|---|----------|-------|---|---|---|
| 1 | Subject  | Marks |   |   |   |
| 2 | English  | 94    |   |   |   |
| 3 | Math     | 78    |   |   |   |
| 4 | Science  | 92    |   |   |   |
| 5 | Computer | 85    |   |   |   |
| 6 | Social   | 89    |   |   |   |
| 7 |          |       |   |   |   |
| 8 |          |       |   |   |   |

- (i) What is the address of the active cell?  
.....
  - (ii) What is the value entered in the cell B6?  
.....
  - (iii) What is the address of the cell in which the text 'Math' is written?  
.....
10. In OpenOffice Calc, enter five subjects and the marks as shown below and find the total and average of marks.



Notes

| Subject        | Marks |
|----------------|-------|
| English        | 98    |
| Social         | 92    |
| Maths          | 100   |
| Science        | 95    |
| II Language    | 92    |
| <b>Total</b>   |       |
| <b>Average</b> |       |

11. For the following data, apply formatting effects as shown below.

| Subject        | Marks |
|----------------|-------|
| English        | 98    |
| Social         | 92    |
| Maths          | 100   |
| Science        | 95    |
| II Language    | 92    |
| <b>Total</b>   |       |
| <b>Average</b> |       |

12. Create a Presentation on the topic “Sensory Organs” using Design Template by choosing ‘From template’ option in Step 1 and follow the steps.

- The presentation should have minimum 5 slides.
- Choose appropriate layout for each slide.
- Write about one sensory organ per slide.
- Format the content in each slide.
- Insert images of the respective sensory organ in each slide.
- Give Animation and transition effects
- Save your presentation in your computer.



Notes



## ANSWERS TO INTEXT QUESTIONS

### 7.1

1. Ruler
2. Font Style
4. Ctrl+N
5. Bold and Underline

### 7.2

1. Automatic recalculation
2. Charts
3. Sorting
4. SUM
5. Cell

### 7.3

1. Slide
2. Slide layout
3. Placeholders
4. Custom Animation
5. F5
6. OLE
7. work area.
8. right, left
9. Worksheet
10. Ctrl+S



## **MODULE–2: Computer Applications (712)**

8. Introduction to Python
9. Control Statements
10. Python Lists
11. Functions in Python
12. Stacks and Queues
13. File Handling in Python
14. Errors and Exceptions
15. Database Management System
16. MySQL
17. Workplace Practices and Work Management



**8****INTRODUCTION TO PYTHON****8.1 INTRODUCTION**

As you are aware that to communicate with a computer system, instructions need to be given to it in any of the programming languages. C, C++, Java, Python etc., are few of the most popular computer programming languages. In this lesson, you will learn about Python programming language in detail. Do you know that Python is a free and open-source high-level programming language? Yes. It is an extremely simple and easy to learn programming language. It was developed by Guido van Rossum in 1989 at the National Research Institute for Mathematics and Computer Science in Netherlands and was released in the year 1991. You will be surprised to know that Python got its name from a BBC comedy series from seventies-“**Monty Python’s Flying Circus**”. In this lesson, we will discuss about the basics of Python along with various data types and operators supported by Python.

**OBJECTIVES**

After reading this lesson you will be able to:

- use interactive and script mode in Python IDLE;
- list the various data types;
- use the operators:
  - arithmetic operators
  - relational operators
  - assignment operators

## MODULE-2

Computer Applications  
(712)



Notes

- logical operators
- membership operators
- identity operators
- get an input from the user by using input statement;
- display the output by using print statement.

### 8.2 PYTHON

Python programming language is a platform independent language (means it can work on any operating systems such as Windows, Linux etc.). The latest version i.e. Python 3.x is being used in web applications, robotics, artificial intelligence, data analysis, machine learning, face detection etc. When we install Python on our computer, Python IDE (Integrated Development Environment) will be automatically installed with it. Python IDE is also called as IDLE (Integrated Development and Learning Environment).

IDLE is the standard, most popular Python development environments. To design a Python program, we basically need an editor to write code, an interpreter (executes the instruction line by line)/compiler (executes the program at one time) and a debugger to catch and remove the errors. Python IDLE provides all these tools as a bundle. You can easily write, edit, run, browse and debug Python programs from a single interface.

When you start IDLE, Python window (called the Python Shell) opens up on the computer screen as shown in Fig. 8.1.

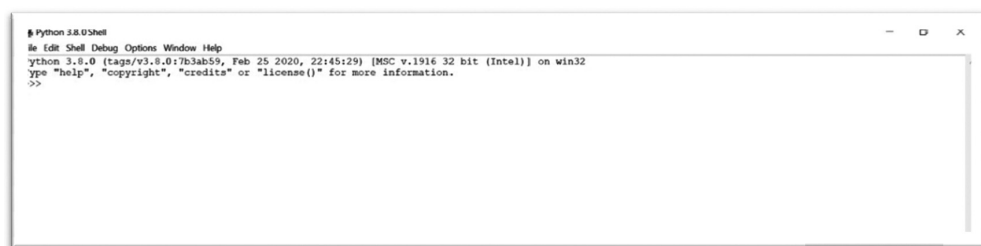
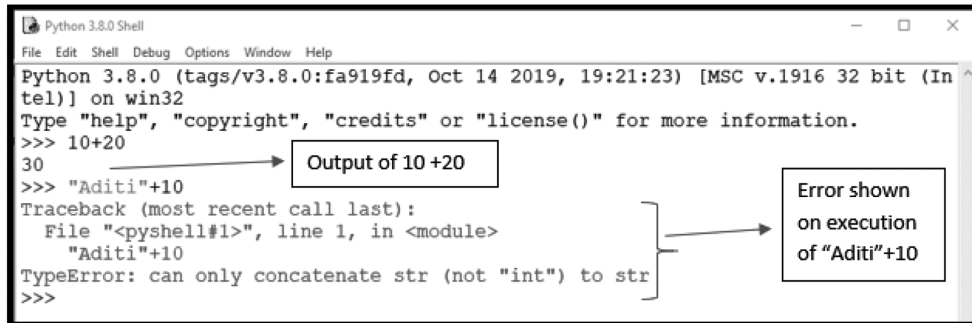


Fig 8.1: Python Shell or Interpreter

It shows you a welcome message stating its version number and a copyright notice. After this information it shows you the primary prompt (>>>) followed by a blinking cursor. It indicates that IDLE is now ready to take commands in Python programming language from the user. IDLE can be used in two modes: Interactive mode and Script mode.

In interactive mode, you can type Python expression/statement at prompt and Python immediately responds with the output of it. After successfully executing the statement, it prompts for the next command with the prompt (>>>) otherwise it lists the error encountered as shown in Fig. 8.2.



```

Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 10+20
30
>>> "Aditi"+10
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    "Aditi"+10
TypeError: can only concatenate str (not "int") to str
>>>
  
```

Output of 10 +20

Error shown on execution of "Aditi"+10

Fig. 8.2: Interactive Mode

Python editor, as shown in Fig. 8.3, allows you to work in script mode i.e., you can create and edit Python source file. In script mode you can write a block of code.

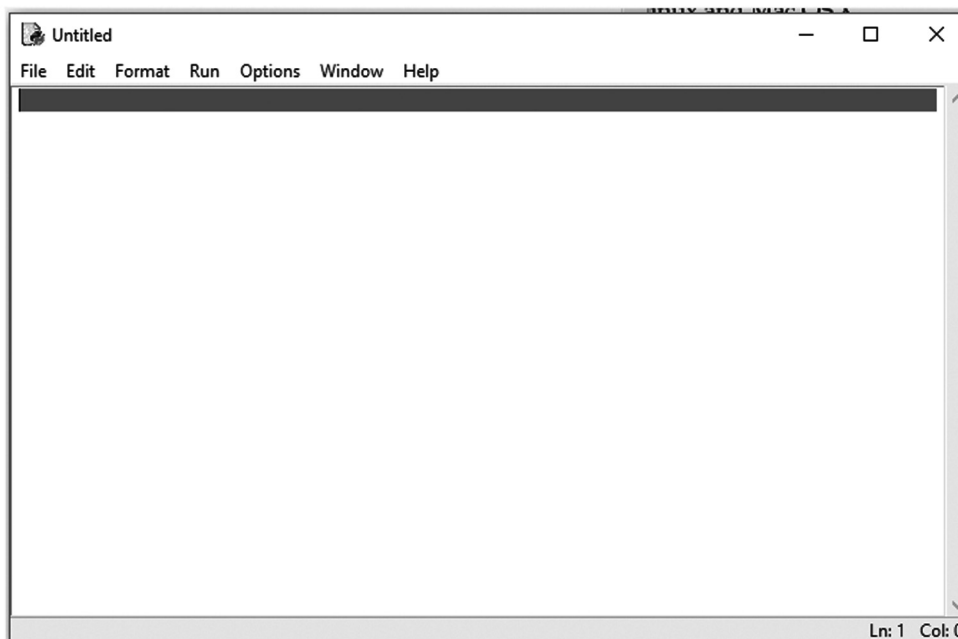


Fig. 8.3: Python Editor

To open the Python script mode, go to **'File' menu --> New File'** as shown in Fig 8.4. A new window will be opened. Write your Python code here. Save the file by clicking **'File --> Save'**. The Python files are stored with extension **.py**. By default, all Python files are stored in Python installation folder.



Notes

## MODULE-2

Computer Applications  
(712)



Notes

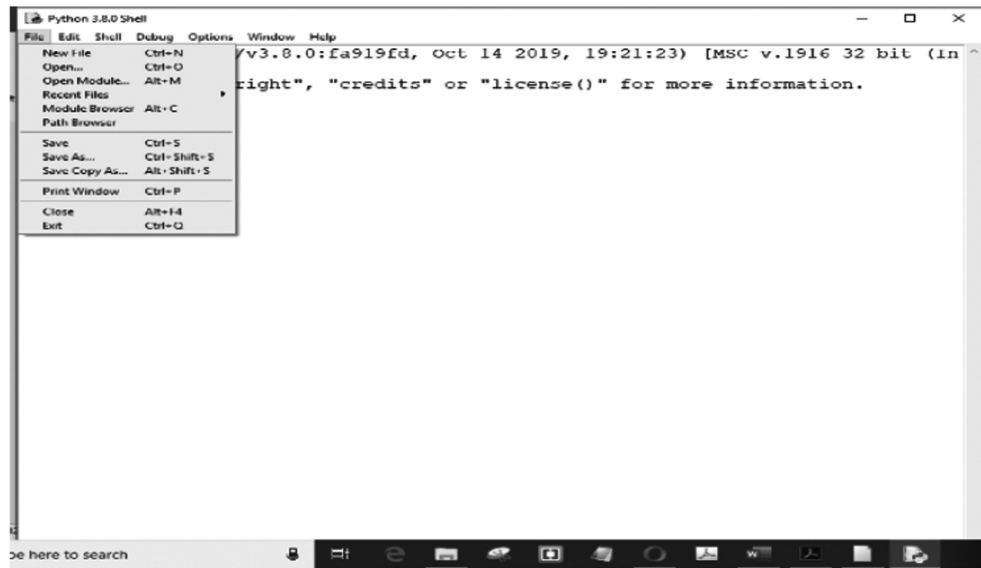


Fig. 8.4: Steps to write a Python script

### Running/Executing a Python Script/Program

You can run your Python program by clicking 'Run' → 'Run Module' or pressing **F5** as shown in Fig. 8.5.



Fig. 8.5: Executing Python Script

*Ctrl+D* or *quit()* is used to leave the interpreter and *Ctrl+F6* is used to restart the shell.

The output of your program will appear on shell as shown in the Fig. 8.6.



Fig. 8.6: Output of Python script

### 8.3 TOKENS

The tokens are the smallest individual unit of a program. The following are the tokens in Python programming language:

**‘keywords’, ‘identifiers’, ‘literals’, ‘operators’, and ‘delimiters’.** Let us discuss them one by one.

The ‘#’ symbol is used to insert comments in a program and it is not an identifier.

#### Keywords

These are the reserved words used by Python interpreter. These words have specific meaning for interpreter. You can get a list of keywords by typing the following commands at the prompt:

```
>>>import keyword
```

```
>>>keyword.kwlist
```

```
[‘False’, ‘None’, ‘True’, ‘and’, ‘as’, ‘assert’, ‘break’, ‘class’, ‘continue’, ‘def’, ‘del’, ‘elif’, ‘else’, ‘except’, ‘finally’, ‘for’, ‘from’, ‘global’, ‘if’, ‘import’, ‘in’, ‘is’, ‘lambda’, ‘nonlocal’, ‘not’, ‘or’, ‘pass’, ‘raise’, ‘return’, ‘try’, ‘while’, ‘with’, ‘yield’]
```

**Literals:** A fixed numeric or non-numeric value is called as literal. Examples of literals are: 502, -178.76, “Rajan” etc.

**Operators:** The operators perform some action on the data. Examples of operators are: +, -, \*\*, /, // etc.

**Delimiters:** These are the symbols which can be used as separators of values, or to enclose some values. Examples of delimiters are: ( ) { } [ ] , ; :



Notes

### 8.4 DATA TYPES

Python consists of various standard data types that are used to define the operations possible while running a code. Data type states the type of the values that are stored, the operations that can be performed on that type, and the range for that type. Various data types offered by Python are listed as follows in Fig 8.7.

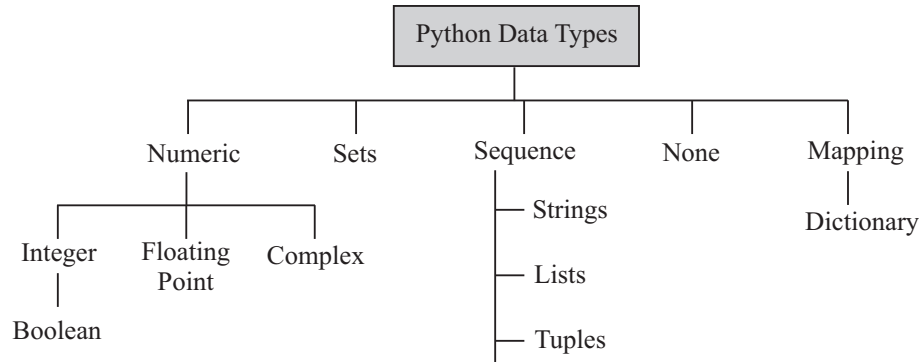


Fig. 8.7: Python Data Types

#### 8.4.1 Numeric

Numeric data type is used to store numerical values. Python supports three different numerical types:

- Integer
- Floating point numbers
- Complex numbers

In addition, **Booleans are a subtype of integers.**

(a) **Integers** are whole numbers without decimal point. These can be positive or negative numbers with unlimited length. Integers also contain Boolean type which is a unique data type, consisting of two constants, True and False. A Boolean True value is Non-Zero, Non-Null and Non-empty.

#### Example 1

| Python Statements |
|-------------------|
| >>>a=20           |
| >>>entry=True     |

Here the value 20 is assigned to the variable a, the value True is assigned to the variable entry.





Notes

- (b) Numbers with fractions or decimal point are called **floating point numbers**. These can be positive or negative with unlimited length.

**Example 2****Python Statements**

```
>>>k=200.789
```

The value 200.789 is assigned to the variable k.

- (c) A **complex number** is an ordered pair of two floating-point numbers denoted by  $a + bj$ , where a and b are the real numbers and j is an imaginary unit.

**Example 3****Python Statements**

```
>>>x=10+5j
```

For accessing different parts of variable x, you can use *x.real* (which is 10) and *x.imag* (which is 5).

**Example 4**

| Python Statements               | Output |
|---------------------------------|--------|
| <pre>&gt;&gt;&gt; x=10+9j</pre> |        |
| <pre>&gt;&gt;&gt; x.real</pre>  | 10.0   |
| <pre>&gt;&gt;&gt; x.imag</pre>  | 9.0    |

**8.4.2 Sets**

The set is an unordered collection of comma separated values enclosed in { }.

**Example 5****Python Statements**

```
>>>s1={12,10,34,23,78}
```

Here s1 is a set which contains the values 12, 10, 34, 23 and 78. All the values of a set are unique. It eliminates duplicate values.

**Example 6**

| Python Statements                           | Output       |
|---|--------------|
| <pre>&gt;&gt;&gt; s2={12,12,15,10,10}</pre> |              |
| <pre>&gt;&gt;&gt; s2</pre>                  | {10, 12, 15} |



Notes

Here it has eliminated the repeated values of 12 and 10 and displayed the values as {10, 12, 15} which is shown in the output column.

**8.4.3 Sequence**

A sequence in Python is an ordered collection of items indexed by integers starting from 0. There are three types of sequence data types available in Python:

- (a) Strings
- (b) Lists
- (c) Tuples

**(a) Strings**

A string is an ordered set of characters enclosed in single or double quotation marks. Strings are immutable (unable to be changed) i.e., the value once assigned to these variables, cannot be changed.

**Example 7**

| Python Statements      |
|------------------------|
| >>>str="I love Python" |

Here the value "I love Python" will be assigned to string str.

**(b) Lists**

A list is an ordered collection of comma-separated values (items) enclosed in square brackets. The values in the list can be modified, i.e., it is mutable. The values that make up a list are called its 'elements'. The elements in a list need not to be the same type.

**Example 8**

| Python Statements                        |
|--|
| list1=[100, 200, 300, 400, 500]          |
| list2=["Raman", 100, 200, 300, "Ashwin"] |
| list3=['A', 'E', 'I', 'O', 'U']          |

In the example 8, the list1 contains 100, 200, 300, 400 and 500 values. The list2 contains Raman, 100, 200, 300, Ashwin and list 3 has A, E, I, O, U.



## Notes

**(c) Tuple**

A tuple is an ordered collection of comma separated values. The values in the tuple cannot be modified, i.e., it is immutable. The values that make up a tuple are called its elements. The elements in a tuple need not be of the same type. The comma separated values can be enclosed in parenthesis but parenthesis is not mandatory.

All numeric (integer, float, complex), strings, tuples and sets data types are immutable in Python while lists and dictionary are mutable data types.

**Example 9**

| Python Statements                 |
|-----------------------------------|
| tup1=('Sunday', 'Monday', 10, 20) |
| tup2=('a', 'b', 'c', 'd', 'e')    |
| tup3=(1,2,3,4,5,6,7,8,9)          |

The tuple tup1 has Sunday, Monday, 10 and 20 elements. i.e., it has both character values and numerical values. The tup2 has a, b, c, d and e (character values) and tup3 has 1,2,3,4,5,6,7,8, and 9 (numerical values).

**8.4.4 None**

The None data type defines the 'null' value or 'no' value. It is neither 'False' nor '0'.

**Example 10**

| Python Statements | Output |
|-------------------|--------|
| >>> x=None        |        |
| >>> print(x)      | None   |

**8.4.5 Mapping**

It is an unordered and mutable data type in Python. Dictionaries fall under this category.

**Dictionary**

Python dictionary is an unordered collection of items where each item is a key: value pair. You can also refer to a dictionary as a mapping between a set of keys/ indices and a set of values. Each key is separated from its value by a colon (:), the



Notes

items are separated by commas, and the entire dictionary is enclosed in curly braces. Dictionary is mutable. You can add new items or change the value of existing items.

**Example 11**

**Python Statements**

```
dict1={'num':10, 1:20}
```

In the above example 'num' and '1' are keys and 10 and 20 are values.



**INTEXT QUESTIONS 8.1**

1. Mention the data type of the following variables:
  - (a) x= 'Raghav'
  - (b) y= 100.56
  - (c) z=10
  - (d) t=15 +6j
  - (e) var1=[120, 130, 140, 'Ajit']
  - (f) var2=('May', 'June', 5, 6)
2. Determine the data type of each of the following:
 

|          |            |
|----------|------------|
| (a) 42   | (b) 4.2    |
| (c) '42' | (d) (4, 2) |

**8.5 VARIABLES**

Variables are like containers which hold data/information in a computer memory. These are used to store the values for some input, intermediate result of calculations or the final result of a mathematical operation. The characteristics of a variable are:

- It has a name.
- It is capable of storing values of certain type.
- It provides temporary storage in RAM.



## Notes

## Variable Naming Conventions

- The variable names are case sensitive. For example, num and NUM are treated as two different variable names.
- All the variable names must begin with a letter or an underscore (\_). After the first initial letter, the variable names may contain letters (a-z, A-Z) / digits (0 to 9) and an underscore(\_), but no spaces or special characters are allowed.
- The keywords should not be used as the variable names.
- Variable names should be short and meaningful.

Let us see some of valid and invalid variable names.

| Examples of Valid Variable Names | Examples of Invalid Variables Names |
|----------------------------------|-------------------------------------|
| calc, salary, stu_name, _region  | avg%, 12class, break, emp-desig     |

## Assigning Values to the Variables

Python is a loosely typed language. You do not need to declare variables with any particular type. The variables will be declared automatically when the values are assigned to it. The assignment operator (=) is used to assign value to a variable.

## Example 12

| Python Statements | Explanation   |
|-------------------|---|
| x=5               | The value 5 is mapped/assigned to the variable x.       |
| design='Manager'  | The value 'Manager' is assigned to the variable design. |
| marks=90.5        | The value 90.5 is assigned to the variable marks.       |

This happens because a variable stores the address of a memory location. For example, the statement x=5 will make the variable x point to the memory location where the value 5 is stored. You can also say that the variable x refers to the value 5.

In the next statement suppose the variable x is assigned to the value “**Strong**”. Now the variable x will point to the memory location where the value “**Strong**” is stored. You can now say that the variable x refers to the value “**Strong**” (refer Fig. 8.8).

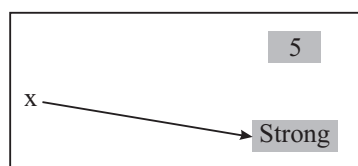


Fig. 8.8: The Variable x Refers to the New Value



Notes

In Python, the variables have no data type. When you say data type of a variable, you are actually referring to the data type of the value that it represents. As a variable does not have any fixed data type its data type keeps changing depending upon the value it represents. You can use the type() method to know the data type of a variable.

**Example 13**

| Python Statements             | Output        |
|-------------------------------|---------------|
| >>>x=10<br>>>> type(x)        | <class 'int'> |
| >>>x=" Strong"<br>>>> type(x) | <class 'str'> |

In Python, you can assign the same value to multiple variables simultaneously. Also multiple values can be assigned to several variables in a single statement.

**Example 14**

| Python Code                 | Explanation   |
|-----------------------------|---|
| >>>a=b=c=100                | This statement will assign the value 100 to the variables a, b and c.   |
| >>>a,b,c=22, 80.5, "Tarang" | This statement will declare the variables a, b and c and assigns the values 22, 80.5 and "Tarang" respectively. i.e., a=22, b=80.5 and c="Tarang" |



**INTEXT QUESTIONS 8.2**

1. Identify the valid and invalid variable name for the following:

| Variable name | Valid/invalid |
|---------------|---------------|
| continue      |               |
| Break         |               |
| Maths-marks   |               |
| _sales        |               |
| annual_exam   |               |
| student.name  |               |



Notes

2. Identify which of the following expressions are correct:
- (a) roll\_no, name, per=10, 'Sarthak', 90.5
  - (b) emp\_no, salary, designation=120, 170000, 'Manager', 60
  - (c) q,w,e=200
  - (d) num1, num2=100, num3=200
  - (e) var1=var2=50

## 8.6 COMMENTS

These are non-executable statements which are ignored by Python interpreter. The comments are used to make the code more readable and understandable. Generally, the programmer use comments for mentioning the purpose of code/program written. In Python, comment start with '#' symbol and extends till the end of the line. Anything written after # in a line is ignored by the interpreter, i.e., it will not have any effect on the program.

### Example 15

#### Python code

```
# This program calculates the sum of two numbers

>>>n1=10

>>>n2=20

>>>sum= n1 + n2 # sum of numbers 10 and 20

>>>print ("sum=", sum)
```

For adding multi-line comment in a program, we can:

- Place '#' in front of each line.
- or
- Use triple quoted string. They will only work as comment, when they are not being used as docstring. (A docstring is the first statement in a class/function/module. You will learn about function in lesson no. 10).



Notes

**Example 16**

```

Python Code
""This is a
    multiline comment""
""""This is a
    multiline comment""""
    
```

**8.7 OPERATORS**

Python provides six different types of operators: These are:

- Arithmetic operator
- Assignment operator
- Membership operator
- Relational Operator
- Logical operator
- Identity operator

Let us learn about them in detail

**8.7.1 Arithmetic Operators**

These operators perform arithmetic operations on operands. The list of arithmetic operators is given in Table 8.1.

**Table 8.1: Arithmetic Operators**

| Operator | Description                               | Example                                |
|----------|---|--|
| +        | Add two operands.                         | 10+5=15                                |
| -        | Subtracts two operands.                   | 10-5=5                                 |
| *        | Multiplies two operands.                  | 10*5=50                                |
| /        | Divides two operands.                     | 10/5=2.0<br>10/3.0 = 3.3<br>10.0/3=3.3 |
| %        | Divides two operands and gives remainder. | 10%5=0<br>10%3=1                       |
| **       | Exponentiation.                           | 2**4= 16                               |
| //       | Integer division.                         | 5//2=2<br>5.0//2=2.0                   |





Notes

### 8.7.2 Relational/Comparison Operator

These operators describe a relationship between two operands. The relational operators are given in Table 8.2.

**Table 8.2: Relational Operators**

| Operator | Description              | Examples                         |
|----------|--------------------------|----------------------------------|
| <        | less than                | 10<15, true                      |
| >        | greater than             | 15>10, true                      |
| <=       | less than or equal to    | 15<=15, true<br>16<=15, false    |
| >=       | greater than or equal to | 16>=15, true<br>14>=15, false    |
| ==       | equal to                 | 15== 15, true<br>16 == 15, false |
| <>       | not equal to             | 15 <> 15, false<br>14 <>15, true |

### 8.7.3 Assignment Operator

The operator '=' stores the value of the expression on the right-hand side of the equal sign to the operand on the left-hand side. In addition to the standard assignment operator shown above, Python also supports compound assignment operators. The assignment operators supported by Python are given in Table 8.3.

**Table 8.3: Assignment Operators**

| Operator | Description                                      | Example                             |
|----------|--|-------------------------------------|
| =        | Assign the value to the variable.                | A = 10                              |
| +=       | Add and assign the value to the variable.        | A+=10 is equivalent to<br>A =A +10  |
| -=       | Subtract and assign the value to the variable.   | A-=10 is equivalent to<br>A =A -10  |
| *=       | Multiply and assign the value to the variable.   | A*=10 is equivalent to<br>A =A *10  |
| /=       | Divide and assign the value to the variable.     | A/=10 is equivalent to<br>A =A /10  |
| %=       | Divide and assign the remainder to the variable. | A%=10 is equivalent to<br>A =A % 10 |



Notes

|            |   |                                   |
|------------|---|-----------------------------------|
| <b>**=</b> | Perform an exponential operation and assign the value to the variable.        | A **=2 is equivalent to<br>A=A**2 |
| <b>//=</b> | Perform an integer division on operands and assign the value to the variable. | A//=2 is equivalent to<br>A=A//2  |

**8.7.4 Logical Operators**

These operators are used to join one or more relational expressions. The logical operators are given in Table 8.4.

**Table 8.4: Logical Operators**

| Operator   | Description   | Example   |
|------------|---|---|
| <b>not</b> | NOT operator. It is a unary operator and it reverses the logical state of its operand. If a condition is true, then logical NOT operator will make it false and vice versa. | >>>not(5>10)<br>true  |
| <b>or</b>  | OR operator. It returns true if any of the conditions joined by this operator is true. It returns false when both the conditions are false.                                 | >>> (10<5) or (15>20)<br>false<br>>>> (10>5) or (15>20)<br>true<br>>>> (10<5) or (15<20)<br>true<br>>>> (10>5) or (15<20)<br>true         |
| <b>and</b> | AND Operator. It returns true if both the conditions joined by this operator are true. It returns false when any of the conditions is false.                                | >>> (20>5 ) and (15>10)<br>true<br>>>> (20<5 ) and (15>10)<br>false<br>>>>(20>5 ) and (15<10)<br>false<br>>>>(20<5 ) and (15<10)<br>false |



### 8.7.5 Identity Operators

These operators are used either to check the type of data contained in a variable or to compare the memory location of two variables. The list of identity operators are given in Table 8.5.

**Table 8.5: Identity Operators**

| Operator      | Description   | Example  | Explanation   |
|---------------|---|--|---|
| <b>is</b>     | Returns true if the two variables on either side of the operator point towards the same memory location otherwise it returns False. Note: The memory location of a variable can be determined by the function- id() | <pre>&gt;&gt;&gt; a=10 &gt;&gt;&gt; b=20 &gt;&gt;&gt; id(a) 1452001344 &gt;&gt;&gt; id(b) 1452001504 &gt;&gt;&gt; a is b False &gt;&gt;&gt; b=a &gt;&gt;&gt; id(a) 1452001344 &gt;&gt;&gt; id(b) 1452001344 &gt;&gt;&gt; a is b True</pre> | <p>Here a and b are two variables with values 10 and 20 respectively. As you can see the memory locations of two variables are different thus the statement '<i>a is b</i>' returns False.</p> <p>Now when the statement <i>b=a</i> is executed both <i>a</i> and <i>b</i> point to the same memory location. Thus, the statement '<i>a is b</i>' returns True.</p> |
|               | It is also used to check the type of data contained in a variable.  | <pre>&gt;&gt;&gt; type(a) is int True</pre>  | As a contains int type data thus the statement <i>type(a) is int</i> returns True.  |
| <b>is not</b> | Returns true if the two variables on either side of the operator does not point towards the same memory location otherwise it returns false.  | <pre>&gt;&gt;&gt; a=10 &gt;&gt;&gt; b=20 &gt;&gt;&gt; a is not b True</pre>  | It returns true as a and b are not pointing towards the same memory location.   |
|               | It is also used to check the type of data contained in a variable   | <pre>&gt;&gt;&gt; a="Hello" &gt;&gt;&gt; type(a) is not int True</pre>   | It returns true because a is not integer data type. It is str type.   |

Notes



Notes

**8.7.6 Membership Operators**

These are used to check whether a specified value/variable is present in a given sequence such as strings, lists, or tuples. The list of membership operators are given in Table 8.6.

**Table 8.6 Membership Operators**

| Operator      | Description   | Example   |
|---------------|---|---|
| <b>in</b>     | Returns true if the specified value/variable is present in a given sequence such as strings, lists, or tuples otherwise it returns false.     | <pre>&gt;&gt;&gt;str="I am learning Python" &gt;&gt;&gt; "Python" in str True &gt;&gt;&gt; "learN" in str False</pre> |
| <b>not in</b> | Returns true if the specified value/variable is NOT present in a given sequence such as strings, lists, or tuples otherwise it returns false. | <pre>&gt;&gt;&gt; l1=[10,20,30] &gt;&gt;&gt; 10 not in l1 False &gt;&gt;&gt; 50 not in l1 True</pre>                  |

**8.7.7 Precedence of Operators**

Python follows the same precedence rules for its mathematical operators that mathematics does (refer Table 8.7). The acronym **PEMDAS** (**P**arentheses, **E**xponentiation, **M**ultiplication/**D**ivision, **A**ddition/**S**ubtraction) is a useful way to remember the order of operations:

**Table 8.7: Precedence of Operators**

|   |                       |
|---|-----------------------|
| *(Exponentiation)<br>-(unary minus), +(unary plus)<br>*(multiply), / (divide), %(modulus), //(floor division)<br>+(add), -(subtract)<br><(less than), <=(less than or equal), >(greater than), >=(greater than or equal to)<br>==(equal), !=(not equal)<br>=(simple assignment) and other assignment operators (arithmetic assignment operator)<br>is, is not (identity operator)<br>in, not in (Membership operators)<br>not (logical NOT), and (logical AND), or (logical OR) | Highest<br><br>Lowest |
|---|-----------------------|



For the operators at the same precedence, the left-to-right rule will be applied. The expression within parenthesis is always evaluated first.

## 8.8 INPUT/OUTPUT

Python supports input/output statements which can be used to feed new data into the computer's memory with the help of input devices such as: keyboard or obtain the output on an output devices such as: monitor, printer etc.

### Print Statement

This statement is used to display a literal or a variable on the monitor. If an expression is given with the print statement, it first evaluates the expression and then print it. To print more than one item on a single line, comma (,) may be used. By default, print uses a single space i.e., ' ' as a separator and a '\n' as a terminator (appears at the end).

### Example 17

| Python Code   | Output           |
|---|------------------|
| <code>&gt;&gt;&gt;print( "How are you?")</code>   | How are you?     |
| <code>&gt;&gt;&gt;msg= "Today is Tuesday"<br/>&gt;&gt;&gt;print (msg)</code>            | Today is Tuesday |
| <code>&gt;&gt;&gt;x="2019"<br/>&gt;&gt;&gt;y="Year"<br/>&gt;&gt;&gt;print(y + x)</code> | Year2019         |
| <code>&gt;&gt;&gt;print('a', 'b', 'c', 'd')</code>                                      | a b c d          |
| <code>&gt;&gt;&gt;print('a', 'b', 'c', 'd', sep='#', end='*')</code>                    | a#b#c#d*         |

The print statement is used to print a blank line when no arguments are passed to it.

### Input Statement

This statement is used to read text (strings) from the user.

### Syntax

#### **input ([prompt])**

where prompt is a message you wish to display on the screen. It is optional. During the execution, input() shows the prompt to the user and waits for the user to input



Notes

a value from the keyboard. When the user enters the value from the keyboard, `input()` returns that value which is usually stored in a variable. For example:

### Example 18

| Python Code   | Output   | Explanation  |
|---|--|--|
| <pre>&gt;&gt;&gt;rno = input ('Enter your roll number ') &gt;&gt;&gt; type(rno)</pre> | <pre>Enter your roll number 10 &lt;class 'str'&gt;</pre> | <p>After the execution of this input statement, 10 will be stored in the variable rno or you can say rno will refer to 10.</p> <p>The input function takes text input and stores it in a variable present on left side of the assignment operator. Here, 10 is stored as string in the variable rno.</p> |

You cannot perform arithmetic operations on string data. Thus, string data needs to be converted into numeric data. The text input taken by the input function can easily be converted into appropriate numeric data type by using `int()`, `float()` and `eval()` functions. The `int()` and `float()` functions take a number, expression, or a string as an argument and return the corresponding integer and float value respectively. The function `eval()` takes a string as an argument, evaluates this string as a number (it can be integer or float), and returns the numeric result.

### Example 19

| Python Code   | Output                             | Explanation  |
|---|------------------------------------|--|
| <pre>&gt;&gt;&gt;y=int (input('enter your marks'))</pre>          | <pre>enter your marks 98</pre>     | <p>Data entered by the user is converted from string to integer and then assigned to the variable y.</p>   |
| <pre>&gt;&gt;&gt;maths=eval (input('enter marks in Maths'))</pre> | <pre>enter marks in Maths 78</pre> | <p>Data entered by the user is evaluated and returned as a numeric result. In this case, <code>eval()</code> evaluates string 78 as a number and returns 78 as numeric result.</p> |

You can use `str()` and `chr()` functions to convert the given data to string and character respectively.



Notes

## 8.9 EXPRESSIONS

An expression is a combination of values, variables, operators etc. When an expression is typed at the Python prompt, the interpreter evaluates it and displays the result.

### Example 21

| Python Expressions            |
|-------------------------------|
| $v = u * t + 1/2 * a * t * t$ |
| $f = (c * 9/5) + 32$          |

### Example 22

The following Python script gets the input of marks from the user for Mathematics, Science and Social Science subjects and then calculates total and percentage.

| Python Code   | Output                           |
|---|----------------------------------|
| <code>a=eval(input('enter marks in Maths'))</code>          | enter marks in Maths 67          |
| <code>b=eval(input('enter marks in Science'))</code>        | enter marks in Science 89        |
| <code>c=eval(input('enter marks in Social Science'))</code> | enter marks in Social Science 96 |
| <code>total=a+b+c</code>                                    | Total= 252                       |
| <code>per=total/3</code>                                    | Percentage= 84                   |
| <code>print("Total=", total)</code>                         |                                  |
| <code>print("Percentage=", per)</code>                      |                                  |

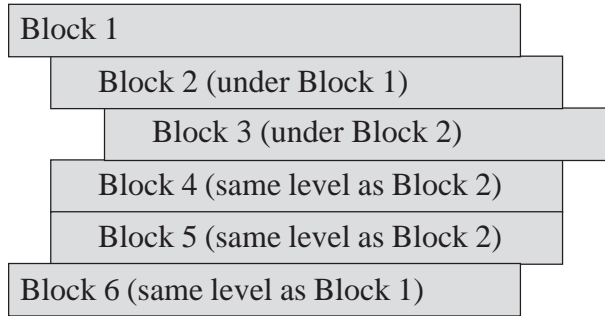
Here Python script mode was used for writing the above Python script.

## 8.10 INDENTATION

Indentation in Python is used to form block of statements also known as 'suite'. In most other programming languages, indentation is used only to beautify the code. But in Python, it is required to put a group of statements in a block. Indentation is typically used in flow of control statements (if, for etc.), function, class etc. Although you can give any number of spaces for indentation but all the statements in a block must have the same indent.



Notes



**Note:**

- The amount of indentation matters in Python. A missing or extra space in a Python block could result in an error or unpredictable results.
- It is important to indent the statements within the same block of code at the same level.



**INTEXT QUESTIONS 8.3**

1. Write a Python expression for the following:

(i) 
$$\frac{2AB+2BC+2CA}{2A}$$

(ii) 
$$\frac{14x^2}{3}$$

(iii) 
$$\frac{b^2 - 4ac}{2a}$$

2. Evaluate the following expressions:

(i)  $85\%2+70-3**3+15$

(ii)  $5//2*5+8-3**2$

(iii)  $5/2*5+8-3**2$

(iv)  $\text{not}(5>10) \text{ or } (10 == 10) \text{ and } (\text{not}(9<15))$



**WHAT YOU HAVE LEARNT**

- Python is a free and open source high-level language which is platform independent, and has a vast library of add-on modules.
- Python IDLE can be used in two modes: Interactive mode and Script mode.
- Tokens are the smallest individual unit of a program. The keywords, identifiers, literals, operators, and delimiters are Python tokens.
- Keywords have specific meaning for interpreter and they cannot be used for any other purpose.
- Various data types offered by Python are: numeric, sets, sequence, none and mapping.
- All numeric (integer, float, complex), strings, tuples and sets data types are immutable in Python while lists and dictionary are mutable data types.
- Variables are like containers which hold data/information in a computer memory.
- The id() method returns the address of memory location to which a variable is referring and the type() method returns the data type of a variable or literal passed to it.
- Python provides six different types of operators: arithmetic, relational, assignment, logical, membership and identity.
- The print statement is used to display a literal or a variable on monitor and the input statement is used to read text (strings) from the user.

**TERMINAL EXERCISE**

1. What is a keyword?
2. What is the difference between a literal and an identifier?
3. What is a difference between a tuple and a list?
4. Write Python program to input a number and print its square.
5. Write Python program to input number of pencils purchased and cost of one pencil and print the total amount to be paid.
6. Write a program to input length and breadth of a rectangle and print its area and perimeter.

**Notes**



Notes



**ANSWERS TO INTEXT QUESTIONS**

**8.1**

1. (a) string (b) floating point (c) integer  
(d) complex number (e) list (f) tuple
2. (a) integer (b) floating point (c) string  
(d) tuple

**8.2**

1. variable name                      valid/invalid  
Continue                              Invalid  
Break                                    Valid  
Maths-marks                          Invalid  
\_sales                                    Valid  
annual\_exam                          Valid  
student.name                          Invalid
2. (a) correct (b) incorrect (c) incorrect  
(d) incorrect (e) correct

**8.3**

1. (i)  $(2*A*B + 2*B*C + 2*C*A)/2*A$   
(ii)  $(14/3)* X*X$   
(iii)  $(b*b-4*a*c)/2*a$
2. (i)  $85\%2+70-3**3+15$   
 $= 85\%2 + 70 - 27 +15$   
 $= 1 + 70 - 27 +15$   
 $= 59$   
(ii)  $5//2*5+8-3**2$   
 $= 2*5+8-9$   
 $= 10+8-9$   
 $= 9$

(iii)  $5/2*5+8-3**2$   
=  $2.5*5+8-9$   
=  $12.5+8-9$   
=  $11.5$

(iv)  $\text{not}(5>10) \text{ or } (10 == 10) \text{ and } (\text{not}(9<15))$   
=  $\text{not}(\text{false}) \text{ or } (\text{true}) \text{ and } (\text{not true}))$   
=  $\text{true or true and false}$   
=  $\text{true and false}$   
=  $\text{false}$

#### Learning Outcomes

- Be proficient in using the operators in Python programming language.
- Be able to get the input from the user using input statement and get the desired result using Python Script.



## MODULE-2

Computer Applications  
(712)



Notes

# 9

## CONTROL STATEMENTS

---

### 9.1 INTRODUCTION

---

In the previous lesson, you have learnt about how to write Python statements. Generally, a computer program is set of instructions which are executed in a sequential order (i.e., from left to right and from top to bottom). All the programs written by you till now, have used the normal sequential flow. However, you can change the normal sequence of the execution of the statements in a program depending upon the requirements of the problem. The statements which are used to alter the order or flow of execution of statements are called '**control flow statements**'. In this lesson, you will learn about the two major categories of control flow statements: '**Selection Statements**' and '**Iterative Statements**'.



### OBJECTIVES

After reading this lesson, you will be able to:

- learn about the usage of selection statements;
- use different forms of selection statement;
- apply looping statements;
- appreciate the need and use of jump statements.

---

### 9.2 SELECTION STATEMENTS

---

These statements allow us to choose the set of instructions for the execution depending upon the given condition. If the condition is true then it executes some set of statements. The selection statements are also called '**conditional statements**' or '**decision statements**'.

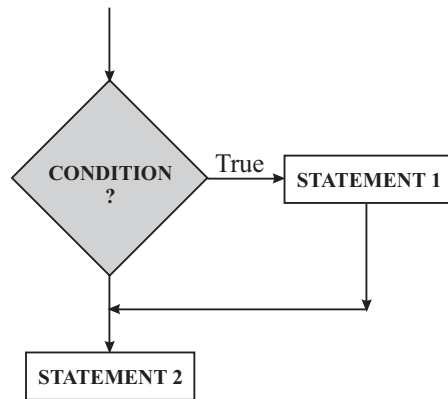


The selection statement in Python is: if statement. The various forms in which if statement can be used in Python, are as follows:

- Simple if statement
- Compound if (if . . . else statement)
- Complex if (if . . . elif . . . else ladder)
- Nested if

### 9.3 SIMPLE IF STATEMENT

This statement will execute the set of statements if the condition is true. In the Fig. 9.1 you can easily identify that if the given condition is true then Statement 1 will be executed. Then it will execute Statement 2.



**Fig. 9.1: Flow chart for if Statement Execution**

The syntax for writing simple if statement is:

#### **Syntax**

if conditional-expression:

Statement-block

In the Example 1, if the conditional expression evaluates to true, then the indented block of code is executed. If the condition is not true, then the indented statements will be skipped. Statements which are out-dented, are not a part of if statement. These are always executed in the normal forward flow of statements, after if statement. The indentation is important while writing if statement.



Notes

**Example 1**

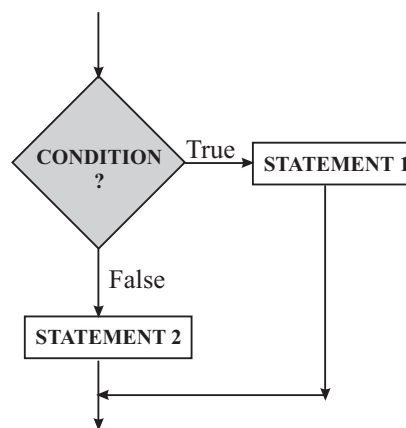
The following Python code will check whether a given number is a positive number or not.

| Program Code  | Output  |
|---|---|
| <pre>num= int (input ('Enter a number')) if num &gt; 0:     print('positive') print('program executed')</pre> | <p><b>Run 1 with input 56</b><br/>Enter a number 56<br/>positive<br/>program executed</p> <p><b>Run 2 with input -12</b><br/>Enter a number-12<br/>program executed</p> |

The above program first gets an input (number) from the user. Then it checks whether the given number is greater than zero or not. If the number is greater than zero, it will display 'positive' and 'program executed' otherwise it will display 'program executed'.

**9.4 COMPOUND IF (IF . . . ELSE STATEMENT)**

In the compound if statement, if conditional expression is true, then the indented statement block (i.e., Statement 1) immediately following the statement if will be executed. If the conditional expression evaluates to false, then the indented statement block (i.e., Statement 2) following the else clause will be executed. At a time, only one block of statements will be executed either Statement1 or Statement2 (refer Fig. 9.2).



**Fig. 9.2: Flowchart for Compound if**



The syntax for writing compound if statement is:

### Syntax

if conditional-expression:

    Statement1

else:

    Statement2

### Example 2

The following program will find whether a given number is an even number or an odd number.

| Program Code   | Output   |
|--|--|
| <pre>num=int (input ('enter a number ')) if num%2 == 0:     print ('EVEN') else:     print ('ODD')</pre> | <p><b>Run 1</b><br/>enter a number 45<br/>ODD</p> <p><b>Run 2</b><br/>enter a number 24<br/>EVEN</p> |

The above program will first get the input a number from the user. Then it checks whether the given number is divisible by 2, i.e., when we divide even number by 2 it will give the remainder as 0. If it returns 0 then it will display EVEN otherwise it will display ODD.

### Example 3

The following program will get two numbers as an input from the user and display the bigger number.

| Program Code  | Output   |
|---|--|
| <pre>num1=int (input ('enter first number ')) num2=int (input ('enter second number ')) if num1&gt; num2:     print ('Bigger number is: ', num1) else:     print ('Bigger number is: ', num2)</pre> | <p>enter first number 45<br/>enter second number 24<br/>Bigger number is: 45</p> |

The above program, will first get input of two numbers num1 and num2. Then it checks whether num1 is greater than num2 or not. If num1 is greater than num2, then num1 variable's value will be displayed otherwise the control goes to else part of the statement and num2 variable's value will be displayed.



Notes

9.5 COMPLEX IF (IF . . . ELIF . . . ELSE LADDER)

In the complex if, there is a ladder of multiple conditions presented by each if, all of these conditions are mutually exclusive. Out of the multiple conditions only one of them will evaluate to true and then the statement following that condition will be executed (refer Fig. 9.3). Once one of the conditional expressions become true all the conditions below it will not be evaluated (checked). If none of the conditional expression becomes true then the statement following the else clause is executed. It is not mandatory to add an else at the end of the ladder. Each of the code blocks (statements) in an if/elif-statement must be indented by the same amount.

Syntax:

```
if conditional-expression1:  
    Statement 1  
elif conditional-expression2:  
    Statement 2  
:  
else:  
    Statement n
```

Let us see it in a flowchart depicted in Fig 9.3.

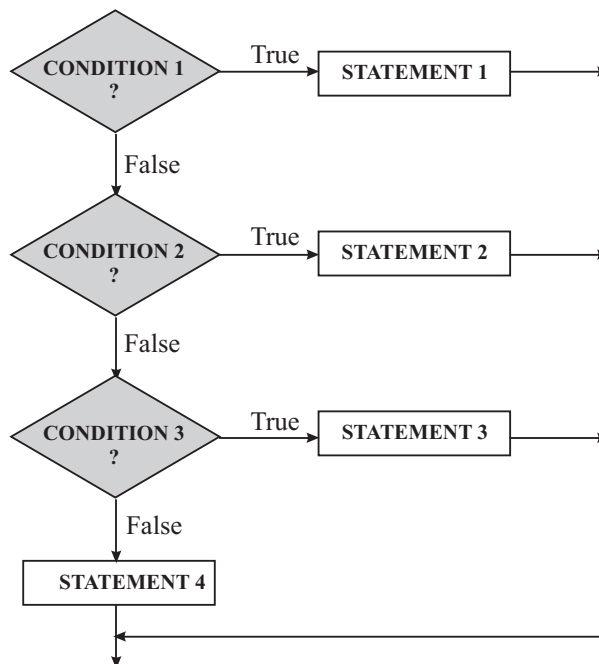


Fig. 9.3: Flowchart of Complex if



**Example 4**

The following program will accept a number of week day (1-7) and print the name of the week day according to the given number (i.e., 1 - Monday....., 7 - Sunday).

| Program Code  | Output  |
|---|---|
| <pre> day=int(input('Enter a number ( 1 - 7):')) if day ==1:     print('Monday') elif day ==2:     print('Tuesday') elif day ==3:     print('Wednesday') elif day ==4:     print('Thursday') elif day ==5:     print('Friday') elif day ==6:     print('Saturday') elif day ==7:     print('Sunday') else:     print('You entered a wrong number') </pre> | <p><b>Run 1</b></p> <p>Enter a number (1 - 7): 5<br/>Friday</p> <p><b>Run 2</b></p> <p>Enter a number (1 - 7): 9<br/>You entered a wrong number</p> |

Notes

The above program will get the input from the user. If the given number is 1 it will display Monday, if it is 2 it will display Tuesday, if it is 3 it will display Wednesday and so on. If the user enters a number other than 1-7, a message 'You entered a wrong number' will be displayed.

**Nested If**

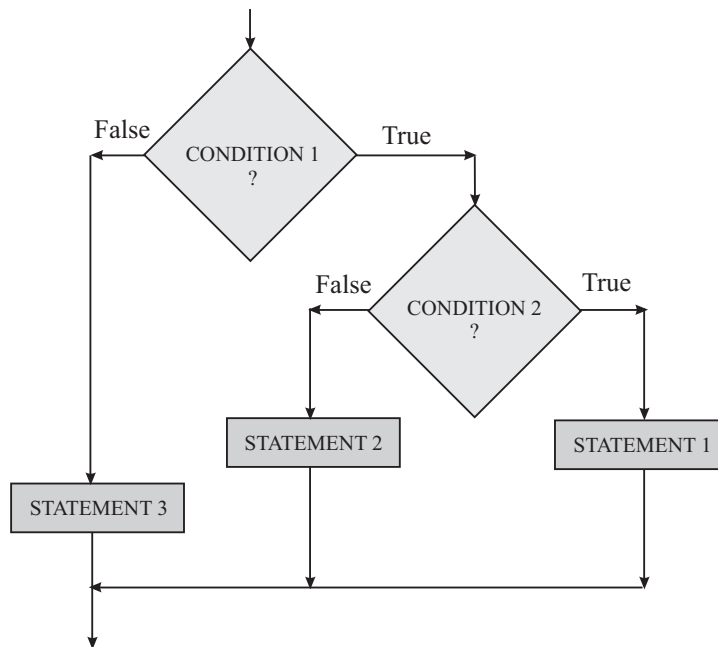
When you use if statement inside another if or else clause statement then it is called nested if statement. The syntax for nested if statement is:



**Notes**

**Syntax**

if conditional-expression 1:  
    if conditional-expression 2:  
        Statement 1  
    else:  
        Statement 2  
else:  
    Statement 3



**Fig. 9.4: Flow chart for Nested If Statement**

In the Fig. 9.4, if the first conditional expression (i.e., conditional expression 1) is true, then the control passes to the conditional expression 2. If the conditional expression 2 evaluates to true, the statement 1 will be executed otherwise statement 2 will be executed. If the first conditional expression (i.e., conditional expression 1) is false, then the conditional expression 2 will not be executed. Then the control reaches directly to second else statement. As a result, statement 3 will be executed.



Notes

**Example 5**

The following program will get three numbers as inputs from the user and display the greatest number from the given numbers.

| Program Code  | Output   |
|---|--|
| <pre>num1=int (input ('enter first number')) num2=int (input ('enter second number')) num3=int (input ('enter third number')) if num1&gt; num2:     if num1&gt;num3:         print ('Greatest number is =', num1)     else:         print ('Greatest number is =', num3) else:     if num2&gt;num3:         print ('Greatest number is =', num2)     else:         print ('Greatest number is =', num3)</pre> | <pre>enter first number45 enter second number24 enter third number78 Greatest number is=78</pre> |

The above program, will input three numbers: num1, num2, num3. If num1 is greater than num2 then it will check whether num1 is greater than num3 or not. If num1 is greater than num3 then num1 will be displayed as the greatest number otherwise num3 will be displayed as the greatest number. If num1 is less than num2 then it will check whether num2 is greater than num3 or not. If num2 is greater than num3 then num2 will be displayed as the greatest number otherwise num3 will be displayed as the greatest number.

**INTEXT QUESTIONS 9.1**

1. What is a selection statement? Write different forms in which if statement can be used in Python.
2. Write the output of the following program segments:
  - (a) a=7  
if a>7:  
    print(a)  
print(a+1)

## MODULE-2

Computer Applications  
(712)



Notes

### Control Statements

- (b) `x=5`  
`if x==5:`  
    `print (x)`  
    `print(x+10)`
- (c) `num=10`  
`sum=0`  
`if num== 1:`  
    `sum=sum+5`  
`elif num== 10 or num == 20:`  
    `sum=sum+10`  
`else:`  
    `sum=sum-5`  
`print (sum)`
- (d) `if 0:`  
    `print("Hello")`  
`else:`  
    `print("welcome")`
- (e) `if 5:`  
    `print("true")`  
`else:`  
    `print("false")`

3. Identify all the errors in the given code and rewrite the code underlining all the corrections made:

- (a) `d=9`  
`if d>2`  
    `print("Yes")`  
`else`  
    `print("No")`
- (b) `k=3`  
`if(k==1):`  
    `print("1")`  
    `print("Continue")`  
`elif(k= =2)`  
    `print("2")`  
`else:`  
    `print("3")`

```
(c) k=5
    if k>5:
        print("Big")
    elseif k>10
        print("Bigger")
    else
        print("Biggest")
```

- Write a Python script to input a number and check if it is positive, negative or zero. Display suitable message.

---

## 9.6 LOOPING STATEMENTS

---

Sometimes you may have to repeat some set of statements repeatedly in a program. Instead of writing the same set of statements again in your program, you can use looping statements. The looping statement allows us to execute a statement or group of statements repeatedly either for a specific number of times or until the condition is true. It stops when the condition becomes false. In the real world, many tasks are cyclical in nature, and therefore these tasks run in a loop.

Humans also perform many tasks repeatedly such as we buy grocery every month, school fee is paid every month, we pay electricity bill, telephone bill every month etc. Similarly you can repeat some set of activities in Python with the help of looping statements. The looping statements in Python are:

- for loop
- while loop

Let us discuss about for loop now.

### For Loop

This loop is used to execute a set of statements for a fixed number of times. Usually if you know the number of times the statements are going to execute then you can prefer to use for loop. Python's for statement iterates over the items of a sequence (list, string, tuple etc.) in the order that they appear in the sequence. Syntax of the for loop is:

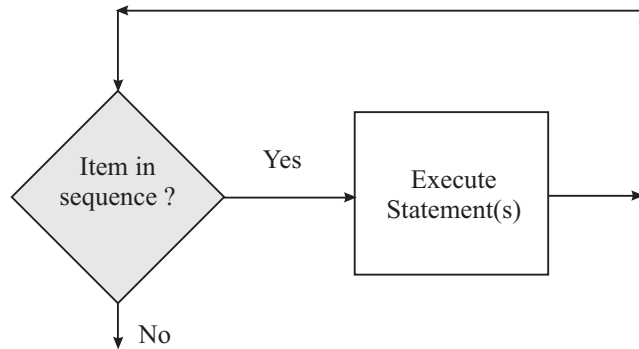
#### Syntax:

```
for loop_variable in sequence:
    statement(s)
```





Notes



**Fig. 9.5: Flow chart for Loop**

In the beginning, the loop variable is assigned to the first value present in the sequence. Next the statement block will be executed. After the execution of the statement block, the loop variable takes the next value present in the sequence. Now the statement block will get executed for the loop variable’s next value. This process continues till the entire sequence is exhausted. The indented statements that follow the loop form the suite/loop body.

**Example 6**

The following program will display “HELLO” word five times using for loop.

| Program Code           | Output |
|------------------------|--------|
| for i in [1,2,3,4,5] : | HELLO  |
| print('HELLO')         | HELLO  |
|                        | HELLO  |
|                        | HELLO  |
|                        | HELLO  |

Initially the first value i.e., 1 will be assigned to the loop variable. Now the body of the loop (here in the example 6, the statement print(‘HELLO’)) will be executed i.e., HELLO will be printed on the screen. After executing the body of the loop, loop variable i will be having next value i.e., 2. Now again the body of the loop will be executed i.e., HELLO will be printed on the screen. This process is repeated for 5 times (i.e., for the values i=1, i=2, i=3, i=4 and i=5 the loop will be executed).

**For Loop With range() Function**

The range() function is used when you need to iterate over a sequence of numbers.

**Syntax**

range([beg], end, [step]) # beg, step are optional

The function range( ) generates a list of values starting from beg (beginning number) till end-1. The default value for step is 1 i.e., if step parameter is not specified 1 is added at every iteration. If the beginning value is not specified the list starts from zero.

**Example7**

The following program will find the factorial of a given number.

| Program Code   | Output                                 |
|--|--|
| <pre>fact=1 num=int(input('enter number')) for i in range(1,num+1):     fact=fact*i print('Factorial=',fact)</pre> | <pre>enter number5 Factorial=120</pre> |

In the above program a variable fact is initialized with value 1. During the execution of the loop the variable i takes values from 1 to 5 (user given number) and the variable fact is multiplied by i at each iteration and the same value is getting stored in fact variable. For the first time when for loop executes for the value i=1, the fact variable will be having the value fact=fact \* 1 i.e., fact=1\*1 => fact=1. In the second time, when the loop executes for the value i=2, the fact variable will be having the value fact=fact\*2 i.e., fact=1\*2=> fact=2. The variable will be having the value 2 now. In the third time, when the loop executes for the value i=3, the fact variable will be having the value fact=fact\*3 i.e., fact=2\*3=> fact=6. In the same way, the loop executes for the values 4 and 5 of the variable i. After the execution of the loop for i=4 and 5 the value of the variable fact will become 5\*4\*3\*2\*1 => 120.

**Note:** The factorial of a number n is calculated as per the product of numbers from 1 to n. i.e., 1\*2\*...\*n.

**while loop**

Another category of looping statement supported by Python is while loop. If the test expression (given condition) is true then the while loop executes the set of statements repeatedly. At any point of time if the test expression becomes false the loop will not be executed and control passes to the statement following the loop.



Notes

**Syntax:**

while (test expression):  
     statement-block

**Example 8**

The following program will get number as an input and display the sum of its digits.

| Program Code  | Output  |
|---|---|
| <pre>sum=0 num=eval(input('Enter a number')) while num&gt;0:     a=num%10     num=num//10     sum =sum + a print("sum of digits =",sum)</pre> | <pre>Enter a number 234 sum of digits = 9</pre> |

In the above program, the while loop is executed till the value of a given number is greater than zero. Inside the loop, the remainder value of a given number after it is divided by 10 will be stored in the variable a. In the next step, the given number is divided (Integer division) by 10 (After division num variable stores value 23 in the above example) and that value will be stored in the given number. The present value of the variable a, is now added to the sum variable. This process continues till the num variable becomes 0. Finally, the value of variable sum will be displayed.

**Else Clause on Loops**

In the same way, if statements had else clause, the looping statements also have an **else clause**.

- If the **else clause** is used with **for loop**, it is executed when the loop terminates after completely iterating the given sequence.
- If the **else clause** is used with **while loop**, it is executed when the condition becomes false.





Notes

### Using Else with for Loop

#### Example 9

The following program will print 10, 20, and 30 and then print “Finished”.

| Program Code         | Output   |
|----------------------|----------|
| for i in [10,20,30]: | 10       |
| print(i)             | 20       |
| else:                | 30       |
| print(“Finished”)    | Finished |

In the above program, the variable *i* take three values 10, 20 and 30 and display them. After this the loop terminates and then the else statement will be executed and ‘Finished’ word will be displayed.

### Using Else with While Loop

Let us consider the following program:

| Program Code      | Output   |
|-------------------|----------|
| i=1               | 1        |
| while i<4:        | 2        |
| print(i)          | 3        |
| i=i+1             | Finished |
| else:             |          |
| print(“Finished”) |          |

In the above program, the variable *i* take the value 1. The while loop will be executed three times for the variable *i*’s values 1, 2 and 3. It displays the values of the variable *i*. After this the loop terminates and then the else statement will be executed and ‘Finished’ word will be displayed.



#### INTEXT QUESTIONS 9.2

1. What is the purpose of looping statements? How many types of looping statements are supported by Python?
2. Give the output of the following code fragments:
  - (a) for ch in ‘abcde’:
 

```
print(ch)
```

## MODULE-2

Computer Applications  
(712)



Notes

### Control Statements

- (b) `k=5`  
`i=10`  
`while i<25:`  
    `print(k+i)`  
    `k=k*5`  
    `i=i+10`  
`print("Bye")`
- (c) `for j in range(2,10,2):`  
    `k=j*3`  
    `if k%6 == 0 :`  
        `print(k)`  
        `print("Loop continues")`
- (d) `sum=1`  
`for I in range (1,10,1):`  
    `sum=sum+ I*5`  
`print(sum)`
- (e) `p=10`  
`while(p<=500):`  
    `p=p+100`  
    `print(p)`  
`print ("ok")`

3. Identify the errors in the following program segments and write the correct program underlining all the corrections made.

- (a) `for n of range(10,50,2)`  
    `print("n=",n)`
- (b) `for k in values(1,10,1):`  
    `print("Learning Python")`
- (c) `i=10`  
`while(i<=100)`  
    `i=i+10`
- (d) `for i in(0,1,2):`  
    `print(j*2)`



Notes

4. Convert the following for loop to while loop:

```
for k in range(5,20,4):  
    print(k)  
print("End")
```

5. Convert the following while loop to for loop

```
k=5  
i=10  
while i<100:  
    print(k+i)  
    k=k*5  
    i=i+10
```

6. How many times will the following loop executed?

(a) `s = 0`  
`i = 0`  
`while ( i < 5):`  
    `s = s* i`  
    `i=i+1`

(b) `s = 0`  
`for i in range(1,10,1):`  
    `s = s+ i`

7. Write a Python script to input a number and check whether it is present in a given list of numbers or not.
8. Write a Python script to input a number and count the occurrences of that number in a given list.

---

## 9.7 JUMP STATEMENTS

---

Jump statements are used to unconditionally transfer the control from one point to another point in the program. These are usually used to either terminate a **loop** instantly or skip the execution of a section of the loop. The jump statements provided by Python are:

- `break`
- `continue`



Notes

**Break Statement**

This statement enables a program to skip over a part of code. A break statement terminates out of the current loop. The program execution resumes at the statement immediately following the loop.

**Example 10**

The following program will print all the numbers of the given list, till a negative number is encountered.

| Program Code                   | Output |
|--------------------------------|--------|
| for i in [0,1,2,-3,-4,5,6,-7]: | 0      |
| if(i<0):                       | 1      |
| break                          | 2      |
| print(i)                       |        |

In the above program, the variable i is given a list of values: 0,1,2,-3,-4,5,6,-7. Each value of i will be checked. If it is less than zero, then the loop will break otherwise the value of i will be printed.

**Continue Statement**

This statement causes the current iteration of the loop to skip and go to the next iteration. All the statements following the continue statement in the loop will not be executed and loop control goes to the next iteration.

**Example 11**

The following program will print all the numbers from 1 to 5 except 2.

| Program Code          | Output |
|-----------------------|--------|
| for i in [1,2,3,4,5]: | 1      |
| if(i ==2):            | 3      |
| continue              | 4      |
| print(i)              | 5      |

In the above program, the loop executes for the values of the variable *i* from 1 to 5. If the value of *i* is equal to 2, then the current iteration will be skipped i.e., the print statement will not be executed. For every other value of *i*, the print statement will be executed and the value of *i* will be printed on the screen.

**INTEXT QUESTIONS 9.3**

I. Give the output of the following code fragments:

1. 

```
for i in range(100,20,-20):  
    if i%3==0:  
        break  
    print ("welcome")
```
2. 

```
for i in range(100,20,-10):  
    if i%7==0:  
        continue  
    print ("welcome")
```

**WHAT YOU HAVE LEARNT**

- The selection statements allow us to choose a set of instructions for execution depending upon the condition given in the program.
- The various forms of if statement in Python are:
  - Simple if statement
  - Compound if (if . . . else STATEMENT)
  - Complex if (if . . . elif . . . else LADDER)
  - Nested if
- An iterative or looping statement allows us to execute a statement or group of statements repeatedly either for a specific number of times or until the condition is true.
- For loop is used to execute a set of statements for a fixed number of times.
- The range() function is used in for loop, when we need to iterate over a sequence of numbers.

**Notes**

## MODULE-2

Computer Applications  
(712)



Notes

### Control Statements

- The while loop is used for executing a set of statements till the test expression becomes false.
- The break statement enables a program to skip over a part of code. A break statement terminates out of the current loop.
- Continue statement causes the current iteration of the loop to skip and go to the next iteration.



### TERMINAL EXERCISE

1. Write a program to input a year in 4-digit form and check whether it is a leap year or not and display proper message.
2. Write a program to input a character. If it is lowercase letter, convert it to uppercase and vice versa.
3. Write a program to reverse a given number.
4. Write a program that will find Pythagorean triplets between 1 to 100. (Pythagorean triplets are such that the square of one number is equal to the sum of the squares of the other two numbers).

Example: 3, 4, 5

$$5^2 = 3^2 + 4^2$$

$$25 = 9 + 16 = 25$$

5. Write a program that will ask the user to enter N integers. Find and display the following:
  - (i) Total number of positive integers
  - (ii) Total number of negative integers
6. Write program to enter a character and check whether it is a vowel or not and display a proper message.
7. Write a program to input 10 numbers and count the numbers which are Armstrong numbers. (The number whose sum of cube of its digits is the number itself is an Armstrong number. For example: 153 is an Armstrong number as  $153 = 1^3 + 5^3 + 3^3$ )



## ANSWERS TO INTEXT QUESTIONS

## 9.1

1. The selection statements allow us to choose a set of instructions for execution depending upon the truth value of the expression. Various forms in which if statement exists are:
  - o simple if statement
  - o compound if (if . . . else statement)
  - o complex if (if . . . elif . . . else ladder)
  - o nested if
2. (a) 8  
(b) 5  
15  
(c) 10  
(d) welcome  
(e) true
3. (a) d=9  
if d>2:  
    print("Yes")  
else:  
    print("No")  
(b) k=3  
if( k==1):  
    print("1")  
    print("Continue")  
elif(k==2):  
    print("2")  
else:  
    print("3")  
(c) k=5  
if k>5:  
    print("Big")  
elif k>10:  
    print("Bigger")  
else:  
    print("Biggest")



Notes

## MODULE-2

Computer Applications  
(712)



Notes

```
4. num=eval(input("enter a number"))
   if num >0:
       print("Its positive")
   elif num <0:
       print("Its negative")
   else:
       print("Its zero")
```

### 9.2

1. A looping statement allows us to execute a statement or group of statements repeatedly either for a specific number of times or until the condition is true. Python supports two types of looping statements:
  - o for loop
  - o while loop
2. (a) a  
b  
c  
d  
e
- (b) 15  
45  
Bye
- (c) 6  
Loop continues  
12  
Loop continues  
18  
Loop continues  
24  
Loop continues
- (d) 226
- (e) 110  
210  
310  
410  
510  
ok



3. (a) 

```
for n in range(10,50,2):  
    print("n=",n)
```
- (b) 

```
for k in range(1,10,1):  
    print("Learning Python")
```
- (c) 

```
i=10  
while(i<=100):  
    i=i+10
```
- (d) 

```
for i in (0,1,2):  
    print(i*2)
```
4. 

```
k=5  
while k<20:  
    print(k)  
    k=k+4  
print("End")
```
5. 

```
k=5  
for i in range(10, 100,10):  
    print(k+i)  
    k=k*5
```
6. (a) 5  
(b) 9
7. 

```
list1=[10, 20,30, 40, 10, 50, 10]  
num=eval(input("enter a number"))  
found=False  
position = 0  
while position <len(list1):  
    if list1[position] == num:  
        found = True  
        break  
position = position + 1  
if found:  
    print(num, "is present at position", position+1)  
else:  
    print(num, "is not present")
```



## MODULE-2

Computer Applications  
(712)



Notes

### Control Statements

```
8. list1=[10, 20,30, 40, 10, 50, 10]
num=eval(input("enter a number"))
count=0
for i in list1:
    if num == i:
        count=count+1
print(num, "is present", count, "number of times")
```

### 9.3

```
1. welcome
welcome
2. welcome
welcome
welcome
welcome
welcome
welcome
welcome
welcome
```

#### Key Learning Outcomes

- Be able to solve the given problems by using looping statements in the program.
- Be proficient in using conditional and jump statements in Python.



# 10

## PYTHON LISTS

---

### 10.1 INTRODUCTION

---

Till now you have learnt about various data types and declaring variables of these data types for storing the information. You have also written down many programs illustrating the usage of numeric data types in the last lesson. The numeric data type stores one single value. Suppose you wish to store marks of all the students of class XII and there are 50 students in class XII. It is not feasible to declare 50 variables for storing marks of 50 students. Declaring so many variables will make the program code lengthier and it is also difficult to remember so many variable names. In this situation, list can help you to store a series of values. In this lesson, you will learn about list, creating lists, adding and updating the elements to/from the list, searching elements in a list, and do some basic operations on the lists and use lists functions.



### OBJECTIVES

After reading this lesson, you will be able to:

- access the list elements;
- add/delete elements in a list;
- search list elements and
- use the list operators and functions.

---

### 10.2 WHAT IS A LIST?

---

Generally, before visiting any market, people prepare a list of items to be purchased. The list of items may be written on a paper one after another. In the same way, you can also prepare a list in Python, which is an ordered collection of



comma-separated values (items) enclosed within square brackets. The values that make up a list are called its **‘elements’**. The main features of the list are:

- The values in the list can be modified, i.e., it is mutable.
- The elements in a list need not be of the same type.

**Declaring/Creating List**

There are two ways of creating a list. These are:

- putting comma separated values/items within square brackets.
- creating a list from an already existing list.

Let us see an example of creating a list by putting comma separated values/items within square brackets.

**Examples**

```
list1=[10, 20, 30, 40, 50]
list2=["Geeta", 100, 200, 300, "Sandeep"]
list3=['A', 'E', 'I', 'O', 'U']
list4=[ ] #empty list
```

Here the list1 contains the elements - 10,20,30,40,50. The list2 has five elements such as, “Geeta”, 100, 200, 300, “Sandeep”. The list3 contains five elements – ‘A’, ‘E’, ‘I’, ‘O’, ‘U’ and list4 is an empty list. The string values should be enclosed either in single or double quotes.

You can easily create a new list from an already existing list. It is possible to copy either an entire list or only selected number of elements of the existing list to the new list. Let us see few examples given in Table 10.1 for creating a list using an existing list.

**Table 10.1: Creating a List**

| Statement        | Description   |
|------------------|---|
| list5=list1[:]   | This statement will copy the list1 to list5 i.e., all the values of list1 will get copied to list5.list5=[10, 20, 30, 40, 50] |
| list6=list1[0:3] | The list6 will have the first, second and third elements of list1 i.e., list6=[10, 20, 30]                                    |
| list7=list1      | Here the list7 will have all the elements of list1 i.e., list7=[10, 20, 30, 40, 50]   |



Notes

### 10.3 ACCESSING LIST ELEMENTS

You have assigned the elements to a list. Now if you want to access the elements in the list, then following two methods can be used:

- Indexing Method (to access a single value)
- Slicing Method (to access a subset of values)

The elements in a list are stored and accessed according to the index value. The index number starts from zero. Let us discuss about Indexing method now.

#### Indexing Method

This method is used to access one single element at a time from a list. Some important points for using this method are:

- The list indices start at 0.
- The list index has to be a positive or negative integer value or an expression which evaluates to an integer value. The positive value of an index will access the elements from the beginning of the list and negative value will access the elements from the end of the list.
- An Index Error appears, if you try to access the element that does not exist in the list.

Consider the total number of elements in the list is max. The index value and its corresponding element in the list are mentioned in the Table 10.2 (the total number of elements in the list is max).

**Table 10.2: Index Value And The Corresponding Element In the List**

| Index value  | Element In The List          |
|--------------|------------------------------|
| 0, or -max   | 1 <sup>st</sup> element      |
| 1, or -max+1 | 2 <sup>nd</sup> element      |
| 2, or -max+2 | 3 <sup>rd</sup> element      |
| max-2, or -2 | 2 <sup>nd</sup> last element |
| max-1, or -1 | Last element                 |



Notes

Let us consider the following lists:

**list1**

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 0   | 1   | 2   | 3   | 4   |
| 100 | 200 | 300 | 400 | 500 |
| -5  | -4  | -3  | -2  | -1  |

**list2**

|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  |
| A  | E  | I  | O  | U  |
| -5 | -4 | -3 | -2 | -1 |

Here you can use the Indexing method to access the elements of the above lists. Let us see the below example.

**Example 1**

| Python Statement | Output |
|------------------|--------|
| list1[2]         | 300    |
| list2[-1]        | U      |

In the above example, list1[2] will display the third element of the list1 because the list index starts from 0. If you try to access list1[2] then it is actually the third element present in the list1. The list2[-1] will display the last element of list2 (i.e., U)

**List Slicing Method**

This method is used to retrieve a subset of values from the list. A slice of a list is basically its sub-list. The syntax used for slicing method is:

**Syntax**

**list [start: stop: step]**

where start is the starting position; stop is the stopping position; step is the step size also known as stride.

If you omit the first index, slice starts from 0 and omitting of stop value will take it to the end of the list. Default value of step is 1 which means it access the next element from the list. Suppose if you mention step as 2 then it will access the next-



Notes

to-next element from the list. This method includes the starting position element but excludes the stopping position's element. So it lists all the elements from the starting position to stopping position -1.

**Example 2**

Consider a list z1 with the values,

z1=[121, 451.78, "KABIR", "RAMESH", 890, 453.90]

|     |        |       |        |     |        |
|-----|--------|-------|--------|-----|--------|
| 0   | 1      | 2     | 3      | 4   | 5      |
| 121 | 451.78 | KABIR | RAMESH | 890 | 453.90 |
| -6  | -5     | -4    | -3     | -2  | -1     |

Then, z1[:] statement will return all the elements of z1, i.e., 121, 451.78, "KABIR", "RAMESH", 890, 453.90. In the same way, z1[2:5] will return the three elements i.e., second, third and fourth index value elements.

Let us see the following examples of list slicing methods:

| Python Statements | Output  |
|-------------------|---|
| z1[:5]            | [121, 451.78, 'KABIR', 'RAMESH', 890]<br>It displays the elements from the zero index value to fourth index value. By default it takes the starting position as 0.            |
| z1[3:]            | ['RAMESH', 890, 453.9]<br>It displays the elements of list z1 from third index value to fifth index value. By default it takes the last index value as the stopping position. |
| z1[::2]           | [121, 'KABIR', 890]<br>It displays the elements from beginning till end, but it omits second number and fourth number because here the step variable has value 2.             |



**INTEXT QUESTIONS 10.1**

- If list1 = [22, 33, 44, 55, 66], what is list1[-1] ?  
 (a) Error            (b) 55            (c) 66            (d) 22



Notes

2. If list1 = [22, 33, 44, 55, 66], what is list1[: -1] ?
  - (a) [22, 33, 44, 55]
  - (b) Error
  - (c) 22
  - (d) [22, 33, 44, 55, 66]
3. What is the output of the following code when it is executed?
 

```
>>>list1 = [10, 20, 30, 40,50,60,70]
>>>list2 = list1[2:6]
>>>print(list2)
```

  - (a) [10, 20]
  - (b) [20,30,40,50,60]
  - (c) [20,30,40,50]
  - (d) [30,40,50,60]
4. If list1= [5,25,63,87,98], what is list1[3]?
  - (a) 25
  - (b) 63
  - (c) 98
  - (d) 87
5. If list1 is [34,27,54,68,73,91,28], which of the following is correct answer for the slicing operation: print(list1[:2])?
  - (a) [34,27]
  - (b) [34,27,54]
  - (c) [27,54]
  - (d) [27,54,68]

#### 10.4 ADDING ELEMENTS TO A LIST

You can also add the elements to the list. Three methods are used for adding elements to the list. These are **insert**, **append** and **extend**.

**Insert Method:** This method is used to insert an element at the given index number. The syntax for using this method is:

##### Syntax

**list\_name.insert(index\_number, value)**

where *list\_name* is the name of the list;

*index\_number* is the index value in that particular position the new value to be inserted;

*value* is the new value which has to be inserted in the list.

##### Example 3

Consider a list z1 with values:

z1=[121, 451.78, "KABIR", "RAMESH", 890, 453.90]





Notes

| Python Statement                   | Output   |
|------------------------------------|--|
| <code>z1.insert(4, "KIRTI")</code> | <code>[121, 451.78, 'KABIR', 'RAMESH', 'KIRTI', 890, 453.9]</code> |

The above statement, will insert the element "KIRTI" in the list `z1` at index number 4, and all the other elements will be shifted accordingly to the next position.

|     |        |       |        |       |     |        |
|-----|--------|-------|--------|-------|-----|--------|
| 0   | 1      | 2     | 3      | 4     | 5   | 6      |
| 121 | 451.78 | KABIR | RAMESH | KIRTI | 890 | 453.90 |
| -7  | -6     | -5    | -4     | -3    | -2  | -1     |

**Append Method** - This method is used to add an element at the end of the list. The syntax for using this method is:

### Syntax

`list_name.append (value)`

where *list\_name* is the name of the list;

*value* is the new value to be inserted in the list.

### Example 4

| Python Statement           | Output   |
|----------------------------|--|
| <code>z1.append(10)</code> | <code>[121, 451.78, 'KABIR', 'RAMESH', 'KIRTI', 890, 453.9, 10]</code> |

|     |        |       |        |       |     |        |    |
|-----|--------|-------|--------|-------|-----|--------|----|
| 0   | 1      | 2     | 3      | 4     | 5   | 6      | 7  |
| 121 | 451.78 | KABIR | RAMESH | KIRTI | 890 | 453.90 | 10 |
| -8  | -7     | -6    | -5     | -4    | -3  | -2     | -1 |

You can see that the new value is added to the end position of the list.

**Extend Method:** This method is used to add one or more element at the end of the list. The syntax for using method is:

### Syntax

`list_name.extend (value(s))`

where *list\_name* is the name of the list;

*value(s)* is/are the new value(s) to be inserted in the list.



Notes

**Example 5**

| Python Statement   | Output   |
|--|--|
| <pre>list1=[100, 200, 300, 400, 500] list2=["Raman", 100, "Ashwin"] list1.extend(list2) print(list1)</pre> | <pre>[100, 200, 300, 400, 500, 'Raman', 100, 'Ashwin']</pre> |

Here list2 is added at the end of the list list1.

**10.5 UPDATING LIST**

You have learnt to access list elements and add elements to the list. Now let us learn to modify the elements in the list. In Python, it is possible to modify a single element in the list or a part of a list. For updating a single element of a list, the index number of the element will be used.

**Example 6**

| Python Code  | Output                               |
|--|--------------------------------------|
| <pre>list1=[100, 200, 300, 400, 500] list1[2]=700 print(list1)</pre> | <pre>[100, 200, 700, 400, 500]</pre> |

Here the value of second element of the list list1 is modified with value 700. For updating multiple elements in the list slice method is used.

Let us see an example for updating multiple elements in the list.

**Example 7**

| Python Code  | Output  |
|--|---|
| <pre>list1=[100, 200, 50, 400, 500, 'Raman', 100, 'Ashwin'] list1[2:5]=[90] print(list1)</pre> | <pre>[100, 200, 90, 'Raman', 100, 'Ashwin']</pre> |

In this example, the list1[2], list1[3] and list1[4] elements will be replaced by the value 90 thus instead of the values 100,200, 50, 400, 500, 'Raman', 100, 'Ashwin' the list1 will now have the values 100,200, 90, 'Raman', 100, 'Ashwin'.



## 10.6 DELETING ELEMENTS FROM THE LIST

You can use the methods ‘**Del**, **Remove**’ or ‘**Pop**’ to delete the elements from a List.

**POP Method** - It removes the element from the specified index, and also returns the element which has been removed. The syntax for using POP method is:

### Syntax

**list\_name.pop(index\_value)**

If index\_value is not given in pop() method, the last element will be deleted.

### Example 8

| Python Code   | Output                                   |
|---|--|
| list1=[100, 200, 90, ‘Raman’,<br>100, ‘Ashwin’]<br>list1.pop(2)<br>print(list1) | 90<br>[100, 200, ‘Raman’, 100, ‘Ashwin’] |

In the above example, the pop() method will remove the element at index number 2 (i.e., number 90) and display it. Now if you use print statement for displaying the present value of list1 all the five elements of list1 elements will be displayed (remember number 90 was deleted from the list).

**Del Method:** This method removes the element from the specified index, but it does not return the deleted value. The syntax for using del method is:

### Syntax

**del list\_name[index\_value]**

Let us see an example using del method.

### Example 9

| Python Code  | Output                   |
|--|--------------------------|
| list1=[100, 200, 90, ‘Raman’, 100]<br>del list1[2]<br>print(list1) | [100, 200, ‘Raman’, 100] |



Here the element at index number 2 i.e., 90 has been deleted. You can also give negative index value to del method. Consider list2=[100, 200, 300]. Now if you write del list2[-1], it will delete the last element from the list2 i.e., now list2 will have 100 and 200 only.

**Using DEL With Slicing**

You can delete multiple elements from a list by using **del** with **slicing** method.

**Example 10**

| Python Code   | Output                    |
|---|---------------------------|
| <pre>list1=[100, 200, 90, 'Raman', 100, 'Ashwin'] del list1[2:4] print(list1)</pre> | [100, 200, 100, 'Ashwin'] |

Here the second and third index value elements i.e., 90 and 'Raman' will be deleted from the list list1.

**Remove Method**

The remove method is used when you know the value of the element to be deleted but you are unaware of the index value of that element. It is used to delete an element by value not by its index. The syntax for using remove method is:

**Syntax**

**list\_name.remove(element\_value)**

Let us see an example of remove method.

**Example 11**

| Python Code  | Output                                      |
|--|---|
| <pre>list1=[100, 200, 50, 400, 500, 'Raman', 100, 'Ashwin'] list1.remove(400) print(list1)</pre> | [100, 200, 50, 500, 'Raman', 100, 'Ashwin'] |

Here the element with value 400 is removed from the list list1.



## INTEXT QUESTIONS 10.2

- Which command is used to add the value 80 at the third position in the list1?  
(a) list1.insert(3, 80)                      (b) list1.insert(2, 80)  
(c) list1.insert(80, 3)                      (d) list1.append(3, 80)
- If list1= [68,75,23,49,100,62,89,34] then what will be output after the following statements got executed?  
list1[3:5]=[1000]  
print(list1)  
(a) [68,75,1000,100,62,89,34]      (b) [68,75,1000,62,89,34]  
(c) [68,75,23,1000,89,34]              (d) [68,75,23,1000,62,89,34]
- If list1= [36, 44, 53, 20, 85, 25, 91, 32], what will be the list1 after list1.pop(2)?  
(a) [36, 53, 20, 85, 25, 91, 32]      (b) [36, 44, 53, 20, 85, 91, 32]  
(c) [36, 44, 20, 85, 25, 91, 32]      (d) [36, 44, 53, 20, 85, 25, 32]
- What will be the output of the following code?  
list1 = [1, 2, 4, 'Sandeep', 'Raman', 'Aastha', 8]  
list1.remove(4)  
print(list1)  
(a) [1, 2, 4, 'Sandeep', 'Aastha', 8]  
(b) [1, 2, 'Sandeep', 'Raman', 'Aastha', 8]  
(c) [1, 2, 4, 'Raman', 'Aastha', 8]  
(d) [1, 2, 4, 'Sandeep', 'Raman', 'Aastha']
- Which method is used to add one or more values at the end of the list?  
(a) add    (b) append  
(c) extend                                        (d) insert



Notes



Notes

**10.7 SEARCHING LIST ELEMENTS**

The values present in the lists can easily be searched using the index method. The value to be searched is passed to it as a parameter. The output of this method is the index value of the number present in the list.

**Syntax**

**list\_name.index(element)**

**Example 12**

| Python Code  | Output |
|--|--------|
| list1=[100, 200, 50, 400, 500, 'Raman', 100, 'Ashwin']<br>list1.index(400) | 3      |

Here you are searching the list list1 for finding the index value of the number 400. The answer is 3 as the element 400 is present at the index value 3. If the value, you are searching in a list is not found, then an error message will be displayed.

**10.8 OPERATIONS ON LIST**

Python provides various operations that can be performed on the lists. These are:

- Indexing (to access a single value)
- Slicing (to access multiple values)
- Concatenation (to add two or more lists)
- Repetition (to replicate the list)
- Membership (to find out whether an element is present in a list or not)

You have already learnt about indexing and slicing operations in the above section. In this section, you will learn about concatenation, repetition and membership operations.

**Concatenation**

It is the process of adding two or more lists using the concatenation operator '+'.

**Example 13**

| Python Code  | Output   |
|--|--|
| list1=[100, 200, 50, 400, 500, 'Raman', 100, 'Ashwin']<br>list1 = list1 + [80]<br>print(list1) | [100, 200, 50, 400, 500, 'Raman', 100, 'Ashwin', 80] |



Notes

Please note that '+' operator is used to add two or more lists. In the above example, [80] is added to the list1. You cannot use '+' operator for adding elements to the list. Let us see the following example.

**Example 14**

| PythonCode  | Output  |
|---|---|
| <pre>list1=[100, 200, 50, 400, 500, 'Raman', 100, 'Ashwin'] list1 = list1 + 40 print(list1)</pre> | <pre>Traceback (most recent call last): File "&lt;pyshell#4&gt;", line 1, in &lt;module&gt; list1=list1+40 TypeError: can only concatenate list (not "int") to list</pre> |

Here, the statement `list1=list1+40` will throw an error such as **"TypeError: can only concatenate list(not "int") to list"**.

The following example shows concatenating of two lists.

**Example 15**

| Python Code  | Output  |
|--|---|
| <pre>list1=[100, 200, 300, 400, 500] list2=[700,800,900] list1 = list1 + list2 print(list1) print(list2)</pre> | <pre>[100, 200, 300, 400, 500, 700, 800, 900] [700,800,900]</pre> |

The statement `list1=list1+list2` implies that list2 is concatenated at the end of list1. So the list1 will have [100, 200, 300, 400, 500, 700, 800, 900]. But if you write `list1=list2+list1` then the values of list1 will be concatenated at the end of list2 and the concatenated value will be stored in the list1. Hence, list1 will have [700,800,900,100,200,300,400,500] and list2 will remain the same i.e., list2=[700,800,900].

**Repetition**

It is the process of replicating a list using operator '\*'. The \* operator is used to add the complete list repeatedly (given number of times) towards the end.

Let us see an example using the operator \*.



Notes

**Example 16**

| Python Code                    | Output   |
|--------------------------------|--|
| list2=["Raman", 100, "Ashwin"] | ['Raman', 100, 'Ashwin', 'Raman', 100, 'Ashwin'] |
| list2=list2*2                  |  |
| print(list2)                   |  |

Here the list2 is replicated twice (because of the statement list2\*2).

**Membership**

It is the process of finding out whether an element is present in a list or not using **'in operator'**. It returns **'True'** if the element is present in the list otherwise it returns **'False'**.

**Example 17**

| Python Code   | Output |
|---|--------|
| list1= [100, 200, 50, 400, 500, 'Raman', 100, 'Ashwin'] | True   |
| 500 in list1  |        |

After the execution of the above code, **'True'** value will be returned as '500' is present in the list list1.

**10.9 LIST FUNCTIONS**

You can use some of functions in the list such as, len( ), max( ), min( ) and count( ). Let us see the use of these functions with an example in the Table 10.2.

**Table 10.2: List Functions**

| Function | Description  | Example  |
|----------|--|--|
| len()    | Returns the length of the list.<br>Syntax:<br>len(list-name) | >>> list1=[100,200,50,400,500, 'Raman', 100, 'Ashwin']<br>>>>len(list1)<br>8 |
| max()    | Returns the element with maximum value from the list.        | >>> list3=[10,20,30,40]<br>>>> max(list3)<br>40                              |





## Notes

|       |  |   |
|-------|--|---|
|       | <p>To use the max() function all the values in the list must be of same type.</p>  | <pre>&gt;&gt;&gt; list4=['A', 'a', 'B', 'C'] &gt;&gt;&gt; max(list4) 'a'</pre> <p>Here it will return the alphabet with maximum ASCII value.</p> <pre>&gt;&gt;&gt; list5=['ashwin', 'bharat', 'shelly'] &gt;&gt;&gt; max(list5) shelly'</pre> <p>Here it will return the string, which starts with character having highest ASCII value.</p> <pre>&gt;&gt;&gt; list5=['ashwin', 'bharat', 'shelly', 'surpreet'] &gt;&gt;&gt; max(list5) 'surpreet'</pre> <p>If there are two or more strings which starts with the same character, then the second character is compared and so on.</p> |
| min() | <p>Returns the element with minimum value from the list.</p> <p>To use the min() function all the values in the list must be of same type.</p> | <pre>&gt;&gt;&gt; list3=[10,20,30,40] &gt;&gt;&gt; min(list3) 10</pre> <pre>&gt;&gt;&gt; list4=['A', 'a', 'B', 'C'] &gt;&gt;&gt; min(list4) 'A'</pre> <p>Here it will return the alphabet with minimum ASCII value.</p> <pre>&gt;&gt;&gt; list5=['ashwin', 'bharat', 'shelly'] &gt;&gt;&gt; min(list5) 'ashwin'</pre> <p>Here it will return the string which starts with character having smallest ASCII value.</p>  |

## MODULE-2

Computer Applications  
(712)



Notes

### Python Lists

|         |   |   |
|---------|---|---|
|         |   | <pre>&gt;&gt;&gt; list5=['ashwin','bharat','shelly', 'ayush'] &gt;&gt;&gt; min(list5) 'ashwin'</pre> <p>If there are two or more strings which starts with the same character, then the second character is compared and so on.</p> |
| count() | It is used to count the occurrences of an item in the list. | <pre>&gt;&gt;&gt; list6=[10,20,10,30,10,40,50] &gt;&gt;&gt; list6.count(10) 3</pre>   |



### INTEXT QUESTIONS 10.3

- If list1 = [1, 3, 2], then what will be list1 \* 2 ?
  - [2, 6, 4]
  - [1, 3, 2, 1, 3]
  - [1, 3, 2, 1, 3, 2]
  - [1, 3, 2, 3, 2, 1]
- What will be the output?

```
numbers = [1, 2, 3, 4]
numbers.index(3)
```

  - 4
  - 3
  - 2
  - 1
- What will be the output of the following code?

```
list1 = [1, 2, 3, 4]
list2 = [5, 6, 7, 8]
print(list1 + list2)
```

  - [1, 2, 3, 4]
  - [5,6,7,8]
  - [5,6,7,8,1, 2, 3, 4]
  - [1, 2, 3, 4,5,6,7,8]
- If list1=['h','e','l','l','o'], what will be list1.count('l')?
  - 5
  - 4
  - 2
  - 1

5. If list1 is [4521,1245,5421,2451], what will be the value of max(list1)?
- (a) 4521 (b) 1245  
(c) 5421 (d) 2451
6. If list1 is [3, 5, 25, 1, 3], what will be the value of min(list1)?
- (a) 3 (b) 5  
(c) 25 (d) 1

**WHAT YOU HAVE LEARNT**

- The list is an ordered collection of comma-separated values (items) enclosed within square brackets.
- The elements in a list are stored according to the index value with index starting from value 0.
- The elements of a list can be accessed using an indexing and slicing method.
- The list index has to be a positive or negative integer value or an expression which evaluates to an integer value.
- The list slicing method is used to retrieve a subset of values.
- There are three methods for adding elements in a list: **insert**, **append** and **extend**.
- We can use **del**, **remove()** and **pop()** to delete elements from a list.
- The concatenation is the process of adding two or more lists using the concatenation operator '+'.  
• The repetition is the process of replicating the list using the operator '\*'.  
• The membership is the process of finding out whether an element is present in a list or not using 'in' operator.

**TERMINAL EXERCISE**

1. What is a list in Python? Write its important features.
2. Explain Indexing and Slicing methods of a list.



## MODULE-2

Computer Applications  
(712)



Notes

### Python Lists

3. Explain Del, Pop and Remove methods of list with the help of an example.
4. Give the difference between concatenation and repetition operation on a list.
5. Write a Python script to display the element with maximum value from the given list.
6. Write a Python script to input a list of numbers and add 2 to all the elements which are even and subtract 2 from all the elements which are odd numbers.
7. Write a Python script to input a list of names and display only those names which begin with 's' or 'S'.



### ANSWERS TO INTEXT QUESTIONS

#### 10.1

1. (c)      2. (a)      3. (d)      4. (d)      5. (a)

#### 10.2

1. (a)      2. (d)      3. (c)      4. (b)      5. (c)

#### 10.3

1. (c)      2. (c)      3. (d)      4. (c)      5. (c)      6. (d)

#### Key Learning Outcomes

- Be able to identify the methods to access the elements in the list and update the list effectively.
- Perform the operations such as indexing, slicing, concatenation, repetition and membership operations on the list efficiently.

**11****FUNCTIONS IN PYTHON**

---

**11.1 INTRODUCTION**

---

You might have seen many machines in your life such as, television, radio, washing machine, calculator etc. All these machines have several functionalities, which can be easily performed/controlled by using the buttons fixed on them. For example, a washing machine can perform washing, rinsing, and drying clothes etc. Even though the buttons fixed on the machine have predefined task associated with them, but the machine only performs the function which you select (For example: if you press rinse button, the machine will do rinsing task). In other words, you can say that having different buttons with specific functions associated with them help us to operate the machine easily. Similarly, instead of writing a very lengthy program to perform a certain task, you can write your program as a collection of several small sub programs (modules) or functions. Each of these small sub programs or functions can perform some specific tasks and can be used in other programs also. The functions make the program code more readable, understandable and reusable. The functions are the most important aspect of an application. In this lesson, you will learn about Python functions in detail.

**OBJECTIVES**

After reading this lesson, you will be able to:

- use built-in functions of Python;
- define and call user defined functions for some specific use;
- pass/return arguments to/from the functions and
- use local and global variables in the program.



Notes

### 11.2 FUNCTIONS

A function is a named group of related programming statements that takes inputs, does some specific processing and produces the result for a particular task. A function may or may not return any value. Some of the major advantages of using functions in a program are:

- Modularity:** The functions provide modularity to a program. It is difficult to manage large programs. Thus, a large program is usually broken down into small units called functions (refer Fig 11.1). This makes the program easy to understand, more organized and manageable.

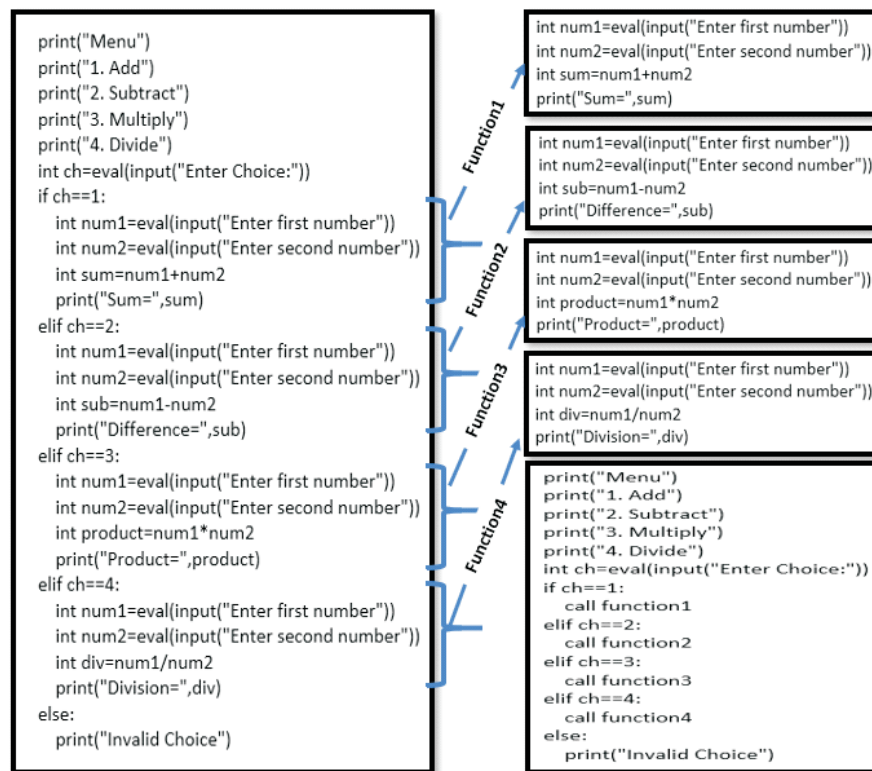


Fig 11.1: Illustrating the Use of Functions in Python

- Code Reusability:** Sometimes, a part of the program code needs to be executed repeatedly. If this part of program code is stored in a function, then you do not need to write the same code again and again. Instead, that particular function can be called repeatedly. This saves time and lines of code in the program. Thus, you can say that functions help to avoid repetition and make the code reusable. Once the function has been defined, it can be used any number of times at any point in your program. You can store the frequently



used functions in the form of modules, which can be easily imported in another program whenever needed.

- **Easy Debugging:** Functions help to debug a large program easily. It is easy to debug small units of a program than a long program.

### 11.3 TYPES OF FUNCTIONS

Basically, you can divide the functions into two categories:

- (i) Built-in Functions
- (ii) User-defined functions

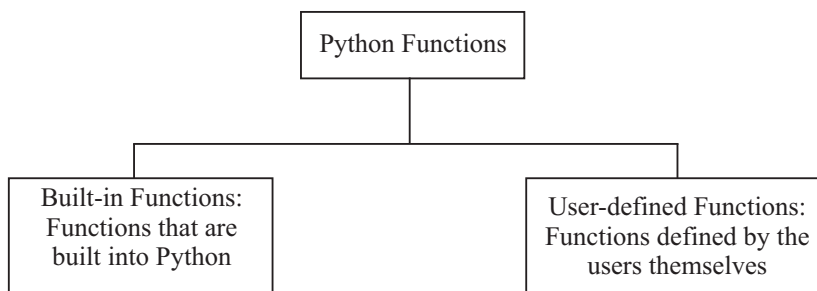


Fig. 11.2: Types of Python Functions

Additionally, Python supports the concept of modules, which are nothing but Python program files containing Python functions, class or variables.

Let us discuss them one by one.

#### 11.3.1 Built in Function

Built in functions are the function(s) that are built into Python and are always available for usage. You do not need to import any module (file) to use these functions. Some of the built-in functions have already been used in your programs such as: input(), print(), eval(), int(), float(), type() etc. Some of the commonly used functions are given in the following Table 11.1.

Table 11.1: List of Built-In Functions

| Function name | Purpose                                    | Example                                 | Explanation   |
|---------------|--|---|---|
| abs()         | It returns the absolute value of a number. | >>> abs(-10)<br>10<br>>>> abs(50)<br>50 | The absolute value of -10 is returned as 10 and absolute value of 50 is returned as 50. |

## MODULE-2

Computer Applications  
(712)



Notes

### Functions in Python

|          |  |   |  |
|----------|--|---|--|
| max()    | This function returns the largest value from the given values.   | >>> max(10, 30,50,20)<br>50   | Here 50 will be displayed because it is the maximum value amongst 10, 30, 50 and 20.   |
| min()    | This function returns the smallest value from the given values.  | >>> min(10, 30,50,20)<br>10   | Here 10 will be displayed because it is the minimum value amongst 10, 30, 50 and 20.   |
| divmod() | This function returns a tuple of quotient and remainder.         | >>>divmod<br>(5,2)<br>(2,1)   | Here 2 is the quotient and 1 is the remainder of the division of 5 by 2. The result is shown in the form of a tuple.                                 |
| round()  | This function rounds a floating point number to n digits places. | >>> round<br>(10.23376,3)<br>10.234<br>>>> round<br>(10.23376,1)<br>10.2<br>>>> round<br>(10.23376)<br>10 | It returns a floating point number rounded to n digits from the decimal point. If n is not provided then, the number is rounded to 0 decimal digits. |

### 11.3.2 User-Defined Functions

So far, you have come across the built-in functions, but it is also possible for you to write your own function(s) according to your own need. These functions can be further combined to form a module which can be used in other programs by importing them. Functions that the user defines for doing certain specific tasks, are referred to as '**user-defined functions**'.

The syntax of user-defined function is:

#### Syntax

```
def function_name (parameter1, parameter2, ...) :  
    statement(s)
```

Here **def:** is a keyword, **function\_name:** is an identifier which is followed by the list of parameters enclosed in parenthesis.

Please note that the list of parameters is optional.





**statement(s):** is the block of statement(s) that form the body of the function and are executed when the function is called.

The function header is the first statement with the keyword `def` and the name of function, which is followed by a colon(`:`). The indented statements following the function header form the **'body of the function'** as shown in Fig 11.3.

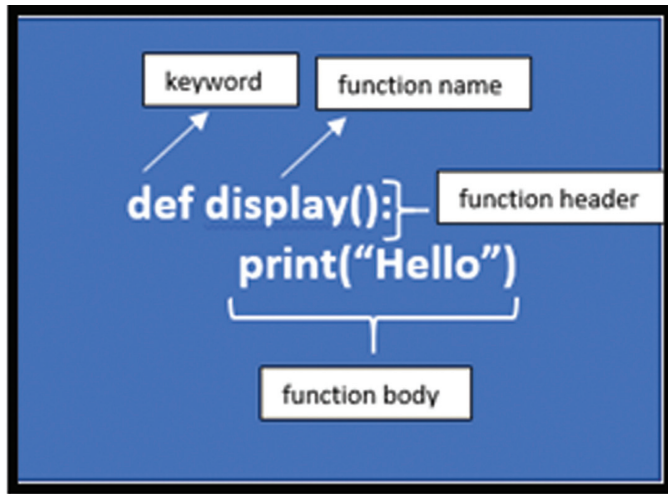


Fig. 11.3: Writing a Function in Python

**Example 1**

```
def display1():
    print("Hello")
```

In this function, the function body has only one statement. If there are more than one statements in the function body then all the statements of the function should be at the same indent level.

**Calling a Function**

The body of the function gets executed only when the function is called. A function is called by giving a function name followed by parenthesis.

The syntax for calling a function is as follows:

**Syntax**

```
function_name()
```

Let us see an example.



Notes

**Example 2**

To call the function **display1()**, you can simply write the statement:

```
>>>display1()
```

In the above case, the function **display1()** will show **Hello** as the output.

When a function call statement is executed, the program control jumps to the respective function definition and start executing the function body as shown in Fig 11.4. After the execution of the function, the program control jumps back to the statement following the function call.

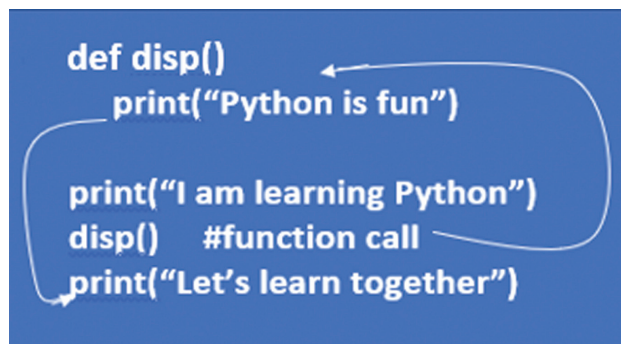


Fig 11.4 Execution of the Function

**Example 3**

```
# function definition starts
def disp():
    print("Python is fun")
#function definition ends
print("I am learning Python")
disp() #function call
print("Let's learn together")
```

**Output**

```
I am learning Python
Python is fun
Let's learn together
```

In the above example, **disp()** is a user-defined function. The program execution starts from the first print statement and '**I am learning Python**' will be displayed.



After this, the function call is encountered, and the program control jumps to the disp() function and executes the statements present in the body of the function (refer Fig. 11.4). The print statement ‘**Python is fun**’ will now be executed and the same text will be displayed. Then the program control returns to the statement following the function call i.e., the next print statement in this program will be executed. It will display ‘**Let’s learn together**’ message.

### Passing Arguments to Functions

You can easily pass the information or values to the functions. You can specify these values inside the parenthesis after the name of the function. Depending upon the requirement, you can pass any number of values to the function. The values provided in the function call are called ‘**arguments**’. These arguments have one to one mapping with the ‘**parameters**’ present in the function header. A parameter is nothing but a variable declared inside the parenthesis of a function definition.

#### Example 4

```
#Function definition
def display(str):
    print(str)
#Function call
display (“I am happy”)
```

In the above example **str** is a parameter and it is present in the function header as shown in Fig 11.5. In the statement display(“**I am happy**”), the string “**I am happy**” is an argument.

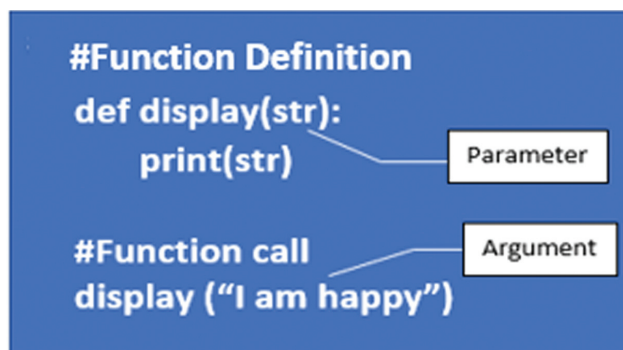


Fig. 11.5: Parameter Passing In Functions

You can pass the arguments in two different ways: These are ‘**Positional arguments**’ and ‘**Keyword/Named arguments**’. Let us discuss about these arguments in detail.



## Notes

**Positional Arguments:** When the arguments are passed as positional arguments, the order in which the arguments are sent or their position in the function call is very important. It is necessary to pass the arguments in the same order as the parameters. The arguments should have one to one correspondence/mapping with the parameters present in the function header. Also, the number and type of arguments passed should match the number and type of parameters present in the function header.

**Example 5**

```
#Function definition
def val_expression(a,b, c):
    result=(a+b+c) * (a-b-c) * (a*b*c)
    print result

#Function call
>>>val_expression(12,6,2)

Output
11520

#Function call
>>>val_expression(5,3,2)

Output
0
```

In the above example, `val_expression` is a function with three parameters `a`, `b`, and `c`. It calculates the value of the expression  $(a+b+c)*(a-b-c)*(a*b*c)$ . When the function `val_expression(12,6,2)` is called first time the arguments 12,6,2 are mapped with the parameters `a`, `b`, and `c` respectively i.e. the parameter '`a`' will get the value '12', '`b`' will get the value '6' and '`c`' will get the value '2'. These types of arguments are called '**positional arguments**'. When the function is called second time, `eval_expression(5,3,2)`, the parameter '`a`' will get the value '5', '`b`' will get value 3 and `c` will get value '2'. You can see that the output has also changed according to the values passed to the function.

**Keyword Arguments:** When the arguments are passed as keyword arguments, the name of the argument is given along with the value. In this case, the position of the arguments does not matter. It can be written in random order.

Let us consider Example 6. Now just observe the arguments passed to the function `val_expression()` during the call.



Notes

**Example 6**

**#Function definition**

```
def val_expression(a,b,c):
    result=(a+b+c) * (a-b-c) * (a*b*c)
    print result
```

**#Function call**

```
>>>val_expression(a=12,b=6,c=2)
>>>val_expression(b=6,a=12,c=2)
>>>val_expression(c=2,a=12,b=6)
```

**Output**

11520  
11520  
11520

In the above example, the val\_expression() function is called with the names of the arguments. The output of the three function calls is same. The function call of val\_expression (a=12,b=6,c=2), val\_expression(b=6,a=12,c=2) and val\_expression(c=2,a=12,b=6) will give the same result. This is because position of the arguments does not matter in the case of keyword arguments.

**Passing List as an Argument**

You can send any type of data as an argument to a function (string, tuple, number, list, dictionary etc.).

Let us see an example of passing a list to the function.

**Example 7**

To display all the elements of a list is passed to the function as a parameter.

| FunctionCode  | FunctionCall                                       | Output                |
|---|--|-----------------------|
| def dispList(vowels):<br>for x in vowels:<br>print(x) | >>>list=['A','E','I','O','U']<br>>>>dispList(list) | A<br>E<br>I<br>O<br>U |



## Notes

**Function Parameters with Default Values**

You can also provide a default value to a parameter by using the assignment operator(=). When the default value is provided to the parameter, it is not mandatory to pass its value during the function call. In case, if you provide new value to it, then it will overwrite the default value.

Let us see an example of a function parameters with default values.

**Example 8**

```
def calc(length, breadth=10):  
    print("area=", length*breadth)  
    print("peri=", 2*(length+breadth))
```

**#Function call**

```
>>>calc(5,7)  
area= 35  
peri= 24  
  
>>>calc(5)  
area= 50  
peri= 30
```

In Example 8, calc() function has two parameters; namely length and breadth. The breadth parameter has a default value 10. Here, during first function call, the arguments 5,7 will be mapped to the parameters length and breadth respectively. The length parameter will take the value 5 and breadth will take the value 7 (instead of its default value 10). In the second function call, only one argument has been passed. So, the length parameter will take value 5 and breadth parameter will take the default value 10.

In a function, more than one parameter can have default values. **But once you have assigned a default value to one parameter, the parameters written after that parameter will also have default values. Otherwise, the program will throw an error message.**

```
def calc(length=10, breadth):  
    print("area=", length*breadth)  
    print("peri=", 2*(length+breadth))
```

In the above function, the default parameter i.e., length is followed by a non-default parameter i.e., breadth. This will throw an error message as shown in Fig. 11.6.

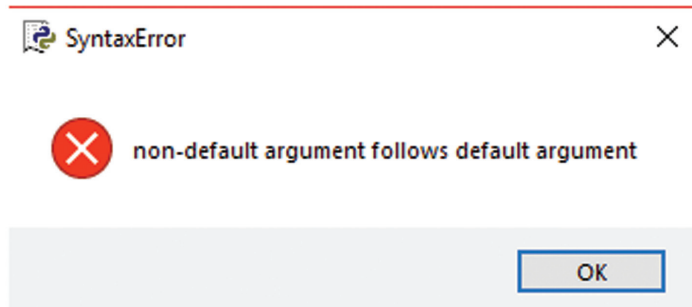


Fig. 11.6: ErrorMessage



Notes

### Returning from a Function

You can use a return statement to return a value from a function. The return statement may contain a constant/literal, variable, expression or a function. If return statement is used without any value, it returns 'None'. When a function returns a value, the function call is either an assignment statement or a display statement.

Let us see an example.

#### Example 9

The following function will check the number passed to it as a parameter and display whether it is positive or negative.

| FunctionCode  | FunctionCall   | Output   |
|---|--|----------|
| <pre>def identify_num(num):     if num &gt;= 0:         return "Positive"     else:         return "Negative"</pre> | <pre>&gt;&gt;&gt; print(identify_num(-3)) #function call1 OR &gt;&gt;&gt; k = identify_num(-3) #function call2 &gt;&gt;&gt; print(k)</pre> | Negative |

In the above example, the first function call (#function call1) worked as a display statement (the function was called first and then print statement is executed which displays the result) and the second function call (#function call2), worked as an assignment statement (because the return value will be assigned to another variable k i.e., k=identify\_num(-3)).



Notes



**INTEXT QUESTIONS 11.1**

1. What is the output of the following code?

```
def display(msg):
    print(msg, 'is a habit.')
display('Creativity')
```

- (a) is a habit
- (b) Creativity is a habit.
- (c) is a habit. Creativity
- (d) Creativity

2. In Python, which keyword is used to define a function?

- (a) function
- (b) def
- (c) import
- (d) try

3. Mention the command to call the following function:

```
def dispMesg():
    num=5
print(num+5)
```

- (a) dispMesg
- (b) dispMesg(num)
- (c) dispMesg()
- (d) defdispMesg()

4. Give the output of the following code:

```
def calc(num):
    if num%2 == 0:
        print(num+5)
    else:
        print(num*5)
calc(7)
```

- (a) 7
- (b) 35
- (c) 12
- (d) 0

5. What will be the output of the following Python code?

```
def func():
    return num *5
num = 5
print(func())
```

- (a) 5
- (b) 25
- (c) error
- (d) 10





Notes

### 11.3.3 Python Modules

A Python module can be defined as a Python program file, saved with an extension (.py). It contains a Python code including Python functions, class, or variables. To use the functions defined inside the modules, you need to import the specific module. Once you import the module, you can use any of its functions or variables in your own code. Some of the most commonly used modules are math, random, string, time, date etc. In this lesson, you will learn about math, random and statistics modules.

The syntax for using modules in the program is:

#### Syntax

```
import module1 [,module2 -----]
```

#### Example

```
import math
```

The import statement tells Python that you want to use the given module in your program. When Python executes the import math statement, it looks for the math module. If the module is found, then the statements in the body of that module will run and the module is made available for you to use. The import statement needs to be given only once in the beginning.

Let us discuss about the functions available in math Module.

#### Math Module

This module provides access to the most widely used mathematical functions. Some of them are listed in Table 11.2 along with their purposes.

**Table 11.2: List of Functions in Python Math Module**

| Function Name | Purpose   | Example   |
|---------------|---|---|
| math.ceil(x)  | It returns the ceiling of x, i.e.; the smallest integer value greater than or equal to x. | <pre>&gt;&gt;&gt;x=15.678 &gt;&gt;&gt;math.ceil(x) 16</pre> <p>Here x=15.678, math.ceil(x) will return the smallest integer value greater than or equal to 15.678, i.e.,16.</p> |

## MODULE-2

Computer Applications  
(712)



Notes

### Functions in Python

|                |   |  |
|----------------|---|--|
| math.floor(x)  | It returns the floor of x; the largest integer value less than or equal to x.   | <pre>&gt;&gt;&gt;x=15.678 &gt;&gt;&gt;math.floor(x) 15</pre> <p>Here x=15.678, math.floor(x) will return the largest integer value less than or equal to x. i.e., 15 will be returned.</p> |
| math.exp(x)    | It returns e**x.<br>It calculates the exponential value with the base set to e. | <pre>&gt;&gt;&gt; x=2 &gt;&gt;&gt;math.exp(x) 7.38905609893065</pre> <p>It returns the value of e<sup>2</sup>.</p>   |
| math.log10(x)  | It returns the base 10 logarithm of x.  | <pre>&gt;&gt;&gt; math.log10(1) 0.0</pre> <p>It computes the logarithm base10 of 1.</p>  |
| math.pow(x, y) | It returns the value of x raised to the power of y i.e., x <sup>y</sup> .       | <pre>&gt;&gt;&gt;math.pow(2,3) 8</pre> <p>Here in this example it returns the value of 2<sup>3</sup>= 8.</p>   |

### Random Module

This module provides the functions to generate random numbers. Some of the most widely used functions of this module are given in Table 11.3.

**Table 11.3: List of Functions in Python Random Module**

| FunctionName     | Purpose   | Example   |
|------------------|---|---|
| random.random()  | It returns a floating point random number in the range 0.0 to 1.0 (upper limit not included). It does not accept any arguments. | <pre>&gt;&gt;&gt;random.random() 0.30833599997701666</pre>  |
| random.randint() | It returns a random integer between the two numbers a and b (including both limits) where a should be less than b.              | <pre>&gt;&gt;&gt;random.randint(2,8) 3</pre> <p>Here the randint() function returns a random integer between 2 and 8.</p> |



|                    |  |   |
|--------------------|--|---|
| random.randrange() | It returns a random integer in the given range. It accepts three arguments as range(start, stop [,step]), step is optional with the default value 1. | <pre>&gt;&gt;&gt;random.randrange (1,10,3) 1 Here randrange() function returns an integer in the range 1,10 with the step of 3.</pre> |
|--------------------|--|---|

It is mandatory to import random module, before using the above functions.

**Statistics Module**

This module provides access to the statistical functions such as mean(), median(), mode() etc. Some of the popular statistics functions and their purpose are listed in Table 11.4.

**Table 11.4: Statistics Module**

| Function Name       | Purpose  | Example  |
|---------------------|--|--|
| statistics.mean()   | It accepts a list/tuple of numbers and returns average/mean value.   | <pre>&gt;&gt;&gt;statistics.mean ([10,12,15,18,20]) 15</pre>   |
| statistics.mode()   | It accepts a list/tuple and returns the element appearing most frequently in it. If all the elements are unique in the list/tuple an error will be generated.  | <pre>&gt;&gt;&gt;statistics.mode ([2,3,4,2,2,3]) 2 Here mode([2,3,4,2,2,3]) function will return the element appearing most frequently in it. In this case, number 2 is the most frequently appearing element in the list. 2 will be returned.</pre> |
| statistics.median() | It accepts a list/tuple of numbers. It returns middle value of the data. If number of elements is odd, then middle value will be returned otherwise the average of two middle values will be returned. | <pre>&gt;&gt;&gt;statistics.median ([1,2,4,6,9]) 4 Here the number of elements is odd i.e., 5. So it will return the middle value from the list of elements. Number 4 will be returned.</pre>  |



Notes

|  |  |   |
|--|--|---|
|  | <p>Note: It is important that the numbers are sorted before you can find the median.</p> | <pre>&gt;&gt;&gt;statistics.median ([4,5,6,8]) 5.5</pre> <p>Here the number of elements is even i.e., 4. So it will return the average of second number and third number (5 and 6) value from the list of elements.</p> |
|--|--|---|

It is mandatory to import statistics module, before using the above functions.

### 11.4 GLOBAL AND LOCAL VARIABLES

You can declare the variables anywhere in the program. The variables declared outside of any function are known as '**global variables**'. The global variable can be accessed in any function in the program.

Let us see an example, using global variable.

#### Example 13

```
a=10
def calc():
    print("inside function", a)
calc()
print("outside function", a)
```

#### Output

```
inside function 10
outside function 10
```

#### Local Variable

The variable declared inside a function is known as '**local variable**'. These variables can be accessed only inside the function in which it is declared.

Let us see an example using both global and local variable.

#### Example 14

```
a=10
def calc():
    x=20
```



Notes

```
print("Local variable:", x)
print("Global variable:", a)
calc()
print("Global variable:", a)
#print("Local variable:", x) is an invalid statement
```

**Output**

Local variable:20  
Global variable:10  
Global variable:10

In the above program, the variable ‘a’ is a global variable and the variable ‘x’ is a local variable. The local variable ‘x’ is visible and accessed only inside the function calc(), in which it is defined as shown in Fig 11.7. If you give a print statement: i.e., print(x), outside the function calc(), the program will throw an error. But the global variable ‘a’ is accessible in the entire program.

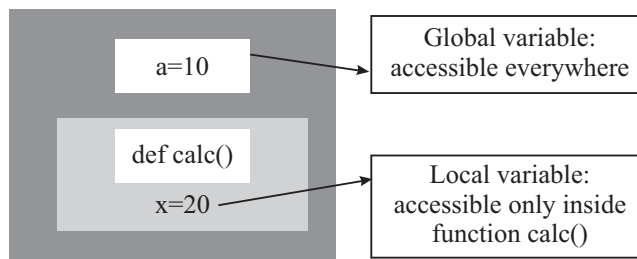


Fig. 11.7: Global and Local variables

Consider the following example:

**Example 15**

```
a=10
def calc():
    a=a*2    # variable a is considered as local
    print("inside function", a)
calc()
print("outside function", a)
```

**Output**

Traceback (most recent call last):



### Notes

File “C:/Users/Vineeta/AppData/Local/Programs/Python/Python38-32/g11.py”,  
line 6, in <module>

```
calc()
```

File “C:/Users/Vineeta/AppData/Local/Programs/Python/Python38-32/g11.py”,  
line 3, in calc

```
a=a*2
```

UnboundLocalError: local variable ‘a’ referenced before assignment

The above program, will throw an error because the variable ‘a’ is considered as local in the function calc(). Any variable which is declared inside the function is considered as local variable. If a variable is assigned with a value or its value is modified anywhere within the function’s body, it’s assumed to be a local variable. If you want to use the global variable, you have to use the keyword ‘**global**’ before the name of the variable.

### Example 16

```
a=10
```

```
def calc():
```

```
    global a # variable a is considered as global now
```

```
    a=a*2
```

```
    print(“inside function”, a)
```

```
calc()
```

```
print(“outside function”, a)
```

### Output

```
inside function 20
```

```
outside function 20
```

Any changes or modifications done to the global variable will be permanent in nature and visible everywhere.

You have seen in the above example, that the value of global variable ‘a’ is changed to the value 20 inside the function calc(). The value of the global variable ‘a’ is changed permanently and even outside the function calc().



**INTEXT QUESTIONS 11.2**



Notes

1. Write the Python expression for the following:
  - (i)  $-x/y + (1+2a)^4 - xy$
  - (ii)  $a + (2x-y)/3x$
2. To include the use of the functions which are present in the math library, you must use the ..... option.
  - (a) math import
  - (b) maths import
  - (c) import math
  - (d) import.math
3. Which of the following cannot be returned by random.randrange(5)?
  - (a) 0
  - (b) 3
  - (c) 2.3
  - (d) 2
4. What will be the value returned by math.pow(3,2.0)?
  - (a) 3
  - (b) 9
  - (c) 3.0
  - (d) 9.0
5. What will be the value returned by statistics.mode([12,14,18,14,20,20,14])?
  - (a) 12
  - (b) 20
  - (c) 14
  - (d) 18
6. What will be the output of the following Python code?
 

```
def f1():
    global x
    x+=1
x=10
f1()
print(x)
```

  - (a) 11
  - (b) 10
  - (c) error
  - (d) 12
7. What will be the output of the following Python code?
 

```
x=100
def f1():
    global x
    x=90
def f2():
    x=80
f1()
f2()
print(x)
```

  - (a) 100
  - (b) 90
  - (c) 80
  - (d) error

## MODULE-2

Computer Applications  
(712)



Notes



### WHAT YOU HAVE LEARNT

- The function is a named group of related programming statements, which perform a specific task.
- Python functions can be divided into two basic types: Built-in and User defined.
- There is no need to import any module (file) to use Python built-in functions.
- The user defined functions start with a keyword def.
- A Python module can be defined as a Python program file, saved with an extension (.py). It contains a Python code including Python functions, class, or variables.
- The information can be passed to functions as parameters. The values provided in the function call are called as '**arguments**'.
- When the arguments are passed as positional arguments, the order in which the arguments are sent or their position is very important.
- When the arguments are passed as keyword arguments, the name of the argument is given along with the value. In this case, the position of the arguments does not matter.
- You can also provide a default value to a parameter by using the assignment operator (=). When a default value is provided to a function it is not mandatory to pass its value during a function call. If the value is provided, it will overwrite the default value.
- A variable declared outside of the function is known as '**global variable**'. A variable declared inside the function is known as '**local variable**'.



### TERMINAL EXERCISE

1. What are functions? What is their need?
2. What are Python modules and how they are used in Python script?
3. How are the built-in functions different from Python modules?
4. What is the difference between keyword and positional arguments?





5. Explain the difference between global and local variable with the help of an example.
6. Write a Python function that takes a number as an argument and print its square.
7. Write a Python function `calc_pow(num1 , num2)` that calculates the power of `num1` to `num2`.
8. Write a Python function that takes a list of numbers as parameters and displays those numbers that are divisible by 100.
9. Write a Python function that takes a list of numbers as arguments and returns the sum of all even numbers.
10. Write a Python function that takes a string as an argument and return the number of vowels present in it.



**ANSWERS TO INTEXT QUESTIONS**

**11.1**

1. (b) Creativity is a habit.      2. (b) `def`      3. (c) `dispMesg()`
4. (b) 35      5. (b) 25

**11.2**

1. (i) `-x/y + math.pow((1+2*a), 4) - x*y`  
(ii) `a+ math.sqrt((2*x - y)/3*x)`
2. (c) `import math`      3. (c) 2.3      4. (d) 9
5. (c) 14      6. (a) 11      7. (b) 90

**Key Learning Outcomes**

- Create user-defined functions in the program and use it efficiently.
- Be proficient in writing error free programs in Python using functions effectively.



## STACKS AND QUEUES

---

### 12.1 INTRODUCTION

---

In the previous lesson, you have learnt about List and the various operations that can be performed on the Lists. Moving one step further, in this lesson, you will learn about two very important data structures viz., Stacks and Queues. Data structure is a format for organizing, processing, retrieving and storing data. The Stacks and Queues allow us to store and retrieve the data sequentially. Also, these can be used to solve various problems. Although Python does not have a built-in specific classes/interfaces for Stack and Queue type, but you can think of a Stack and queue as a constrained List. The List methods make it very easy to use a list as a Stack and Queue. In this lesson, you will learn about Stacks and Queues and their implementation in Python by using Lists



### OBJECTIVES

After reading this lesson, you will be able to:

- define a Stack and a Queue;
- perform insertion and deletion operations on Stacks and Queues;
- implement Stacks and Queues using Lists.

---

### 12.2 STACK

---

A stack is a linear data structure that provides temporary storage of the data in such a way that the element stored at the last will be the first one to be retrieved. This method is also called **Last In First Out (LIFO)**. In real life, you can think of stack as a stack of copies, stack of plates, stack of chapatis etc., (refer to Fig 12.1).



Stack of Plates



Stack of Coins



Stack of Books

Fig 12.1 Examples of Stack in Real Life

The first plate we put on the stack of plates is the last one to be removed. Similarly, the last plate we put on the Stack is the first one to be removed from the Stack. Think of some more examples of Stack that you have seen in real life. Let us learn some operations that can be performed on Stacks.

### Operations on Stack

The Stack is controlled by two operations: **'push'** and **'pop'**. Both these operations take place from one end of the Stack, usually called **'top'**. The push operation adds an item (element) in to the top of the Stack. The pop operation removes an item from the top of the Stack. These two operations (Push and Pop) implement the **LIFO** (Last In First Out) method. In the LIFO method, the element which is added at the end of the stack is the first one to be removed from the stack.

After learning the operations that can be performed on Stacks, let us now learn to write the code in Python to implement a Stack.

### 12.3 IMPLEMENTATION OF STACK AS A LIST

The Stack can be easily implemented in Python by using the methods of List. The basic operations that can be performed on a Stack are:

1. **Creation of Stack:** Initially the Stack will be empty i.e., there is no element in the Stack. An empty Stack can be created using List by using the following statement.

```
stack= [ ]
```

The above statement creates an empty Stack **'stack'**.

2. **Push (Insertion) Operation In The Stack:** This operation, as shown in Fig. 12.2, takes place with the help of `append ()` function. This function refers to inserting an element in the Stack i.e., at the top of the Stack.



Notes





Notes

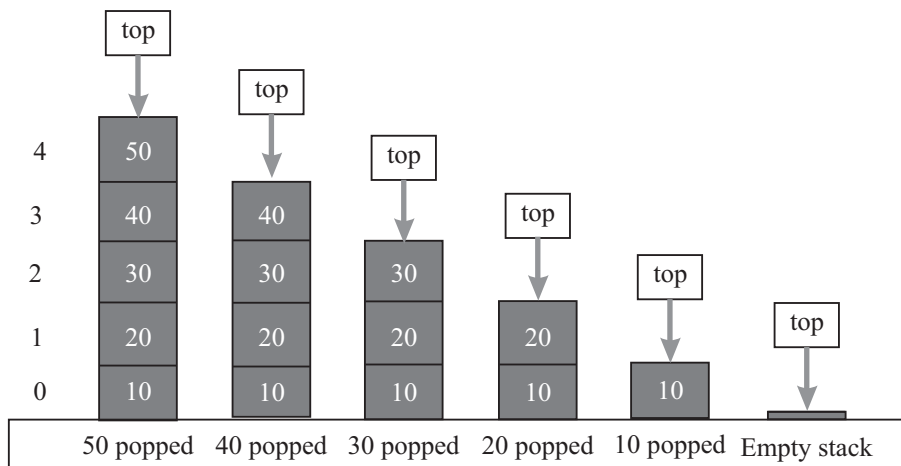
Stack has five elements: **10, 20, 30, 40,** and **50** with **50** as the top most element of the Stack as shown at Fig. 12.3. Let us now try to delete an element from the Stack.

To delete an element from the Stack, the following Python statement will be used:

```
stack.pop()
```

As you can see from Fig. 12.3, the element **50** is deleted from the Stack. This happens because `pop()` function deletes the top most element from the Stack. The next `pop()` operation results in the deletion of the element **40** from the Stack.

Similarly, as you apply more `pop()` operations on the Stack, more elements from the top of the Stack will be deleted. (refer Fig. 12.3).



**Fig. 12.3: Pop Operations in Stack**

- The `pop()` operation refers to the removal of an element. Only one element can be removed at a time using `pop()` operation and that too the top most element. To remove more than one element call the `pop()` operation again.
- Traversal of Stack:** Traversal implies accessing each and every element of the Stack and displaying them. Before displaying the elements, first you have to calculate the length of the Stack and using a looping statement you can access each element of the Stack. Let us see an example.

```
length=len(stack)
for i in range(length-1,-1,-1):
    print (stack[i])
```

Here `len()` function calculates the length of the Stack which is stored in the variable '**length**'. After this operation, **for** loop is used to fetch each element



from length-1 to 0 and display the Stack element using a print statement. To access the elements of the Stack, index values are used. The first element of the Stack is at index 0 i.e., stack[0].

Let us now write a complete Python script to implement Stack as a List.

```
stack=[]                # Creation of empty Stack
def push(x):
    stack.append(x)     # pushing an element into the Stack
def pop():
    if (stack==[]):    # checking for empty Stack
        print ("Stack Empty" )
    else:
        print ("Deleted element is : ",stack.pop())
        # deleting an element from the Stack
def display():
    length=len(stack)  # calculating length of the Stack
    for i in range(length-1,-1,-1): # traversing each element of the Stack
        print (stack[i])
ans='y'
while (ans=="y"):
    print ("\n 1. PUSH")
    print ("2. POP ")
    print ("3. Display")
    choice=eval(input("Enter your choice: "))
    if (choice==1):
        a=int(input("Enter any number :"))
        push(a)
    elif (choice==2):
        pop()
    elif (choice==3):
        display()
else:
    print("Invalid Input")
ans= input("Do you want to continue? y or n: ")
```



**Notes**

**Output**

1. PUSH

2. POP

3. Display

Enter your choice: 1

Enter any number :23

Do you want to continue? y or n: y

1. PUSH

2. POP

3. Display

Enter your choice: 1

Enter any number :45

Do you want to continue? y or n: y

1. PUSH

2. POP

3. Display

Enter your choice: 1

Enter any number: 67

Do you want to continue? y or n: y

1. PUSH

2. POP

3. Display

Enter your choice: 3

67

45

23

Do you want to continue? y or n: y



### Notes

```
1. PUSH
2. POP
3. Display
   Enter your choice: 2
   Deleted element is : 67
   Do you want to continue? y or n: y
```

```
1. PUSH
2. POP
3. Display
   Enter your choice: 3
   45
   23
   Do you want to continue? y or n: n
```

In the above example, we have created three functions

- (i) push(),
- (ii) pop() and
- (iii) display().

The push() function in the program will call the append() function and add the element(s) in the Stack. The pop() function, will check whether the Stack is empty or not. If the Stack is empty, then it will display that the Stack is empty. If the Stack is not empty, then it will use stack.pop() and remove the last element from the list. The display() function will calculate the length of the Stack and using for loop, it displays all the elements from the list.

When you run this program, first it gets an input from the user for requesting the operation to be performed. If the user chooses the option 1, then it will perform push() operation, if the user chooses option 2, then it will perform pop() operation. If the user chooses option 3, then the program will perform the display() operation. If the user chooses any other option except 1, 2 or 3, then it will display an error message **'it is an invalid input'**.

Once it has completed the task of push, pop or display then it will ask the user, **'Do you want to continue?'**, if the user gives **'y'**, then it will perform the above process once again. If the user gave **'n'**, then it will stop the execution of the program.





Notes

## 12.4 APPLICATIONS OF STACK

You have learnt the push and pop operations in Stack and implementation of Stack in Python. But you might be wondering how the computer uses this Stack. Let us now learn some important ways in which Stack is used as mentioned below by the computer:

- The Stack is used to store the return address of the program at the time of function call. After the execution of the function, the return address is popped from Stack and the control is returned back to the calling function.
- Converting an infix expression to postfix and to evaluate the postfix expression. Consider an arithmetic expression such as  $A * B$ . This type of notation is referred to as **infix** since the operator '\*' is in between the two operands ('A' and 'B'). In postfix the same expression will be denoted as  $AB *$ .
- Reversing an array (array is a data structure that contains groups of the same data type elements), converting decimal numbers into binary numbers etc.



### INTEXT QUESTIONS 12.1

1. The Stack works on the ..... principle.
 

|          |                   |
|----------|-------------------|
| (a) LILO | (b) FIFO          |
| (c) LIFO | (d) None of these |
2. Deleting an element from a Stack is called .....
 

|          |          |
|----------|----------|
| (a) LIFO | (b) FIFO |
| (c) pop  | (d) push |
3. An user performed the following operations on Stack:
 

```
push(1);
push(2);
push(3);
pop();
push(4);
pop();
pop();
push(5);
```



**Notes**

pop();

pop();

push(6);

What will be the contents of the Stack when it is displayed?

- (a) 123456
- (b) 6
- (c) 654321
- (d) 65

4. The five items: A, B, C, D, and E are pushed in a Stack, one after other starting from A. Four items are popped from the Stack and F and G are pushed in the Stack. Now one item is popped from the Stack. The popped item is
  - (a) G
  - (b) F
  - (c) A
  - (d) C
5. Which of the following is not an application of Stack?
  - (a) Implementation of recursion.
  - (b) Evaluation of a postfix expression.
  - (c) Job scheduling.
  - (d) Reverse a string.

**12.5 QUEUE**

A Queue is a linear data structure that provides temporary storage of data in such a way that the first element stored in the Queue will be the first one to be retrieved. This method is also called as **First In First Out (FIFO)**.

In real life, you can think of Queue as a line of vehicles waiting at a petrol pump or line of people waiting at the bus stop etc. (refer to Fig. 12.4).



**Fig. 12.4: Real Life Examples of Queue**

The first person to enter the queue is the first one to leave the queue. Similarly, the last person to join the queue is the last person to leave the queue. Let us learn some operations that can be performed on queues.



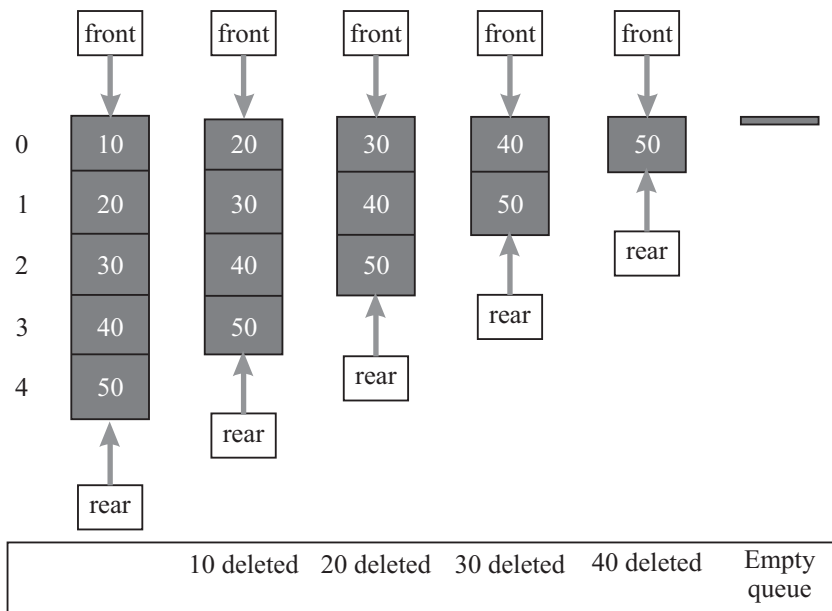


**Notes**

Here, **10** will be added to the queue as shown in Fig. 12.5. When the first element, **10** is added to the queue, the number **10** is the first and the last element. In this case, both front and rear points to the same element. When the next element **20** is added to the queue, the rear end points to the element **20** and the front end points to the element **10**. In the same way, the elements **30**, **40** and **50** will be added to the Queue. Whenever, an element is added to the queue, the front element will remain the same, but the rear element will change.

- 3. Deletion From The Queue:** The deletion operation as shown in Fig. 12.6 takes place with the help of pop() function. The pop() function removes the last element. To delete the first element from the queue implemented as a List we use pop(0). This will delete the first element from the List. Before removing an element from the queue, it is necessary to check whether the queue is empty or not. To check whether the queue is empty or not, you can use the statement :

```
if( queue== [])
    print("Queue is Empty")
```



**Fig. 12.6: Deletion Operation on Queue**

Here, **'queue'** is the name of the Queue. To understand the deletion operation, let us reconsider the queue, which was created during the insertion operation. The queue has five elements: **10, 20, 30, 40, 50** with front end pointing to the



## Notes

element **10** of the queue and rear end pointing to the element **50** as shown in Fig. 12.6. Let us now try to delete an element from this queue. To delete an element from the queue, the following Python statement will be used.

```
queue.pop(0)
```

Here, as you can see from Fig. 12.6, element **10** is deleted from the queue. This happens because `pop(0)` function deletes the top most element i.e., element present at position 0 from the queue. After deletion, the element **20** will occupy the top position. The next deletion operation results in the deletion of element **20** from the queue.

In all these cases, the rear end remains unchanged. Similarly, as you perform more deletion operations on the queue, more elements from the top of the queue will be deleted. (refer Fig. 12.6).

If only one element is left in the queue, then front end and rear end points to the same element. The deletion of this element from the queue makes the queue empty.

- 4. Traversal of Queue:** Traversal implies traversing elements of the queue and displaying them. Thus, to perform the display operation, you first calculate the length of the queue and use a looping statement to display each element of the queue.

For example:

```
length=len(queue)
for i in range(0,length,1):
    print (queue[i])
```

Here `len()` function calculates the length of the queue which is stored in the variable '**length**'. After this, **for** loop is used to access each element of the queue from 0<sup>th</sup> position to length-1 position and display it using `print` statement. To access the elements of the queue the index values are used. The first element of the queue is at index 0 i.e., `queue[0]`. The second element will be at index 1 i.e., `queue[1]` and so on.

Let us now write a complete Python program to implement Queue as a list.

### Sample Program to Implement Queue As A List

```
queue=[] # Creation of empty queue
```



## Notes

```

def insertion(x):
    queue.append(x)           # inserting an element into the queue
def deletion():
    if (queue==[]):          # checking for empty queue
        print ("Queue Empty")
    else:
        print ("Deleted element is : ",queue.pop(0))
                                # deleting an element from the queue
def display():
    length=len(queue)        # calculating length of the queue
    for i in range(0, length,1): # traversing each element of the queue
        print (queue[i])
    ans='y'
    while (ans=="y"):
        print( "1. Insertion")
        print ("2. Deletion")
        print ("3. Display")
        choice=eval(input("Enter your choice: "))
        if (choice==1):
            a=input("Enter any number :")
            insertion(a)
        elif (choice==2):
            deletion()
        elif (choice==3):
            display()
    else:
        print("Invalid Input")
    ans= input("Do you want to continue? y or n: ")

```

### Output

1. Insertion
  2. Deletion
  3. Display
- Enter your choice: 1  
Enter any number :10

**Notes**

Do you want to continue? y or n: y

1. Insertion
2. Deletion
3. Display

Enter your choice: 1

Enter any number: 20

Do you want to continue? y or n: y

1. Insertion
2. Deletion
3. Display

Enter your choice: 3

10

20

Do you want to continue? y or n: y

1. Insertion
2. Deletion
3. Display

Enter your choice: 2

Deleted element is : 10

Do you want to continue? y or n: y

1. Insertion
2. Deletion
3. Display

Enter your choice: 3

20

Do you want to continue? y or n: n

In the above example, three functions are created. Those are:

- (i) insertion(),
- (ii) deletion() and
- (iii) display().

The insertion() function calls queue.append() for adding the elements in the queue. The deletion() function uses queue.pop() for deleting the elements in the queue. The display() function uses for loop for fetching the elements in the queue and uses print statement for displaying those elements.



**Notes**

When you run the program, it asks for the input from the user. If the user gives option 1 then it will insert the element in a queue. If the user gives option 2, then it will delete the elements from the queue, if the user gives option 3, then it will display all the elements from the queue.

You can observe the output of the program. Initially, the option 1 is given for inserting **10, 20** and **30** to the queue. When the option 2 is chosen, then the first element, which was inserted in the queue i.e., **10** will be removed from the queue. At any point of time, if the user is given option 3, then all the elements from the queue will be displayed.

**Applications of Queue**

Let us now learn some important ways in which queue is used.

- Queues find their use in CPU scheduling, printer spooling, message queuing, computer networks etc.
- In time sharing systems (the system allows multiple users to share the computer resources simultaneously), the queues help in scheduling of jobs.



**INTEXT QUESTIONS 12.2**

1. Inserting an element to a queue takes place from ..... end.
 

|            |          |
|------------|----------|
| (a) Front  | (b) Rear |
| (c) Middle | (d) Any  |
2. User performed the following operations on a queue:  
 Inserted 10, 20 and 30. Then the user deleted one element and inserted 40.  
 Then again he deleted two elements and inserted 50 into the queue.  
 What will be the contents of the queue, if you try to display it?
 

|              |              |
|--------------|--------------|
| (a) 30 40 50 | (b) 20 30 40 |
| (c) 50 40    | (d) 10 50    |
3. Which of the following is not an application of queue?
 

|                      |                       |
|----------------------|-----------------------|
| (a) Printer spooling | (b) computer networks |
| (c) Job scheduling   | (d) Reverse a string  |





### WHAT YOU HAVE LEARNT

- A Stack is a linear data structure that provides temporary storage of data in such a way that the element stored in the last will be the first one to be retrieved.
- Stack works on a **LIFO** principle and is controlled by two operations: **push** and **pop**.
- The push operation adds an element to the top of the Stack.
- The pop operation removes an element from the top of the Stack.
- A Queue is a linear data structure that provides temporary storage of data in such a way that the element stored as first will be retrieved first.
- A Queue works on a **FIFO** principle and is controlled by the insertion and deletion operations.
- An insertion operation adds an element to the queue and takes place from the rear end of the queue.
- The deletion operation removes an element from the queue and takes place from the front end of the queue.



### TERMINAL EXERCISE

1. What is a Stack? Why is it called a LIFO structure?
2. Implement a Stack using list in Python containing the names of the students.
3. What will be the output of the following program?
 

```
st = ["Python", "Programming", "Language"]
st.push("Highlevel")
st.push("dynamic")
print(st)
print(st.pop())
print(st.pop())
print(st)
```
4. What is a queue?
5. What will be the output of the following code?
 

```
que=[]
```



Notes



Notes

```
que.append(9)
que.append(10)
que.append(11)
que.append(12)
que.pop(0)
que.pop(0)
print(que(0))
```

6. Write a Python program to input a string and display it in reverse order using a stack.
7. Write a user defined function which takes stack as a list as parameter and return 1 if it is empty otherwise return 0.
8. Write a user defined function which takes queue as a list as parameter and return 1 if it contains only one element otherwise return 0.
9. Implement a queue using list in Python containing the names of the cities.
10. Write a menu driven program to implement a queue as a list containing student name and total.



## ANSWERS TO INTEXT QUESTIONS

### 12.1

1. (c) LIFO
2. (c) pop
3. (b) 6
4. (a) G
5. (c) Job Scheduling

### 12.2

1. (b) Rear
2. (d) 10 50
3. (d) Reverse a string

### Key Learning Outcomes

- Be proficient in performing the push and pop operations in Stack.
- Be able to insert and delete the elements in queues.



# 13

## FILE HANDLING IN PYTHON

---

### 13.1 INTRODUCTION

---

In the previous lessons, you have seen how to manipulate values stored in different variables of different data types. These variables store data temporarily in the Random-Access Memory (RAM). The stored data remains there in the memory as long as the program is running, and as soon as the program gets over, this data also gets erased from the memory. That's the major issue. We need to store the data permanently for future use. Python allows to store the data permanently in the form of files on the secondary storage devices (hard disk or even cloud). The data, which is stored on the secondary storage devices is permanent in nature and does not get erased, when the program gets over. When you perform the input/output operations on the files, the data gets transferred from the secondary storage to RAM and vice versa. In this lesson, you will learn about file handling, in Python.



### OBJECTIVES

After reading this lesson, you will be able to:

- list the need of the data file;
- distinguish between the text and the binary file;
- open and close the files;
- use different methods for writing the data in a file and
- read and write the data in files (text and binary).



### Notes

### 13.2 INTRODUCTION TO FILES

A file is a named location on a secondary storage device, which contains related information. You might be familiar with different kinds of files such as image files, video files, music files, and text files etc. Python supports following types of files for data storage:

- (a) text files
- (b) binary files

The text files are the sequence of lines of text, where each line includes a sequence of characters. All the lines are terminated by a special character, called the **End of Line (EOL)** character. Usually the comma ‘,’ or the new line character ‘\n’ is considered as the EOL character. It ends at the current line and tells the interpreter that a new line has begun. The text files are stored in human readable form and these can also be created using any text editor.

The binary files deal with non-text files such as images or executable files and store the data in the binary format i.e., in the form of 0’s and 1’s. It is treated as raw data. So, when you work on a binary file; the raw bit pattern(s) read from the file that needs to be interpreted into the correct type of data in the program. Python provides, special module for encoding and decoding of the data stored in a binary file. The binary files are not stored in the human readable form.

### 13.3 OPENING AND CLOSING FILES

Before performing any operation on the files, first, you need to open the file. After opening it, you can perform the desired operations and then the file needs to be closed.

#### Opening a File

There are two ways of opening a file in Python:

- `open()` function and
- with `open()` statement.

Let us discuss about these functions in detail.

#### `open()` Function

Python has a built-in function – `open()` to open a file. This function takes a file name and mode as two separate arguments.



## Notes

The syntax for using `open()` function is:

**Syntax**

```
file_object = open(filename [, mode])
```

Where, `filename`, is the name of the file to be opened. The name of the file can be a string constant or a variable. If the file is not present in the same folder/directory in which you are working, you need to give the complete path where the file is present. The mode is an optional parameter and it describes the type of operation you are going to perform on that particular file such as read, write, append etc. (refer Table 13.1). It is important to remember that the default mode used in `open()` function is reading mode. The `file_object` is the name of the object returned by the `open()` function. The file objects contain methods and attribute that can be used to collect the information about the file you have opened to manipulate it.

Let us see the list of Python file modes.

**Table 13.1: Modes of Python Files**

| Python FileModes |   |
|------------------|---|
| Mode             | Description   |
| 'r'              | Open a file for reading (default).  |
| 'w'              | Open a file for writing. Creates a new file, if it does not exist or truncates the file if it exists.                     |
| 'x'              | Open a file for exclusive creation. If the file already exists, the operation fails.                                      |
| 'a'              | Open a file for appending the data at the end of the file without truncating it. Creates a new file if it does not exist. |
| 't'              | Open a file in the text mode (default).   |
| 'b'              | Open a file in the binary mode.   |
| '+'              | Open a file for updating (reading and writing).   |

Let us see the examples of using different modes with brief description about them.



Notes

**Example 1**

| Python Statement                            | Description  |
|---|--|
| <code>file= open("test.txt","w")</code>     | It will open a text file called ' <b>test.txt</b> ' in the writing mode. You can write any content in the file if you have used the writing mode.  |
| <code>f = open("test.txt")</code>           | It will open a text file called ' <b>test.txt</b> ' in reading mode. If you didn't give any mode, by default the program will consider the mode as reading mode. The file content can be read if you have used the reading mode.                                       |
| <code>file= open("example","rb+")</code>    | It will open a binary file called ' <b>example</b> ' in reading and writing mode.  |
| <code>file= open("example.txt","a+")</code> | It will open a text file called ' <b>example.txt</b> ' in appending (writing at the end) and reading mode.   |
| <code>file= open("example","wb+")</code>    | It will open a binary file called ' <b>example</b> ' in reading and writing mode. The "w+" mode deletes all the contents present in the file first and initially there will be nothing to read. You first need to write something and then perform the read operation. |

Let us try to open the files using **with open()** statement.

**with open() Statement**

The syntax for using **with open()** is

**Syntax**

with open() as file object :

**Example 2**

with open("example.txt","r") as file:

*file manipulation statements*

Here, the file 'example.txt' is opened in reading mode.



The difference between opening a file using **open()** function and **with open()** statement is that using **with open()** ensures that once you finish using the file all the resources allocated to the particular file objects, gets de-allocated automatically i.e., it does the clean-up operation and also closes the file.

Let us discuss how to close the file.

### Closing A File

After performing the desired operations on the file, you need to close it. This can be done by using a `close()` method.

The syntax for using a `close()` method:

### Syntax

```
fileobject.close()
```

When you close a file, it frees up the resources that were tied with it. It is a good practice to close a file once you have finished using it. If you reassign the file object to some other file, Python will automatically close the file attached to it earlier.



### INTEXT QUESTIONS 13.1

1. State whether the following are True or False.
  - (a) A file opened in append mode is used for adding new data at the end of the file.
  - (b) A file opened in `r+` mode allows only reading the file.
  - (c) A file opened in `wb` mode opens the file in the text mode for writing the content into the file.
  - (d) A file opened in `w` mode erases the content of the file, if the file already exists.
  - (e) The data stored in the files are permanent in nature.
2. Write Python statements for the following:
  - (a) Open the text file “abc.txt” for reading purpose.
  - (b) Open the binary file “stu.dat” for both reading and writing purposes.
  - (c) Open the binary file “item.dat” to add new data at the end of the file.
  - (d) Open the text file “game.txt” for reading purpose using **with** statement.
  - (e) Close the text file “abc.txt” (use file object as *fileobj*).



Notes

**13.4 WRITING TO A FILE**

After opening a file in the ‘w’ writing mode, you can use **write()** and **writelines()** methods for writing the content in a file(refer Table 13.2).

**Table 13.2: The Methods for Writing the Content in File**

| Method              | Description   |
|---------------------|---|
| <b>write()</b>      | It takes a string as a parameter and writes the data in a file. It can also be used to write numeric values but numeric value needs to be converted to string, before storing it in the file. |
| <b>writelines()</b> | It takes a sequence of strings as a parameter and writes to a file.   |

Let us now understand the above functions with the help of examples.

**write() Method**

It takes a string as a parameter and writes it in the open file. The ‘\n’ character can be added at the end of the string for storing the data with end of line character.

The syntax for using write() method is:

**Syntax**

fileobject.write(string)

Let us see an example using write() method.

**Example 3**

| Python Code                                  | Explanation  |
|--|--|
| file = open(“poem.txt”, “w”)                 | Here the file ‘poem.txt’ is opened in writing mode.                                |
| file.write(“Twinkle twinkle little star,\n”) | Then write() method is used to write one line at a time in the file.               |
| file.write(“How I wonder what you are.\n”)   | (“\n” character will insert a new line break in the file.)                         |
| file.write(“Up above the world so high,\n”)  |  |
| file.write(“Like a diamond in the sky.\n”)   | Then the close() method is used to close the file after writing the content to it. |
| file.close()                                 |  |





Notes

### Writing List To File

You can also write a list (refer the lists in Python lesson) in an open file.

#### Example 4

| Python Code   | Explanation   |
|---|---|
| <pre>myList=["One", "Two", "Three", "Four", "Five"] file1 = open("text.txt", "w") for line in myList: # write line to output file file1.write(line) file1.write("\n") file1.close()</pre> | <p>Before writing a list to a file, first you need to create a list. Here '<b>myList</b>' is a list which contains the elements One, Two, Three, Four and Five. Then '<b>text.txt</b>' file is opened in the write mode. This returns a file object, '<b>file1</b>'. To write the content in the file line by line, for loop is used. The for loop fetches each element of <b>myList</b> and write the same element in the file. The close() method at the end closes the access to the file.</p> |

### Writing Numeric Values to File

To write the numeric values in the file, you need to first convert the numeric values to string as the write() method takes only string arguments.

Let us see an example.

#### Example 5

```
num = 185
file.write(str(num))
```

Here str() method is used to convert the numeric value to string.

#### writelines() Method

You can write a list/sequence of strings together to a file using writelines() method. It takes a sequence of strings as an input and writes to a file object that is opened.

The syntax for using writelines() method is:

#### Syntax

```
fileobject.writelines(list/sequence of strings)
```

Let us see an example of writelines() method.



Notes

**Example 6**

| Python Code   | Explanation   |
|---|---|
| <pre>myList=["One\n", "Two\n", "Three\n", "Four\n", "Five\n"] file = open('text.txt', 'w') file.writelines(myList) file.close()</pre> | <p>The list of elements which need to be written in the file is stored in the list variable <b>'myList'</b>. Then the file <b>'text.txt'</b> is opened in the writing mode. The entire content of <b>'myList'</b> variable i.e., the values, "One\n", "Two\n", "Three\n", "Four\n", "Five\n" will be stored in the file 'text.txt'.</p> |

**13.4.1 Appending to a File**

Suppose the file you are going to use already exist and you want to add some more content in the same file, in that case, you can open the file in an append mode. The new content will be added at the end of the file. If the file does not exist, then a new file will be created and the data will be added to the new file.

The syntax for opening the file in append mode is as follows:

**Syntax**

file\_object = open(filename, a)

where **'a'** stands for append mode (refer Table 13.1 for file modes).

Let us see an example of using the file in append mode.

**Example 7**

| Python code   |
|---|
| <pre>file = open("text.txt", "a") file.write("Python is a cross platform language.\n") file.write("It is easy to read and write.\n") file.close()</pre> |

In this example, **'text.txt'** file is opened in append mode. Then the two lines of content are added to it by using write() method. The following table shows the content of **'text.txt'** before and after append mode.



## Notes

Table 13.3: Content of the File Before and After using Append Mode

| Content of the File 'text.txt' Before Append Mode                                     | Content of the File 'text.txt' After Append Mode   |
|---|--|
| I am learning Python.<br>This is my first file handling program.<br>I am enjoying it. | I am learning Python.<br>This is my first file handling program.<br>I am enjoying it.<br>Python is a cross platform language.<br>It is easy to read and write. |

If you observe the second column of the Table 13.3, then you notice that two more lines are added to the existing content of the file 'text.txt'.

### Difference Between 'w' Mode and 'a' Mode

When an already existing file is opened in 'w' mode all the existing content will get erased from the file and only the new content will be saved in the file. When an already existing file is opened in 'a' mode the existing content will not get erased and the new content will be added at the end of the existing file.

If you open the 'text.txt' file in 'w' mode instead of 'a' mode in the above program, the file 'text.txt' will appear as follows:

### Example 8

#### Python Code

```
file = open("text.txt","w")
file.write("Python is a cross platform language.\n")
file.write("It is easy to read and write.\n")
file.close()
```

In this example, 'text.txt' is opened in the writing mode. The two lines of content is added into the text file by using write() method (refer Table 13.4).

Table 13.4: Content of Same File Before and After Opening in Write Mode

| Content of the File 'text.txt' Before Opening In Write Mode                           | Content of the File 'text.txt' After Opening In Write Mode            |
|---|---|
| I am learning Python.<br>This is my first file handling program.<br>I am enjoying it. | Python is a cross platform language.<br>It is easy to read and write. |



Notes

Here the file ‘**text.txt**’ is opened in the writing mode. The existing content of the file (refer the first column of Table 13.4) will get erased and the new content (refer the second column of Table 13.4) will be added to it.

**13.5 READING FROM A FILE**

The **readline()**, **readlines()** and **read (n)** methods are used to read the data from a file (refer Table 13.5).

**Table 13.5: Methods of Reading from File**

| Method      | Description  |
|-------------|--|
| readline()  | It reads one line(i.e. till newline) at a time from a file and returns that line.  |
| readlines() | It reads the entire content of the file in one go and returns it as <b>list of lines</b> .   |
| read(<n>)   | It reads a string of size (here n) from the specified file and returns it. If ‘ <b>n</b> ’ is not given, it reads and returns up to the end of the file. |

Let us now understand each of these methods with examples. The content of the file ‘**read\_file.txt**’ is given below. This file will be referred for all the examples given below:

I am learning Python.

This is my first file handling program.

I am enjoying it.

Now let us use the read methods one by one.

**readline() Method**

The syntax for using this method is:

**Syntax**

fileobject.readline()

It reads one line at a time from a file and returns that line. i.e., it reads the content from a file till it encounters a new line character (“\n” or “;”). Let us understand readline() method by following example.



Notes

**Example 9**

| Python Code   | Output   |
|---|--|
| <pre>file = open("read_file.txt", "r") x = file.readline() # reads first line print(x) #prints first line print(file.readline()) # reads and prints second line</pre> | <pre>I am learning Python. This is my first file handling program.</pre> |

The file will be read sequentially i.e., from the top. When the first time `readline()` method is called it returns the first line of the file '**read\_file.txt**'. The second call to the `readline()` method returns the second line. The `readline()` method returns an empty string when the end of the file has been reached.

**readlines() Method**

This method is used to read all the lines in one go and returns in the form of list of lines. It returns a list containing each line in the file as a list item.

The syntax for using `readlines()` method is:

**Syntax**

```
fileobject.readlines()
```

Let us see an example using `readlines()` method.

**Example 10**

| Python code  | Output  |
|--|---|
| <pre>file = open("read_file.txt", "r") x = file.readlines() print(x)</pre> | <pre>['I am learning Python\n', ' this is my first file handling program.\n', ' I am enjoying it.']</pre> |

In the above example, '**read\_file.txt**' is opened in read mode. Then the `readlines()` method is used to read the entire content of the file '**read\_file.txt**' and store it in the variable '**x**'. The `print(x)` statement will display the entire content of the file.

**read(n) method**

This method reads at most `n` bytes from the specified file and returns them. If the size parameter is not given or a negative value is specified as size, then it reads and returns up to the end of the file.



Notes

The syntax for using read() method is:

**Syntax**

```
fileobject.read([n])
```

Here **n** specifies the number of bytes to be read from the file.

Let us see an example using read() method.

**Example 11**

| Python code                      | Output                           |
|----------------------------------|----------------------------------|
| file = open("read_file.txt","r") | I am # reads first 5 bytes       |
| print(file.read(5))              | learning P # reads next 10 bytes |
| print(file.read(10))             |                                  |

Here in the above example, **'read\_file.txt'** is opened in read mode. Then read() method is used to read the content from the file. The statement print(file.read(5)) reads first 5 bytes and then displays the values on the screen. The statement print(file.read(10)) reads the next 10 bytes and displays it.

If you use the read() method without mentioning any bytes in it, it will read the entire content of the file. Let us see an example.

**Example 12**

| Python code                 | Output                                  |
|-----------------------------|---|
| file = open("text.txt","r") | I am learning Python.                   |
| print(file.read())          | This is my first file handling program. |
| #reads till the end of file | I am enjoying it.                       |

**13.6 BINARY FILES**

Till now you have seen the various methods of file object along with the examples of reading and writing the data in the form of strings to the text files by using Python. Although you can use the text files to write complex data structures, but the text files store these structures as simple text. For reading the data, you may need to analyze the text and process it back to its original data structure.

To store Python data structure in its original form, binary files are used. In binary files, complex data structure is written in their original form, thus at the time of reading, you get the original data back.



## Notes

To store and retrieve the data in binary files, '**Pickle**' module is used. The pickle module is used to store any kind of object in a file as it allows us to store Python objects with their structure. The pickle module supports two major operations: '**Pickling**' and '**Unpickling**'. Pickling is the process whereby a Python object is converted into a byte stream and unpickling is the process by which a byte stream is converted back into the desired object. The process of pickling and unpickling a Python data object is known as '**object serialization**'.

To use the pickle module, you first need to import it. It provides two important methods; viz, dump and load.

Let us learn these methods one by one.

**dump method**

This method is used to store the object data to the binary file.

The syntax for using dump() method is

**Syntax****dump(object, file\_object)**

where object: is the Python object that needs to be written to a binary file,  
file\_object: is the name of the binary file object on which the write operation has to be performed.

Let us see an example using dump() method.

**Example 13**

| Python code   | Explanation   |
|---|---|
| <pre>import pickle list=[5,10,15,20,25] file=open("file1.dat","wb") pickle.dump(list,file) file.close()</pre> | <p>This Python code first creates a list and then writes this list to a binary file '<b>file1.dat</b>' using dump method.</p> |

Here the value of the '**list**' variable will be written to the file '**file1.dat**' using dump() method.

Consider another example in which the created dictionary has been written in to a binary file.



Notes

**Example 14**

| Python code  | Explanation   |
|--|---|
| <pre>import pickle dict={'Rohit':450, 'Anvi':310,'Mudit':315,'Ravi':425} file=open("file1.dat","wb") pickle.dump(dict,file) file.close()</pre> | <p>This Python code first creates a dictionary and then writes this dictionary to a binary file <b>'file1.dat'</b> using dump method.</p> |

**Load method**

This method is used to retrieve the data from the pickled file.

The syntax for using load method is:

**Syntax**

`object = load(file_object)`

where **object:** is the Python object, which is being read from the binary file and **file\_object:** is the name of the binary file object from which the read operation needs to be performed.

Let us see an example using load method.

**Example 15**

| Python code   | Output  |
|---|---|
| <pre>import pickle file=open("file1.dat","rb") dict=pickle.load(file) print (dict) file.close()</pre> | <pre>{'Rohit': 450, 'Anvi': 310, 'Mudit': 315, 'Ravi': 425}</pre> |

Here the file **'file1.dat'** is opened in reading mode. Then the load method is used to retrieve the content from the pickled file **'file1.dat'**. Using the `print( )` statement, the content of the file is displayed.





## INTEXT QUESTIONS 13.2

1. State whether the following are True or False.
  - (a) The `readline()` method reads the file till the new line and returns that line.
  - (b) The `writelines()` method is used to write a list of strings in a file.
  - (c) The `file.read(5)` statement will always read 5 bytes from the beginning of the file.
  - (d) The `file.readlines()` method reads the first line from the file.
  - (e) An advantage of using **with** method is that any files opened will be closed automatically after you are done.



## WHAT YOU HAVE LEARNT

- A **file** is a named location on a secondary storage device, which stores the data permanently.
- In Python, the files are divided into two broad categories: **text files** and **binary files**.
- There are two ways of opening a file in Python: **open() function** and **with open() statement**.
- The file modes describe the type of operation you are going to perform on that particular file such as read, write, append etc.
- Closing a file frees up the resources that were tied with the file.
- The **write() and writelines()** methods are used to write the data in a file.
- The **write()** method takes a string as a parameter and writes it in the file and **writelines()** method writes a list/sequence of strings to a file.
- The **readline(), readlines(), read(n)** methods are used to read data from a file.
- The **readline()** method reads one line(i.e. till newline) at a time from a file and returns that line.
- The **readlines()** method reads the entire content of the file in one go and returns it as **list of lines**.



Notes

## MODULE-2

Computer Applications  
(712)



Notes

### File Handling in Python

- The **read(n)** method reads a string of size (here n) from the specified file and returns it.
- To store and retrieve the data in binary files, **Pickle** module is used.

The pickle module supports two major operations: **Pickling** and **Unpickling**. Pickling is the process whereby a Python object is converted into a byte stream and Unpickling is the process by which a byte stream is converted back into the desired object.

- The process of pickling and unpickling a Python data object is known as '**object serialization**'.
- The pickle module provides two important methods; **dump** and **load**.



#### TERMINAL EXERCISE

1. What is the difference between text and binary files?
2. How is write() method different from writelines() method?
3. Explain the difference between write and append mode of opening a file.
4. Write a Python script to read a text file "Quotes.txt" and display only those lines that start with "E".
5. Create a dictionary having names of 5 states and their population. Store it in a binary file.
6. Read a binary file which contains list of multiples of 5 and display the maximum value stored in it.
7. Read a binary file containing a list of numbers and count the numbers stored in it.
8. Write a python program to copy the contents of one file to another.
9. Write a Python script to create a list containing names of 7 monuments and write them to a file, "monuments.txt"
10. Write a Python script to create a string containing the following statements:  
The roots of education are bitter, but the fruit is sweet.  
Education is the most powerful weapon, which you can use to change the world.  
Strive for Progress, Not perfection.  
Write these strings in a file, "quotes.txt".



**ANSWERS TO INTEXT QUESTIONS**

**13.1**

1. (a) True    (b) False    (c) False    (d) True    (e) True
2. (a) `file= open ("abc.txt","r")`  
(b) `f= open ("stu.dat","rb+")`  
(c) `file =open ("item.dat", "ab")`  
(d) `with open("game.txt","r") as file:`  
(e) `fileobj.close()`

**13.2**

1. (a) True    (b) True    (c) False    (d) False    (e) True

**Key Learning Outcomes**

- Be proficient in handling files in Python and undertake operations such as reading, writing, appending etc., in text files.
- Be able to store the data and retrieve the stored data from the binary file.



**Notes**



## ERRORS AND EXCEPTIONS

---

### 14.1 INTRODUCTION

---

By now, you must have been experienced in writing programs in Python language. Sometimes you might get error messages, such as invalid syntax, variable not defined etc. In case of any error in the Python program, the program terminates as soon as it encounters that error. In Python programming, an error can be a syntax, logical error or exception. The errors that occur at program's runtime (during execution of the program) are called '**Exceptions**'. It is always a good practice to add safeguards to your programs so that the common errors and exceptions can be handled properly. In this lesson, you will learn different types of exceptions and methods to handle them.



### OBJECTIVES

After reading this lesson, you will be able to:

- define exception;
- list the types of error that occur in Python programming;
- distinguish between error and exception and
- handle the errors, which occur during runtime by using try... except... else... finally.

---

### 14.2 ERRORS AND EXCEPTIONS

---

An exception is an error that occurs during the execution of a program. You might have seen trace backs (it is a report containing the function calls made in your



program at a specific point) while running the script. A traceback gives you an idea of what went wrong in your program, including where the error occurred and what type of error it was. Python uses these errors to indicate that something exceptional has occurred, and your program cannot continue unless it is handled.

Errors are broadly classified into two types: Syntax errors and Logical errors. Syntax error occurs due to violation in the rules of writing syntax and causes a program to crash or not run at all.

Some of the standard most frequent built-in exceptions are: Zero Division Error, Index Error, TypeError, NameError, SyntaxError, ValueError etc.

Let us discuss them one by one.

### (a) Zero Division Error

Did you ever try to divide any number by zero in Python? If you try this, then you will get the ‘Zero Division Error’. This error will be raised when division or modulo by zero takes place.

Let us consider an example.

#### Example 1

```
#finding inverse of a number
num=int(input("enter a number"))
rev_num=1/num #raises exception when num is 0
print("Inverse of a number=", rev_num)
```

#### Output

```
enter a number0
```

```
Traceback (most recent call last):
```

```
File"C:/Python/Python36/e1.py", line 3, in <module>
```

```
rev_num=1/num
```

#### Zero Division Error: division by zero

In the above example, if you run this program, it will ask for the input of a number. If the user enters **0**, the variable ‘**num**’ is assigned to the value 0. The program, statement, `rev_num=1/num` i.e., `rev_num=1/0`; will raise an exception ‘**Zero Division Error: division by zero**’. This error occurred while running the program. This error can be avoided by requesting the user not to give the input of a number as 0 or by checking the input in the program whether it is 0 or not.

**(b) Index Error**

This error will be raised when an index is not found in a sequence. In the following example, list1 is having 3 elements. You have learned that the index value of a list begins at 0. i.e., the index values of the three elements are 0,1, and 2. If you try to print the index value of 3 from the list list1, it will raise an index error.

Let us take an example, for raising an **IndexError** exception by accessing the element, which is not available in the list.

**Example 2**

```
#displaying elements of a list
```

```
list1=[10,20,30]
```

```
print("First element=", list1[0])
```

```
print("Third element=", list1[3]) #index value 3 is not present in the list1
```

**Output**

```
First element= 10
```

```
Traceback (most recent call last):
```

```
File "C:/Python/Python36/e2.py", line 4, in <module>
```

```
print("Third element=", list1[3])
```

**Index Error: list index out of range**

Here, during the runtime of the program, first it displays the list1[0] element. When it encounters the statement for displaying the element of list1[3], then the program will throw an '**Index Error: list index out of range**'. Now by reading this error message, you can change the statement list1[3] with list1[1] or list1[2] in the program.

**(c) Name Error**

This error will be raised by the program when an identifier is not found in the local or global namespace. It means the variable name is not declared/defined in that particular program.

Let us understand this error with the help of an example.

**Example 3**

```
#adding two numbers
num1=10
num2=20
print("Sum of two numbers=", num1+num3) #num3 is not a variable name
```

**Output**

Traceback (most recent call last):

File "C:/Python/Python36/e3.py", line 4, in <module>

```
print("Sum of two numbers=", num1+num3)
```

**NameError: name 'num3' is not defined**

In the above example, '**num3**' variable is not declared/defined. When the program executes the statement `num1+num3`, it will raise an error '**NameError: name num3 is not defined**'. From this error, you can understand that you have used the variable '**num3**' in your program, which is not defined. You can solve this error by changing the variable name as '**num2**' instead of '**num3**'.

**(d) Syntax Error**

This error will be raised when there is an error in the syntax of the statements in the program.

**Example 4**

```
#create a list with numbers 10, 20, 30
>>>list=<10,20,30> #list elements are enclosed in square brackets
```

**Output**

SyntaxError: invalid syntax

In the above example, the list elements should be enclosed in square brackets i.e., `list=[10,20,30]`. Since the statement `list=<10,20,30>` is not as per the syntax, '**syntax error: invalid syntax**' will be raised. This error can be avoided by following the right syntax, in this case, `list=[10,20,30]`

**(e) Type Error**

This error will be raised when an operation or function is attempted that is invalid for the specified data type.



Let us understand this error with the help of an example.

### Example 5

```
#multiplying two numbers
x="4"
y="2"
x*y
```

### Output

Traceback (most recent call last):

File "<pyshell#10>", line 1, in <module>

```
x*y
```

**TypeError: can't multiply sequence by non-int of type 'str'**

Here, x, y are the two identifiers holding values of string data type. In the assignment statements of the above example, it should have the statement as **x=4** and **y=2** instead of **x="4"** and **y="2"**.

### (f) Value error

This error will be raised when a built-in operation or function receives an argument that has the right type but an inappropriate value.

### Example 6

```
>>>num=int(input("enter number"))
```

### Output

```
enter number@12 #@12 can not be converted to integer
```

Traceback (most recent call last):

File "<pyshell#7>", line 1, in <module>

```
num=int(input("enter number"))
```

**ValueError: invalid literal for int() with base 10: '@12'**

Here the user entered the value '@12', which cannot be converted into an integer. This throws a ValueError: invalid literal for int() with base 10. This error can be avoided by giving an integer value to the program such as 12.



Some of the other important built-in exceptions are listed in Table 14.1.

**Table 14.1: List of Built-in Exceptions in Python**

| Exception         | Description   |
|-------------------|---|
| EOFError          | This error will be raised when the input() function hits the end-of-file condition. |
| ImportError       | This error will be raised when the imported module is not found.                    |
| IndentationError  | This error will be raised when there is an incorrect indentation in the code.       |
| KeyError          | This error will be raised when a key is not found in a dictionary.                  |
| KeyboardInterrupt | This error will be raised when the user hits the interrupt key (Ctrl+c or delete).  |
| MemoryError       | This error will be raised when an operation runs out of memory.                     |
| RuntimeError      | This error will be raised when an error does not fall under any other category.     |



**Notes**



**INTEXT QUESTIONS 14.1**

- Name the type of error that will be produced when executing the following statements:
  - `num1="num"+10`
  - ```
var1="RAM"
var2="SHYAM"
print(var1+var3)
```
  - `num2=int("#20")`
  - ```
num=10
ifnum>10
printnum
```



- (e) `var1=10`  
`var2=0`  
`var3=var1/var2`
- (f) `list1=['A','B','C','D']`  
`print list1[4]`

2. Choose the correct option

- (a) `num=['A','B']`  
`print(num[2])` statement will throw which of the following error.
  - (i) `NameError`
  - (ii) `ValueError`
  - (iii) `IndexError`
  - (iv) `TypeError`
- (b) `int('23.78')` statement will throw which of the following error.
  - (i) `NameError`
  - (ii) `ValueError`
  - (iii) `IndexError`
  - (iv) `TypeError`
- (c) Which of the following is not a Python exception?
  - (i) `NameError`
  - (ii) `ValueError`
  - (iii) `TitleError`
  - (iv) `TypeError`

---

### 14.3 RAISING AN EXCEPTION

---

Python provides a `raise` statement that allows a program to manually throw (raise) an exception in Python. This is usually done for the purpose of error-checking. The general syntax for the `raise` statement is as follows.

#### **Syntax**

```
raise [Exception [, argument [, traceback]]]
```



Where,

**Exception** is the type of exception (for example, ValueError),

**Argument** is a short message that describes an error. You can give your own message to notify the error. The argument is optional; if not supplied, the exception argument is None and

**Traceback** is also optional and if it is present, then the traceback object is used for the exception.

Let us understand with the help of an example.

### Example 7

```
num=int(input("Enter a number"))
if num%2 ==0:
raise Exception('number should not be divisible by 2')
```

### Output

Enter a number12

Traceback (most recent call last):

File "C:/Users/Vineeta/AppData/Local/Programs/Python/Python36/error2.py",  
line 3, in <module> raise Exception('number should not be divisible by 2')

Exception: number should not be divisible by 2

When an even number is given as an input to the program, it will display an exception on the screen, offering clues about what went wrong in the program. Now you can understand that you should not give an even number to this program as an input.

## 14.4 EXCEPTION HANDLING

In the previous section, you have learnt about the different types of errors and exceptions that may occur while executing the programs. If you know that a particular section of the program is likely to cause an error, you can tell Python what to do if it happens. Instead of letting the error crash your program, you can intercept it, do something about it and allow the program to continue.

Error handling will be done in Python as follows:

- (i) The moment when an error occurs the state of execution of the program will be saved.



- (ii) Normal flow of the program execution will be interrupted (stopped for a moment).
- (iii) A special function or piece of code known as '**exception handler**', will be executed.
- (iv) The execution of the program is resumed with the previously saved data.

This process of error handling is called '**exception handling**'.

We have already discussed few built-in exceptions. The programmer can also define his/her own exceptions. The exception handling will be done in Python with the help of **try** statement.

The syntax for using try statement is:

### Syntax

```
“try” “:” suite
“except” ExceptionName “:” suite
[“except” ExceptionName “:” suite]
.
.
.
[“except” “:” suite]
[“else” “:” suite]
[“finally” “:” suite]
```

Here suite means a group of individual statements, which make a single block of code. The **try** statement may have multiple **except** clauses with '**Exception Names**' to handle the specific exception, and one optional **except** clause. The **else** and **finally** clauses are optional.

Let us take few examples to understand the usage of **try** statement.

### **try....except clause**

When you use **try...except** clause in the script, and if an error occurred, then **try** block code execution will be stopped and the program control will be transferred to the **except** block.

Let us rewrite the Example3 using **try...except** clause.



Notes

**Example 8**

```
#adding two numbers

try:
    n1=100
    n2=200

    print("Sum of two numbers=", n1+n3)
except:
    print("An error has occurred")
```

**Output**

An error has occurred.

If you compare the output of Example 3 and Example 8 you can find that Example 8 does not end abruptly, but displays the specified error message. As there is an error in the print statement (variable num3 is not declared), a try block code execution will be stopped and the program control will be transferred to the except clause. The print statement present in the except clause will be executed.

Let us take another example.

**Example 9**

```
#dividing two numbers

n1=10

while n1!=100:

    try:
        n1=int(input("Enter dividend"))
        n2=int(input("Enter divisor"))
        n3=n1//n2

        print("The quotient is=",n3)
    except:
        print("An error has occurred")
```



### Notes

### Output

Enter dividend12

Enter divisor4

The quotient is= 3

Enter dividend34

Enter divisor0

An error has occurred

Enter dividend65

Enter divisor2

The quotient is= 32

Enter dividend100

Enter divisor23

The quotient is= 4

In this example, the while loop has been used. If the value of the variable '**n1**' (dividend) is not 100, then it will get the value of the variable '**n2**' (divisor). It will then perform the division operation and displays the quotient after the division. If the user enters value 100 to the variable '**n1**', then it will find out the quotient by getting value for another variable '**n2**' also. After this, the while loop will stop executing further (because we have used the condition while n1!=100). If you observe the output, when we give '**n1**' as **12**, and '**n2**' as **4** it finds out the quotient as 3. When we give '**n1**' as **34** and '**n2**' as **0**, then the program displays the statement in except clause and the program does not end abruptly when an error occurs. If you do not use the exceptions in the program, it will terminate when you enter divisor i.e., '**n2**' as 0, because division by 0 is not defined.

In the above code, the error message does not say about the type of error occurred during runtime. To know more about the errors, you can use multiple **except** statements.

### **try..except with multiple except statements**

You can give a name of the exception in the except statement. In this case, when an error occurs, all except clauses will be searched until it finds the one that matches the exception. If no except clause matches the criteria, then the control will be transferred to the except clause without any exception name.

**Notes****Example 10**

```
#dividing two numbers
n1=10
while n1!=100:
try:
    n1=int(input("Enter dividend"))
    n2=int(input("Enter divisor"))
    n3=n1//n2
    print("The quotient is=",n3)
except ZeroDivisionError: print("Division by zero not defined")
except: print("An error has occurred")
```

**Output**

Enter dividend25

Enter divisor5

The quotient is= 5

**Enter dividend86****Enter divisor0****Division by zero not defined**

Enter dividend78

Enter divisor5

The quotient is= 15

**Enter dividend34****Enter divisor0****Division by zero not defined**

**Enter dividend34****Enter divisor2****The quotient is = 17**

Enter dividend100

Enter divisor6

The quotient is = 16

In this program, two except clauses are used. When a user enters 0 as divisor, the error '**Division by zero not defined**' will be displayed. If any other error occurs (except ZeroDivisionError) during the execution time, then "An error has occurred" message will be displayed.

**try..except with else clause**

You can also use else clause in try statement. The else clause is used to specify the statements in the code, which will be executed if there is no exception occurs during the execution of program.

Let us consider an example for try..except with else clause.

**Example 11**

#dividing two numbers

n1=10

while n1!=100:

try:

n1=int(input("Enter dividend"))

n2=int(input("Enter divisor"))

n3=n1//n2

print("The quotient is=",n3)

except ZeroDivisionError: print("Division by zero not defined")

except SyntaxError: print("Syntax error")

except TypeError: print("Data types do not match")

except: print("An error has occurred")

else: print("Execution is normal")



**Output**

Enter dividend12  
 Enter divisor3  
 The quotient is= 4  
 Execution is normal

Enter dividend12  
 Enter divisor0  
 Division by zero not defined

Enter dividend100  
 Enter divisor5  
 The quotient is= 20  
 Execution is normal

In the above example, you can observe that the **else** clause will be executed only when none of the exceptions is executed. Whenever an error occurs during the execution time, the required exception will be executed and else clause will not be executed.

**try..except with finally clause**

The finally clause is called '**clean-up**' or '**termination clause**', because it is executed under all circumstances, i.e., this clause is always executed irrespective of whether an exception has occurred in a try block or not.

Let us understand the use of finally clause with an example.

**Example 12**

```
#dividing two numbers
n1=0
while n1!=100:
    try:
        n1=int(input("Enter dividend"))
        n2=int(input("Enter divisor"))
```

## MODULE-2

Computer Applications  
(712)



Notes

```
n3=n1//n2

print("The quotient is=",n3)

except ZeroDivisionError: print("Division by zero not defined")

except: print("An error has occurred")

else: print("Execution is normal")

finally: print("done!!!!")
```

### Output

```
Enter dividend25
Enter divisor5
The quotient is= 5
Execution is normal
done!!!!

Enter dividend15
Enter divisor0
Division by zero not defined
done!!!!

Enter dividend100
Enter divisor4
The quotient is= 25
Execution is normal
done!!!!
```

In this program, the print statement given in the finally clause is always executed irrespective of whether exception occurs or not. So **'done!!!!'** will be displayed, after every iteration of while loop. The print statement given in the else clause will be executed only when there is no error occurred during the execution.



## INTEXT QUESTIONS 14.2

- How many except statements can a try-except block have?
  - zero
  - one
  - at least one
  - more than one
- When will the else part of try-except-else be executed?
  - it will always be executed
  - when an exception occurs
  - when no exception occurs
  - It will not be executed at all
- Is the following code valid?
 

```
try:
    # Python statements
except:
    # Python statements
else:
    # Python statements
```

  - no, else cannot be used with try.
  - no, else cannot be used with except.
  - no, else must come before except.
  - yes, the code is valid.
- When will the finally block executed?
  - if there is no exception.
  - if there is an exception.
  - only when certain specified condition is satisfied.
  - it is always executed.
- What is the output of the following code?
 

```
try:
    print("Start" , end=" ")
finally:
    print("End")
```

  - Start
  - End
  - Start End
  - no output



Notes

## MODULE-2

Computer Applications  
(712)



Notes



### WHAT YOU HAVE LEARNT

- In Python programming an error can be a syntax error, logical error or an exception. The syntax errors are the errors that occur when you do not follow the prescribed format for the statements, functions, loops etc., in the program. The errors that occur at runtime are called '**Exceptions**'.
- Some of the built-in exceptions are: ZeroDivisionError, IndexError, TypeError, NameError, SyntaxError, ValueError, ImportError, EOFError, IndentationError, KeyError, MemoryError etc.
- Python provides a raise statement that allows a program to force a specified exception to occur at runtime.
- When you use try...except clause in the script and if an error occurs, then a try block code execution will be stopped, and the program control will be transferred to the except block.
- You can use try..except with multiple except clauses. In this case, when an error occurred, all except clauses will be searched until it finds the one that matches the exception.
- In the try statement, the else clause can also be used. The else clause specifies the statements, which will be executed, if there is no exception occurred in the program.
- The finally clause is called '**clean-up**' or '**termination clause**', because it is executed under all circumstances.



### TERMINAL EXERCISE

1. What are the errors and exceptions in Python?
2. Explain the following exceptions with an example:  
ZeroDivisionError, IndexError, TypeError, NameError, SyntaxError, ValueError
3. What do you understand by raising an exception? Explain with example.
4. How is exception handling done in Python?



5. Give the output for the following code:

```
d=int(input("Enter Day"))
m=int(input("Enter Month"))
if d>=1 and d<=31:
    if m>=1 and m<=12:
        print("valid Day and Month")
    else:
        raise ValueError("Invalid Month")
else:
    raise ValueError("Invalid Day")
print(d,m)
```

if input is:

- (i) 2, 10                      (ii) 10, 200                      (iii) 90, 75



**ANSWERS TO TEXT QUESTIONS**

**14.1**

1. (a) TypeError                      (b) NameError                      (c) ValueError  
 (d) SyntaxError                      (e) ZeroDivisionError                      (f) IndexError
2. (a) (iii) IndexError                      (b) (ii) ValueError                      (c) (iii) TitleError

**14.2**

1. (d) more than one                      2. (c) when no exception occurs  
 3. (d) yes, the code is valid                      4. (d) it is always executed  
 5. (c) Start End

**Key Learning Outcomes**

- Be able to identify the type of error that may occur in Python programming during runtime.
- Be proficient in evaluating and handling the exception in the program.

## MODULE-2

Computer Applications  
(712)



Notes

# 15

## DATABASE MANAGEMENT SYSTEM

---

### 15.1 INTRODUCTION

---

Computers are used in every sphere of human life. In our everyday life, computers are put to use for all sorts of applications ranging from complex calculation in the field of scientific research to teaching, entertainment etc. The goal of computerisation is to get the required result in a short span of time after processing the data. Existence of every application is based on data and the computer is used essentially as a Data Processor Machine. In this lesson, we will learn about how data is stored in a given format and how it can be retrieved and accessed in an efficient manner as per the requirement of the user.



### OBJECTIVES

After reading this lesson, you will be able to:

- explain about database and DBMS (Database Management System);
- create RDBMS (Relational Database Management System);
- create tables;
- use constraints.

---

### 15.2 DATA AND INFORMATION

---

Data in computer terminology means raw facts and figures, which itself has no meaning. For example, the data can be your name, age, date of birth etc. Data after processing is known as information.



**Fig. 15.1: The Relation Between Data and Information**

For example, information can be a weekly schedule of classes as mentioned at Table 15.1. If a computer coaching institute is providing different courses on different days, the courses such as MS Office, system analysis etc., and days such as Mon, Tue., will be termed as data items. Consider Table 15.1, which shows the weekly schedule of classes of computer coaching institute. If you represent the data in a meaningful manner, it will be considered as information.

**Table 15.1: Schedule of Classes**

| Days | Topic            |                       |                       |
|------|------------------|-----------------------|-----------------------|
|      | I Period         | II Period             | III Period            |
| Mon  | Operating System | MS-Office             | MS-Office             |
| Tue  | MS-Office        | MS-Office             | System analysis       |
| Wed  | Operating System | Programming in Python | Computer Fundamentals |
| Thu  | Hardware Concept | Hardware Concept      | SystemAnalysis        |
| Fri  | Operating System | Programming in Python | MS-Office             |

Now, let us discuss about a database.

### 15.3 DATABASE

A database is a computer-based record system, which can be defined as a collection of logically related data items that are stored together at a centralized location to serve multiple applications. For example, the details of all trains running to a particular destination can be considered as a database. Here each train details such as train number, name, source, destination, schedule arrival time, schedule departure time, day of travel, number of seats etc., will be stored in tables. The collection of tables is called as database.

#### 15.3.1 Need for Databases

Imagine a ticket booking system in a railway reservation centre, where there are four counters all having separate copies of the same data. As a result, copies of the



Notes



### Notes

same ticket can be issued through different counters to different people. Hence, it may create seating problems. The problem can be resolved if the data is stored at one place and these counters will refer to the centralized data. So, when a ticket is issued from one counter, then it will automatically get updated at the centralized location and the same ticket cannot now be booked from the other counters. The Indian railways are maintaining the train data in the centralized database. A centralised database is a database that is located, stored and maintained at a single location. Hence, whoever tries to search the train details (e.g., from New Delhi to Nagpur or Agra to Hyderabad) the result will be same to all because of the centralised database.

### Advantages of using Database

**Reduces Data Redundancy:** The database management system does not maintain separate copies of the same data, rather all the data are kept at one place. The applications that require the same data refer to the centrally maintained database. If any change has to be made, it will be made at just one place and the same will be reflected in other applications too. This way duplication of data will be reduced. The duplication of data is known as **data redundancy**.

**Controlled Data Inconsistency:** When data redundancy is not controlled, there may be occasions on which the two entries of the same data do not match i.e., when one of them stores the updated version and the other does not. In that case the database is said to be inconsistent and needs controlling the data redundancy.

**Sharing of Data:** In a database system, all the applications refer to the same data stored at a central location. If data is to be shared among several other users, the system will avoid possible anomalous results.

**Enforces Standard:** The database management system ensures that, all the data follows the applicable standard laid down by the organisation.

**Integrity Can Be Maintained:** When a database contains the data to be used by many users it is important to keep integrity checks to ensure that the data values satisfy certain specified rules for example minimum balance in a bank account should be Rs.2000.

**Ensures Security:** A database ensures security and privacy by restricting only the authorized persons to access the database.

---

### 15.4 DATABASE MANAGEMENT SYSTEM (DBMS)

---

DBMS is a set of programs which create and manage a database. It provides the facilities to:





- define structure of data storage i.e., creation of database,
- organise the database i.e., arrange the databases in a specified order,
- provide mechanisms for data manipulation, i.e., to retrieve specified data according to a condition,
- update database i.e., add new records, edit/delete data items etc., and
- provide data security against unauthorised access.

### Relational Database Management System (RDBMS)

Dr. E.F. Codd described the relational model in 1970. In Relational database, the data is organised in the form of a relation wherein a relation is nothing but a table which is a collection of rows and columns. A row in a table represents a relationship among a set of values. Since a table is a collection of such relationships, it is generally referred to as a **Relation** or **Relational**.

For example, (Table 15.2), a sample database is represented in a relational form:

**Table 15.2: Employee Table and Department Table**

| Relation : Employee |         |         |        | Relation: Department |           |
|---------------------|---------|---------|--------|----------------------|-----------|
| Emp_no              | E_Name  | Dept_no | Salary | Dept_no              | Dept_name |
| 1                   | Samayra | 10      | 6400   | 10                   | Accounts  |
| 2                   | Shirom  | 20      | 6000   | 20                   | Admin     |
| 3                   | Kiaan   | 20      | 5200   |                      |           |

Here, the Employee table has four columns (Emp\_no, E\_Name, Dept\_no and Salary; Emp\_no refers to Employee number, E\_Name refers to the Name of the employee, Dept\_no refers to the department number, Salary refers to the Salary of the employee). In the same way, Department table contains two columns (Dept\_no, Dept\_name; Dept\_no refers to the department number, Dept\_name refers to the department name).

A Relational Database Management System can be defined as a Database Management System where all logically related data items are organised strictly as tables and all database operations work on these tables.

Let us discuss some of the terms used in the relational model.

#### Terms Used in the Relational Model

**Relation:** The data in the table is arranged in rows and columns. Each table in a database has a unique table name that identifies its contents. A relation has the following properties:



**Notes**

- For a row, each column must have an atomic value i.e., only one value.
- All rows are distinct i.e., no two rows are identical.
- The ordering of rows is immaterial.
- For any given column of a table, all values are of the same kind.

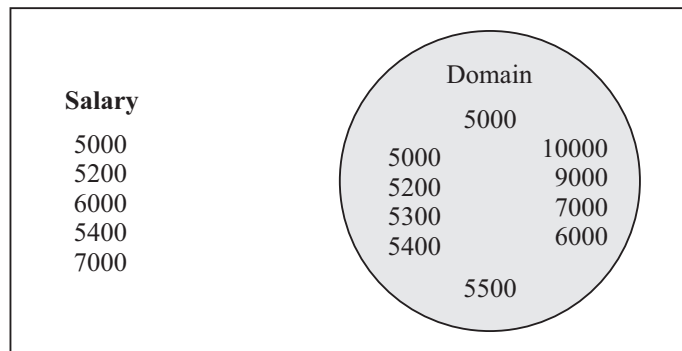
**Tuple:** A row or a record of a table is referred to as a tuple.

**Attribute:** The columns or fields of a table are called as attributes. Every column (attribute) in a table must have a unique name. In the above example (Emp\_no, E\_Name, Dept\_no and Salary all are attributes. A column must have values of same type.

**Degree:** The total number of attributes in a relation is called the degree of a relation. A relation having three attributes is said to be of Degree 3. The degree of Employee table (refer table 15.2), is 4.

**Cardinality:** The total number of tuples in a relation is called cardinality of that relation. The cardinality of the Employee is 3 (refer table 15.2).

**Domain:** A domain is a set of values from which one or more attributes (column) draw their actual values. For example, for the attribute salary, the values can be from a range of Rs.5000 to Rs.10000 (refer Fig. 15.2).



**Fig. 15.2: Set of Values for Salary Attribute - Domain**

**Keys:** These refer to the attributes, which specify how rows of a relation are distinct from each other. Let us discuss some of the keys.

- (i) **Primary Key:** It is a set of one or more attributes that can uniquely identify a row within a relation. For example, in the Employee table, Emp\_no contains unique values and cannot contain NULL (Not defined) values. The primary key is non-redundant i.e., it does not allow duplicate values for any two rows in the same relation.



Notes

- (ii) **Candidate Key:** All the attributes in a relation capable of becoming a primary key are candidate keys. For example, in Table 15.3a, the admission no. and roll no. (Roll\_no and Adm\_no) attributes of the relation Student can serve as the primary key. So, both these attributes are called as candidate keys.
- (iii) **Alternate Key:** A candidate key which is not the primary key is called as an alternate key. For example, in Student relation (refer Table 15.3a), if Roll\_no is chosen as primary key, the Adm\_no becomes the alternate key.
- (iv) **Foreign Key:** A foreign key is used to represent the relationship between tables. In other words, a non-key attribute whose values are derived from the primary key of some other table is known as foreign key. For example, in the Department table, (defined at Table 15.2) Dept\_no will be the primary key whereas, Dept\_no will be the foreign key in the table Employee (Table 15.2).
- (v) **Composite Key:** In certain tables, a single attribute cannot be used to identify rows uniquely and combination of two or more attributes is used as primary key. Such keys are called composite keys. For example, refer the Order table (Table 15.4b) where the combination of the supplier no (S\_no) and order no (O\_no) can serve as the primary key.

## 15.5 THE RELATIONAL ALGEBRA

The relational algebra is a collection of operations on relations or tables. Each operation takes one or more relations and produces another relation as its result. Some of the operations include selection, projection, union, Cartesian product etc.

### (i) Selection

The selection operation creates a horizontal subset of a given relation. It selects tuples that satisfy a given condition. For example, consider relation – Student (as shown at Table 15.3a, where Roll\_no refers to roll number of the student, Adm\_no refers to the admission number of the student, S\_name refers to student's name, stream refers to the student's stream such as science, commerce and humanities).

**Table 15.3a: Student Relation**

| Roll_no | Adm_no | S_name  | Stream     |
|---------|--------|---------|------------|
| 1       | 101    | Samayra | Science    |
| 2       | 102    | Shirom  | Science    |
| 3       | 103    | Kiaan   | Commerce   |
| 4       | 104    | Tarun   | Humanities |



Notes

For a condition *stream = Science*, the result could be as shown in Table 15.3b.

**Table 15.3b: Student Relation of Science Stream**

| Roll_no | Adm_no | S_name  | Stream  |
|---------|--------|---------|---------|
| 1       | 101    | Samayra | Science |
| 2       | 102    | Shirom  | Science |

**(ii) Projection**

The projection operation creates a vertical subset of a given relation i.e., it selects specified attributes or column from a given relation.

For example, for the above relation student selecting *Roll\_no* and *S\_name* will display the result as shown in Table 15.3c.

**Table 15.3c: Project Operation on Student**

| Roll_no | S_name  |
|---------|---------|
| 1       | Samayra |
| 2       | Shirom  |
| 3       | Kiaan   |
| 4       | Tarun   |

**(iii) Join**

It involves retrieval of information from more than one table having a common field. When extraction of information is from more than one table, the concept of JOIN will be used.

For example, consider the relation; Supplier and Order (as shown in Table 15.4a and b).

**Table 15.4a: Supplier**

| S_No | S_name  | Phone      |
|------|---------|------------|
| 1    | Rajeev  | 9777777777 |
| 2    | OM      | 9888888888 |
| 3    | Pradeep | 9666666666 |
| 4    | Nitesh  | 9555555555 |

The attribute S\_No refers to the supplier number, S\_name refers to the supplier name, Phone refers to supplier's phone number.

**Table 15.4b: Order**

| O_No | S_No | O_Date     | Amount |
|------|------|------------|--------|
| 101  | 1    | 25-11-2018 | 4000   |
| 102  | 2    | 10-09-2018 | 5000   |
| 103  | 2    | 25-08-2018 | 6000   |
| 104  | 3    | 03-03-2018 | 4500   |

The attribute O\_No refers to the order number, S\_No refers to the supplier number, O\_date refers to the order date and Amount refers to the amount of the order.

To display the details of all suppliers along with the amount who have placed the order,

JOIN operation needs to be performed which will display the result as shown in Table 15.4c.

**Table 15.4c: Supplier and Order Join Relation**

| S_No | S_name  | Phone      | Amount |
|------|---------|------------|--------|
| 1    | Rajeev  | 9777777777 | 4000   |
| 2    | OM      | 9888888888 | 5000   |
| 3    | Pradeep | 9666666666 | 6000   |

#### (iv) Union

Union operation is a binary operation that requires two tables/relations as its operand. Two relations are Union compatible when:

- Both relations are of same degree i.e., both tables must have the same number of columns.
- The domains of corresponding attributes of both relations must be the same i.e., the first attribute of the first table must be the same as the first attribute of the second relation.

For example, in the above given relation student (refer Table 15.3a) and the relation New\_admission (refer Table 15.5) which is given below are Union compatible.



Notes



**Notes**

**Table 15.5: New\_Admission**

| Roll_no | Adm_no | S_name  | Stream     |
|---------|--------|---------|------------|
| 5       | 106    | Krishna | Science    |
| 6       | 107    | OM      | Humanities |

After union operation the relation Student union New\_Admission will give the following results (refer table 15.6):

**Table 15.6: Student Union New\_Admission**

| Roll_no | Adm_no | S_name  | Stream     |
|---------|--------|---------|------------|
| 1       | 101    | Samayra | Science    |
| 2       | 102    | Shirom  | Science    |
| 3       | 103    | Kiaan   | Commerce   |
| 4       | 104    | Tarun   | Humanities |
| 5       | 106    | Krishna | Science    |
| 6       | 107    | OM      | Humanities |

**(v) Cartesian Product**

Cartesian Product is a binary operation and is denoted by letter “X”.

The Cartesian Product of two relations A and B results a new relation (A X B) which has degree (number of columns) equal to sum of degrees of the two relations operated upon and cardinality (number of rows) will be equal to(=) number of rows of relation A multiplied by number of rows of relation B.

The Cartesian Product of the two relations results in a relation with all possible combinations of the tuples of two relations which are involved in the Cartesian product. All tuples of first relation are combined with all tuples of the second relation to form the resultant relation.

For example- consider the following relations; student and teacher (refer Tables 15.7a and 15.7b).

**Table 15.7a: Student**

| Roll_no | Adm_no | S_name  | Stream  |
|---------|--------|---------|---------|
| 1       | 101    | Samayra | Science |
| 2       | 102    | Shirom  | Science |



Notes

Table 15.7b: Teacher

| T_Id | Tname | Subject    |
|------|-------|------------|
| T01  | Karan | Economics  |
| T02  | Aditi | Statistics |

The Cartesian Product of **Student** and **Teacher** “(Student X Teacher)” will be:

Table 15.7c: Cartesian Product of Student and Teacher

| T_Id | Tname | Subject    | Roll_no | Adm_no | S_name  | Stream  |
|------|-------|------------|---------|--------|---------|---------|
| T01  | Karan | Economics  | 1       | 101    | Samayra | Science |
| T01  | Karan | Economics  | 2       | 102    | Shirom  | Science |
| T02  | Aditi | Statistics | 1       | 101    | Samayra | Science |
| T03  | Aditi | Statistics | 2       | 102    | Shirom  | Science |



## INTEXT QUESTIONS 15.1

## 1. Fill in the blanks:

- (i) Processed data is referred as .....
- (ii) Duplication of data results in the problem of .....
- (iii) The columns of a table are referred as .....
- (iv) The attribute which uniquely identifies the rows of a table is known as .....
- (v) When two or more attributes are used as Primary Keys, they are known as .....
- (vi) In RDBMS the data is organised in the form of a .....
- (vii) The number of attributes in a relation is known as .....
- (viii) A set of values from which attributes of a table draw their value is known as .....

## 2. Match the following:

- |                 |  |
|-----------------|--|
| (i) Foreign key | (a) Used to select specified attribute |
|-----------------|--|

## MODULE-2

Computer Applications  
(712)



Notes

- |                        |   |
|------------------------|---|
| (ii) Project Operation | (b) Introduced by E.F. Codd                           |
| (iii) Tuple            | (c) Used to represent the relationship between table. |
| (iv) Cardinality       | (d) Refer to rows of a table                          |
| (v) Relational model   | (e) Number of rows in a table                         |



### WHAT YOU HAVE LEARNT

- A primary key is a set of one or more attributes that can uniquely identify a row within a relation.
- All attributes in a relation that are capable of becoming a primary key are candidate keys.
- A candidate key which is not the primary key is called an alternate key. For example, in Student relation, if Roll\_no is chosen as primary key, the Adm\_no becomes the alternate key.
- A foreign key is used to represent the relationship between tables. A non-key attribute whose values are derived from the primary key of some other table is known as foreign key.



### TERMINAL EXERCISE

1. What is a database? Explain its need.
2. What do you mean by DBMS? What are its features?
3. What do you mean by Relational Algebra? Explain selection and projection operation.
4. What do you mean by Relation? Explain its properties.
5. Given the following relations, recognize primary key, alternate key, candidate key and foreign key.

#### Student

Adm\_No, Roll No, Board\_RollNo, Name, Address

#### Exam

Board\_RollNo, Mark1, Mark2





## MODULE-2

Computer Applications  
(712)



Notes

# 16

## MYSQL

---

### 16.1 INTRODUCTION

---

You have already learnt how DBMS provides an efficient and convenient environment to store data in a database. In this lesson you will learn about MySQL which is a Relational Database Management Systems (RDBMS), used to store, retrieve, modify and administrate a relational database using Structured Query Language (SQL). SQL creates an interface between the user and database. Also the SQL commands/queries help to create, transform and retrieve information from SQL Server, Oracle, Informix, Postgres, etc. You will learn about the SQL commands which you can use in database for doing data creation and data manipulation activities.



### OBJECTIVES

After reading this lesson, you will be able to:

- create databases and tables;
- insert records into the table;
- use alter table commands for adding, deleting and modifying the columns;
- update and delete the records in the table and
- use aggregate functions and perform calculations..

---

### 16.2 MYSQL

---

Developed in the mid-90s, MySQL was one of the first open-source Relational Database Management System. It is pronounced as '**My Sequel**'. It is used to

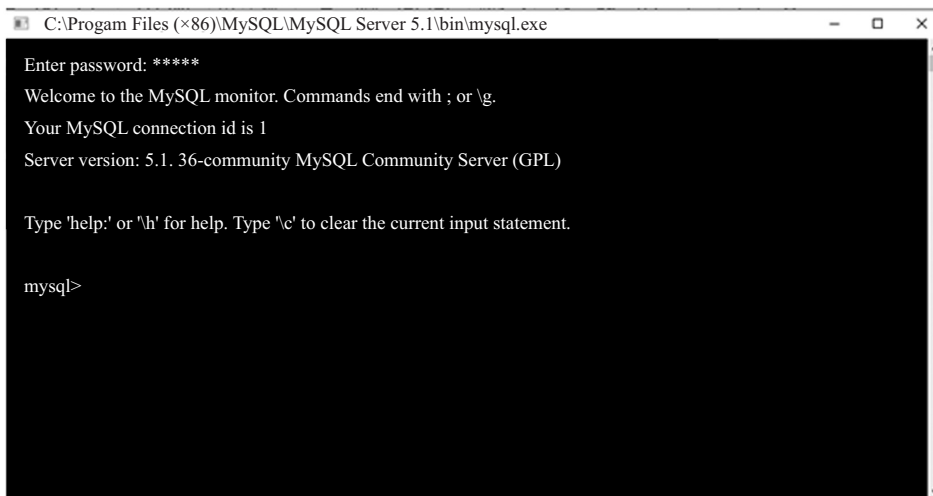
store, retrieve, modify and administrate a database using SQL. It is free to use and can be downloaded from MySQL official website.

### Features of MySQL

- It is easy to use. You can interact with MySQL with only a few simple SQL statements.
- It consists of a solid data security layer that protects sensitive data from intruders.
- It is compatible to run on many operating systems, like Windows\*, Linux\*, many varieties of UNIX\* (such as Sun\* Solaris\*, AIX, and DEC\* UNIX), OS/2, FreeBSD\*, and others.
- It is faster, more reliable and cheaper because of its unique storage engine architecture.
- It handles large databases.

### Start MySQL

Choose **'Start'** menu --> then choose **'Programs'**-->then choose **'MySQL'**. It opens a command line interface and prompts for providing password, type the user name and password (It requires the same password which was entered during the installation) (refer Fig 16.1).



```

C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin\mysql.exe
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.36-community MySQL Community Server (GPL)

Type 'help;' or 'h' for help. Type 'c' to clear the current input statement.

mysql>

```

Fig. 16.1: MySQL Command Line Client

To exit from MySQL, type **'quit'** or **'exit'**. As you know that, MySQL uses Structured Query Language (SQL) for accessing, retrieving the elements from the database. Let us learn about SQL now.



Notes



### Notes

### 16.3 SQL

It is a language specifically oriented around relational databases. It is a non-procedural language and requires us to specify what is to be done as opposed to how to do it. Some of the points to be kept in mind while using SQL are:

- SQL is a case insensitive language which means you can use uppercase, lowercase alphabets for writing the statements/queries. Each command is SQL terminated by a semicolon ‘;’
- SQL offers various commands using which you can create and manage relational databases. These commands are classified into following categories depending on their purpose:

**Data Definition Language (DDL):** It deals with database structure, and its descriptions of how the data reside in the database. Examples of DDL queries are – CREATE, DROP, ALTER.

**Data Manipulation Language (DML):** It consists of those commands that are used to retrieve, manipulate and update the data present in tables. It deals with data manipulation and includes most common SQL statements. Examples of DML queries are – INSERT, UPDATE, DELETE.

**Data Control Language (DCL):** DCL commands deal with the commands related to control the access to databases and tables. Examples of DCL queries are – GRANT, REVOKE.

In this lesson, you will learn about DDL and DML commands.

### 16.4 DATA TYPES

A data type specifies what kind of values can be stored in a column. In MySQL, the various data types can be used are:

- Numeric
- Character
- Date

#### 16.4.1 Numeric Data Types

These are mainly used to store numbers with or without fraction part. The most commonly used numeric data types are:



- **Integer:** This data type is used to store integer numbers (without any fraction part).

Syntax of Integer data type is:

**Integer or Int**

- **Decimal:** The decimal data type has fixed precision and scale.

The syntax of decimal data type is:

Syntax:

**Decimal[ (p[ , s] )]**

where:

- ‘p’ is the precision or the total number of significant decimal digits, where the most significant digit is the left-most non-zero digit and the least significant digit is the right-most known digit.
- ‘s’ is the scale or the number of digits from the decimal point to the least significant digit.

**Example:** Decimal (6,2) which can have maximum 6 digits with 2 digits after decimal.

### 16.4.2 Character Data Types

These are used to store character data. It can store all alphanumeric values. The most commonly used character data types are:

- **Char:** It is used for storing fixed length character strings between 1 to 255 characters in length. Maximum size of char data type is 2000 bytes.

The syntax for using char data type is:

**Syntax: Char (size)**

It can store fixed-length character strings having maximum length of size bytes.

- **Varchar:** It is a variable length string. If you declare data type as varchar, then it will occupy space for NULL values (NULL is used to represent unknown value. It is not the same as zero or a space or any other character.) It can have a maximum of 4000 bytes.

The syntax for using varchar data type is:

**Syntax: Varchar (size)**

It can store variable-length character string having maximum length of size bytes.



Let us see an example of creating variables of char, varchar data types.

### Example

#### Student\_Name char (15):

Here, if you store 'Student\_Name' variable's value as 'Kavita', then the first six places of the fifteen characters are filled with 'Kavita' and the remaining 9 spaces are filled with blank spaces. The size of the 'Student\_Name' is always fifteen.

#### Student1 varchar (15):

Once to declare this, if you store 'Student1' variable's value as 'Kavita', then the first six places of the fifteen characters will be filled with 'Kavita' and the remaining 9 spaces are filled with NULL value.

### 16.4.3 Date Data Type

It is used to store valid date values. The valid date formats are: YYYY-MM-DD or DD/MON/YY or YYYY/MM/DD or MM/DD/YY or DD-MON-YYYY. The default format is: 'YYYY-MM-DD'.

The syntax for using date data type is:

#### Syntax: Date

**Example:** join\_date date

Here join\_date is an attribute of date data type. Some of the valid values it can take are:

1945-11-22, 1998/05/15

Before storing data in a database, you need to first create a database and then open it. Let us understand commands to create and open a database.

---

## 16.5 CREATING AND OPENING A DATABASE

---

As you have learnt in the previous lesson that database is a collection of tables. Let us create database now using SQL command.

- **Create database:** This command is used to create a database.

The syntax for creating a database is:

```
CREATE DATABASE <DatabaseName>;
```



The DatabaseName can be any name given for the database. For example, to create a database employee, you can use the following command.

```
create database employee;
```

It will create an employee database, now the database is made active using 'USE' command.

```
USE <database name>;
```

This command will open the database. For example, to open the database **employee** you can use the following command.

```
use employee;
```

MySQL is a relational database system, thus all the data is stored in it in the form of tables. After creating a database the next step is to create tables to store the data.

---

## 16.6 CREATING A TABLE

---

A table consists of rows (records or tuples) and columns (fields or attributes). To create a table, you can use **CREATE TABLE** command.

The syntax for creating a table is:

### Syntax:

```
CREATE TABLE <TableName>  
  (<ColumnName1> <Data Type1>,  
  <ColumnName2> <Data Type2>,  
  ..... ,  
  <ColumnNameN> <Data TypeN>);
```

In the above command, TableName is the name of the table, ColumnName1 to ColumnNameN are the column names. Each column should have the information about what datatype you can use. Before creating a table, first you decide the data you are going to store in the table. Also think of the data types of the data which will be stored in the table.

Let us create a table using the following command.

### Example

Now, using create table command, you can create employee table, having the following columns:



Notes

employee number as integer,  
 name as character(10),  
 job as character(10),  
 mgr as integer,  
 hiredate as date,  
 salary as integer,  
 commission as integer and  
 department number as integer.

**create table emp(empno int, ename char(10), job char(10), mgr int, hiredate date, sal int, comm int, deptno int);**

The name of the table and name of the columns/attributes should not have space in between. Also, it cannot have special characters. Only underscore character is allowed. For eg.,

**Invalid names:** emp name, class-XII, grade\*

**Valid names:** emp\_name, class\_XII, grade

You can view the structure of the table by using **DESC** command.

Let us see the structure of **emp** table by giving the command:

**desc emp;**  
**or**  
**describe emp;**

```
mysql> Desc emp;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno | int(11) | YES  |     | NULL    |       |
| ename  | char(10) | YES  |     | NULL    |       |
| JOB    | char(10) | YES  |     | NULL    |       |
| MGR    | int(11) | YES  |     | NULL    |       |
| HIREDATE | date   | YES  |     | NULL    |       |
| sal    | int(11) | YES  |     | NULL    |       |
| COMM   | int(11) | YES  |     | NULL    |       |
| DEPTNO | int(11) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.04 sec)
```

Fig. 16.2: Describe Command





Notes

The above command, will display all the columns/attributes present in a given table, along with their data type (refer Fig. 16.2).

Now we have created a table, so the space for storing the data has been reserved. The next step is to store the data/records in the created table. But before adding data to your table, let us learn how you can change the structure of the table i.e., add/modify/delete columns of the table.

---

## 16.7 ADDING, MODIFYING AND DELETING COLUMNS

---

You have checked the name of the columns of the table. Now you want to add one new column 'age' to the **emp** table. In this case, you can use **ALTER TABLE** command. This command is used to add, modify and delete columns from a table.

To add a new column 'age' of type integer in the **emp** table, you can write the following command:

```
alter table emp add age integer;
```

The above command will add a new column 'age' in the **emp** table. Initially this column will not have any data. Later in this lesson, you will learn how to insert values in the newly added columns.

Suppose you now feel that the 'age' column in the **emp** table is no longer needed and you want to delete that column. For deleting the particular column you can use the following command:

```
alter table emp drop age;
```

You can also modify the existing column. The following command will change the size of 'ename' column from char(10) to char(30).

```
alter table emp modify ename char(30);
```

This is how, you can use alter table command for adding/modifying and dropping columns in the table.

---

## 16.8 ADDING RECORDS TO TABLE

---

After creating and editing the structure of the table, the next step is to add records to it. (record is a group of fields/columns/attributes within a table). This can be done using **INSERT INTO** command.

This command is used to add new record/tuple/row to a table. You can give values for all the columns present in the table by using this command.



The syntax for using insert into command is:

**Syntax:**

```
INSERT INTO <TableName> VALUES
    (<value1>, <value2>, ...<value n>);
Or
INSERT INTO <TableName>
    <Column1, Column2, ...Column n>
VALUES (<value1>, <value2>, ...<value n>);
```

The ‘**TableName**’ is the name of the table in which you want to add records. The Column1, Column2...Column n are the name of the columns/attributes/fields in the table (Example empno, ename, job etc., are the column names for the emp table). The value1, value2...valuen are the values for the corresponding columns. When you do not mention the column names, then you will have to give the values for all the columns in the predefined order. We will learn more about it in the following examples:

Let us see an example using insert into command.

**Example**

The following command will insert a record into the table **emp** with appropriate values for all its attributes,

```
insert into emp
values
(7876, "VINEET", "ANALYST", 7782, "2010-08-12", 9600, NULL, 20);
```

The above command, assigns value **7876** to ‘**empno**’, ‘**VINEET**’ to ‘**ename**’, ‘**ANALYST**’ to job, **7782** to mgr, ‘**2010-08-12**’ to hiredate, **9600** to sal, **NULL** to comm and **20** to deptno attributes respectively.

The values for char and date data types must be enclosed in quotes. The standard date format is “yyyy-mm-dd”. If you do not want to provide value for any particular column, then you can store NULL in that column. Insert command needs to be given as many times as the number of records you want to store in the table.

**Inserting Null Values**

There are two ways of inserting NULL values in the column/columns of the table. If you do not have any data for particular columns, you can leave them at the time of giving values in INSERT INTO command. For those columns NULL values will be inserted automatically. You can also insert NULL values explicitly like we did in the previous example.

Let us see an example, in which you will insert values to few columns in the table.

### Example

The following command will insert employee number as 7771, name as 'AKSHAT' and depart number as 20..

```
insert into emp (empno, ename, deptno) values (7771, "AKSHAT",20);
```

This command will insert a new record in the emp table. The columns empno, ename and deptno will be having the values given in the command. The remaining columns in the table will be having NULL values.



### INTEXT QUESTIONS 16.1

- ..... commands are used to change the structure of the table.
  - DDL
  - DML
  - DCL
  - TCL
- Create table command is a ..... command.
  - DDL
  - DML
  - DCL
  - TCL
- Choose the command used to open the STUDENT database.
  - open student;
  - use student;
  - create database student;
  - show student;
- Choose the command which is used to display the structure of the table STUDENT.
  - show struct student;
  - struct student;
  - desc student;
  - disp struct student;

### 16.9 SELECT COMMAND

Now, suppose you have inserted 15 records in the **emp** table using insert into command. You want to see all the records in the table. For viewing all the records, you can use the select command.



Notes



Notes

The syntax for using select command is:

**SELECT \* FROM <TableName>;**

Let us see an example.

**select \* from emp;**

```
mysql> select * from emp;
+-----+-----+-----+-----+-----+-----+-----+
| empno | ename  | JOB      | MGR  | HIREDATE | sal   | COMM  | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+
| 7876 | VINEET | ANALYST  | 7782 | 2010-08-12 | 9600  | NULL  | 20     |
| 7499 | AJAY   | CLERK    | 7876 | 2005-04-01 | 19200 | NULL  | 30     |
| 7698 | VIJAY  | MANAGER  | 7900 | 2001-07-02 | 15000 | NULL  | 30     |
| 7782 | DEEPA  | PRESIDENT | NULL | 2006-09-21 | 95700 | NULL  | 10     |
| 7782 | DEEPA  | PRESIDENT | NULL | 2006-09-21 | 95700 | NULL  | 10     |
| 7902 | KARTIK | SALESMAN | 7900 | 2009-05-05 | 15000 | 20    | 20     |
| 7900 | AASHA  | ANALYST  | 7782 | 2009-07-18 | 34200 | NULL  | 30     |
| 7566 | SANDEEP | ANALYST  | 7782 | 2007-06-22 | 29400 | NULL  | 20     |
| 7839 | HARI   | CLERK    | 7868 | 2005-02-26 | 36000 | NULL  | 30     |
| 7654 | JAYA   | MANAGER  | 7900 | 2003-04-19 | 60000 | NULL  | 30     |
| 7934 | RAMESH | CLERK    | 7876 | 2004-11-30 | 18000 | NULL  | 10     |
| 7788 | SUNDER | ANALYST  | 7782 | 2007-03-13 | 13200 | NULL  | 20     |
| 7369 | RANA   | CLERK    | 7876 | 2005-02-18 | 11400 | NULL  | 20     |
| 7844 | NISHTHA | MANAGER  | 7900 | 2007-01-17 | 36000 | NULL  | 30     |
| 7521 | RITA   | CLERK    | 7876 | 2008-09-22 | 15000 | NULL  | 20     |
+-----+-----+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

**Fig. 16.3: Output of Select Statement**

This command displays (refer Fig. 16.3) all the columns/attributes of all the records present in the table. If you want to display only specific columns of the table then you can specify the column names in the **SELECT** command instead of \*..

Let us see an example.

**Example**

The following command will display employee number and name of all the employees from emp table.

**select empno, ename from emp;**

You have to use the same column names as given at the time of creating a table in your SQL commands (refer Fig. 16.4). You can also perform calculations by using arithmetic operators in the select statement. Let us see how to use arithmetic operators in **SELECT** statement.



Notes

```
mysql> select empno, ename from emp;
+-----+-----+
| empno | ename |
+-----+-----+
| 7876  | VINEET |
| 7499  | AJAY  |
| 7698  | VIJAY |
| 7782  | DEEPA |
| 7782  | DEEPA |
| 7902  | KARTIK |
| 7900  | AASHA |
| 7566  | SANDEEP |
| 7839  | HARI  |
| 7654  | JAYA  |
| 7934  | RAMESH |
| 7788  | SUNDER |
| 7369  | RANA  |
| 7844  | NISHTHA |
| 7521  | RITA  |
+-----+-----+
15 rows in set (0.00 sec)
```

Fig. 16.4: Displaying Specific Columns Using SELECT Command

### Select Command with Arithmetic Operators

The arithmetic operators like +, -, \*, / can be used in the SELECT command to perform calculations.

Suppose in **emp** table, you wish to calculate annual salary of each employee, you can do so in the select statement itself. For calculating annual salary, we have to multiply monthly salary by 12 i.e.,  $sal * 12$ .

```
select ename, sal*12 from emp;
```

```
mysql> select ename, sal*12 from emp;
+-----+-----+
| ename | sal*12 |
+-----+-----+
| VINEET | 115200 |
| AJAY  | 230400 |
| VIJAY | 180000 |
| DEEPA | 1148400 |
| DEEPA | 1148400 |
| KARTIK | 180000 |
| AASHA | 410400 |
| SANDEEP | 352800 |
| HARI  | 432000 |
| JAYA  | 720000 |
| RAMESH | 216000 |
| SUNDER | 158400 |
| RANA  | 136800 |
| NISHTHA | 432000 |
| RITA  | 180000 |
+-----+-----+
15 rows in set (0.00 sec)
```

Fig. 16.5: Performing Calculation in SELECT



This command will display **name** and **annual salary** (which is obtained by multiplying salary of employees by 12) of all records.

You can observe the Fig. 16.5. Here you have calculated the annual salary, but the heading for that column is getting displayed as sal\*12. Instead of getting it displayed as sal\*12, if you want to display the output column name as '**Annual Salary**'. You can use column alias for this case. Let us discuss about column alias.

### Using Column Alias

Column alias is used to change the display name of a column. It does not rename the column. It simply displays a different column name in the output. You can use 'AS' keyword to specify the alias name. But 'AS' keyword is optional.

The above command may be rewritten as:

```
select ename, sal*12 as "Annual Salary" from emp;
```

This command will display employee name and annual salary of all the employees from **emp** table. Instead of sal\*12, the column name will be displayed as '**Annual Salary**'.

### Eliminating Duplicate Values

By using '**Distinct**' keyword in SELECT command you can eliminate the duplicate values and display the unique data in the table.

The command to display different types of jobs present in the **emp** table is:

```
select distinct job from emp;
```

```
mysql> select distinct job from emp;
+-----+
| job   |
+-----+
| ANALYST
| CLERK
| MANAGER
| PRESIDENT
| SALESMAN
+-----+
5 rows in set (0.00 sec)
```

**Fig. 16.6: Eliminating Duplicate Values**

You can see that in the job column, each type of job is appearing only once (refer Fig.16.6). This happens because of the distinct keyword which eliminates the duplicate entries in the table.



Notes

## Where Clause

You can use the where clause in the **SELECT** command to display the data based upon the specified condition. To specify the condition, you can use all types of relational and logical operators in the command.

### Relational Operators in Where Clause

The table 16.1 shows the relational operators that can be used in the where clause:

**Table 16.1: Relational Operators**

| Operator | Description              |
|----------|--------------------------|
| =        | Equal to                 |
| <        | Less than                |
| >        | Greater than             |
| >=       | Greater than or equal to |
| <=       | Less than or equal to    |
| != or <> | Not equal to             |

Earlier you tried to display the distinct job names from emp table. Now if you want to display the details of the employee who are presently working as ‘**MANAGER**’, then you can use the following command:

```
select * from emp where job="MANAGER";
```

```
mysql> Select * from emp where job="MANAGER";
+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | ename  | JOB      | MGR  | HIREDATE | sal   | COMM | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 7698  | VIJAY  | MANAGER  | 7900 | 2001-07-02 | 15000 | NULL | 30     |
| 7654  | JAYA   | MANAGER  | 7900 | 2003-04-19 | 60000 | NULL | 30     |
| 7844  | NISHTHA | MANAGER  | 7900 | 2007-01-17 | 36000 | NULL | 30     |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

**Fig. 16.7: Fetching Conditioned Data Using Where Clause**

You can observe that the above command will display only those records present in the table **emp** whose **job** is ‘**Manager**’ (refer Fig. 16.7). In the command, the relational operator ‘=’ is used in the where clause.

In the next example, the relational operator ‘>’ will be used to display employee number, name and salary of those employees whose salary is more than 30000.

```
Select empno, ename, sal from emp where sal>30000;
```



Notes

```
mysql> Select empno, ename, sal from emp where sal>30000;
+-----+-----+-----+
| empno | ename  | sal   |
+-----+-----+-----+
| 7782  | DEEPA  | 95700 |
| 7782  | DEEPA  | 95700 |
| 7900  | AASHA  | 34200 |
| 7839  | HARI   | 36000 |
| 7654  | JAYA   | 60000 |
| 7844  | NISHTHA | 36000 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

**Fig. 16.8: Fetching Conditioned Data Using Relational Operator**

You can see the output in Fig. 16.8. It displayed the **employee number (empno)**, **name (ename)** and **salary (sal)** of those employees whose **salary** is more than 30000. Try using >= or <= operators and observe the difference with respect to > and < operators.

Let us see the use of logical operators in where clause.

**Logical Operators In Where Clause**

The logical operators listed in Table 16.2 are used to join two or more conditions in the SELECT command. The various types of logical operators which can be used are:

**Table 16.2: Logical Operators**

| Operator | Description   |
|----------|---|
| NOT      | Negates a condition.  |
| AND / && | If both the given conditions are true then it will become true. |
| OR /     | If any of the condition is true then it will become true.       |

If you want to display **empno, ename** from emp table whose salary is greater than or equal to 5000 and less than 10000, then you can use:

```
select empno, ename from emp where sal>5000 and sal<10000;
```

Here, the emp table was having only one record of the emp with the name **‘VINEET’** whose salary is greater than 5000 and less than 10000 (refer Fig. 16.9).





Notes

```
mysql> select empno, ename from emp where sal >5000 and sal < 10000;
+-----+-----+
| empno | ename |
+-----+-----+
| 7876  | VINEET |
+-----+-----+
1 row in set (0.00 sec)
```

Fig. 16.9: Applying Logical Operator - AND

If you want to display empno, empname of emp table whose job is analyst or clerk, then you can use the command:

```
select empno, ename from emp where job="ANALYST" OR job="CLERK";
```

The output of the command can be seen in Fig. 16.10.

```
mysql> Select empno, ename from emp where job="ANALYST" or job="CLERK";
+-----+-----+
| empno | ename |
+-----+-----+
| 7876  | VINEET |
| 7499  | AJAY  |
| 7900  | AASHA |
| 7566  | SANDEEP |
| 7839  | HARI  |
| 7934  | RAMESH |
| 7788  | SUNDER |
| 7369  | RANA  |
| 7521  | RITA  |
+-----+-----+
9 rows in set (0.00 sec)
```

Fig. 16.10: Using Logical Operator OR

Suppose in the emp table, if you want to display employee number and department number of those employees whose department number is not 20, then you can use the command:

```
select empno, deptno from emp where deptno != 20;
```

The output of the command can be seen in Fig. 16.11.

```
mysql> Select empno, deptno from emp where deptno != 20;
+-----+-----+
| empno | deptno |
+-----+-----+
| 7499  | 30    |
| 7698  | 30    |
| 7782  | 10    |
| 7782  | 10    |
| 7900  | 30    |
| 7839  | 30    |
| 7654  | 30    |
| 7934  | 10    |
| 7844  | 30    |
+-----+-----+
9 rows in set (0.00 sec)
```

Fig. 16.11: Using Logical Operator - NOT



Notes

**Like Operator – Pattern Matching**

The like operator is used to fetch the data from the table which matches a specified pattern. It uses two wild card characters (refer table 16.3):

**Table 16.3: Like Operators**

| Wild Card Character | Description  |
|---------------------|--|
| % (percentage)      | It is used to represent any sequence of zero or more characters. |
| _ (underscore)      | It is used to represent a single character.                      |

**Example**

The following command will display names of those employees from emp table whose name starts with ‘A’.

```
select ename from emp where ename like “A%”;
```

Here like “A%” means it will display all the employee name starts with ‘A’ only and can be of any leangth (refer 16.12).

```
mysql> Select ename from emp where ename like "A%";
+-----+
| ename |
+-----+
| AJAY  |
| AASHA |
+-----+
2 rows in set (0.00 sec)
```

**Fig. 16.12: Using Like Operator - %**

Let us see another command, which will display the names of those employees from the emp table whose name is having ‘A’ alphabet.

```
select ename from emp where ename like “%A%”;
```

```
mysql> Select ename from emp where ename like "%A%";
+-----+
| ename |
+-----+
| AJAY  |
| VIJAY |
| DEEPA |
| DEEPA |
| KARTIK|
| AASHA |
| SANDEEP|
| HARI  |
| JAYA  |
| RAMESH|
| RANA  |
| NISHTHA|
| RITA  |
+-----+
13 rows in set (0.00 sec)
```

**Fig. 16.13: Displaying Employee Names having Alphabet ‘A’**

As you know that ‘%’ represents any sequence of zero or more characters. Here, the output is the names of all the employees (refer Fig. 16.13) whose names either starts with ‘A’ or ends with ‘A’ or has ‘A’ at any other position in the ename column.

If you want to display the name of employees whose name is at least 5 characters long, then you can use the following command:

```
select ename from emp where ename like "_____";
```

```
mysql> Select ename from emp where ename like "_____";
+-----+
| ename |
+-----+
| VINEET |
| VIJAY |
| DEEPA |
| DEEPA |
| KARTIK |
| AASHA |
| SANDEEP |
| RAMESH |
| SUNDER |
| NISHTHA |
+-----+
10 rows in set (0.00 sec)
```

Fig. 16.14: Displaying Names With Minimum Five Characters

Here, the pattern has five underscores and one ‘%’. The output of this command will display the names of all the employees whose name is five or more characters long (refer Fig. 16.14).

Let us try to write the above command without ‘%’ symbol.

```
select ename from emp where ename like "_____";
```

```
mysql> Select ename from emp where ename like "_____";
+-----+
| ename |
+-----+
| VIJAY |
| DEEPA |
| DEEPA |
| AASHA |
+-----+
4 rows in set (0.00 sec)
```

Fig. 16.15: Displaying Names With Five Characters

Now observe the output. Here, the pattern has only five underscores and the output will be the names of all employees whose name is exactly five characters long (refer Fig. 16.15). Each underscore is replaced by one single character.



Notes



Notes

You can try with NOT LIKE operator which will be used to select the rows that do not match the specified pattern.

**Order By Clause**

This clause is used to arrange the results of the **SELECT** command in either ascending or descending order. The default order is ascending order. However, you can also add **ASC** keyword for ascending order. To arrange the data in descending order, use the keyword **DESC**. This clause can also be used to arrange the data in a particular order on multiple columns.

**Example**

The following command will display the data of employees from emp table in ascending order of their salary.

```
select * from emp order by sal;
or
select * from emp order by sal asc;
```

```
mysql> Select * from emp order by sal;
```

| empno | ename   | JOB       | MGR  | HIREDATE   | sal   | COMM | DEPTNO |
|-------|---------|-----------|------|------------|-------|------|--------|
| 7876  | VINEET  | ANALYST   | 7782 | 2010-08-12 | 9600  | NULL | 20     |
| 7369  | RAMA    | CLERK     | 7876 | 2005-02-18 | 11400 | NULL | 20     |
| 7788  | SUNDER  | ANALYST   | 7782 | 2007-03-13 | 13200 | NULL | 20     |
| 7521  | RITA    | CLERK     | 7876 | 2008-09-22 | 15000 | NULL | 20     |
| 7902  | KARTIK  | SALESMAN  | 7900 | 2009-05-05 | 15000 | 20   | 20     |
| 7698  | VIJAY   | MANAGER   | 7900 | 2001-07-02 | 15000 | NULL | 30     |
| 7934  | RAMESH  | CLERK     | 7876 | 2004-11-30 | 18000 | NULL | 10     |
| 7499  | AJAY    | CLERK     | 7876 | 2005-04-01 | 19200 | NULL | 30     |
| 7566  | SANDEEP | ANALYST   | 7782 | 2007-06-22 | 29400 | NULL | 20     |
| 7900  | AASHA   | ANALYST   | 7782 | 2009-07-18 | 34200 | NULL | 30     |
| 7839  | HARI    | CLERK     | 7868 | 2005-02-26 | 36000 | NULL | 30     |
| 7844  | NISHTHA | MANAGER   | 7900 | 2007-01-17 | 36000 | NULL | 30     |
| 7654  | JAYA    | MANAGER   | 7900 | 2003-04-19 | 60000 | NULL | 30     |
| 7782  | DEEPA   | PRESIDENT | NULL | 2006-09-21 | 95700 | NULL | 10     |
| 7782  | DEEPA   | PRESIDENT | NULL | 2006-09-21 | 95700 | NULL | 10     |

15 rows in set (0.06 sec)

**Fig. 16.16: Order By Clause**

Here, you can see that the entire data of table is sorted in ascending order of salary (refer Fig. 16.16).

To display the data of all the employees from emp table in descending order of their salary, you can use:

```
Select * from emp order by sal desc;
```



## INTEXT QUESTIONS 16.2

1. Consider the table 'empsal'.

**ID Salary**

101 54000

103 NULL

105 75000

109 NULL

Choose the command to display ID of those employees whose salary is other than NULL.

- (a) select ID from empsal where salary <> NULL;
  - (b) select ID from empsal where salary is NULL;
  - (c) select ID from empsal where salary is not NULL;
  - (d) select ID from empsal where salary not is NULL;
2. Consider the table 'empsal'.

**ID Salary Dept**

101 54000 10

103 NULL 20

105 75000 20

109 NULL 30

Choose the command to display the following output:

**Dept****10****20****30**

- (a) select dept from empsal;
- (b) select dept from empsal where dept<> duplicate;
- (c) select distinct dept from empsal;
- (d) select unique dept from empsal;



Notes



Notes

3. Which wild card character is used to replace a single character in string matching pattern in SQL?
  - (a) @
  - (b) %
  - (c) #
  - (d) \_
  
4. Which clause is used to arrange the data in any particular order in SQL?
  - (a) Order by
  - (b) Group by
  - (c) Like
  - (d) Is null

**16.10 AGGREGATE FUNCTIONS**

The aggregate functions work on multiple rows and perform certain calculations and return a single value as a result. There are five types of aggregate functions:

**Table 16.4: Aggregate Functions**

| Aggregate Function | Purpose  |
|--------------------|--|
| max()              | To find the maximum value of the specified column.     |
| min()              | To find the minimum value of the specified column.     |
| avg()              | To find the average of values of the specified column. |
| sum()              | To find the sum of values of the specified column.     |
| count()            | To count the number of values of the specified column. |

Let us learn these functions one by one by using table emp.

First let us see some examples of max() function given in Table 16.5.

**Table 16.5: Examples of Max() function**

| Command                                       | Purpose  | Result |
|---|--|--------|
| Select max(sal) from emp;                     | It displays the maximum salary of an employee.   | 97700  |
| Select max(sal) from emp where job="MANAGER"; | To find the highest salary paid to MANAGERS you can add a condition where job="Manager". | 74400  |

Let us see some examples using min() function given in table 16.6.

**Table 16.6: Examples of Min( ) Function**

| Command                                       | Purpose  | Result |
|---|--|--------|
| Select min(sal) from emp;                     | To find the lowest salary paid to an employee.           | 11600  |
| Select min(sal) from emp where job='MANAGER'; | To find the lowest salary paid to MANAGERS.              | 20400  |
| Select min(sal+comm*sal) from emp;            | To find the lowest salary + commission paid to employee. | 357000 |



Notes

Let us see some examples using avg() function given in table 16.7.

**Table 16.7: Examples of avg( ) Function**

| Command                                       | Purpose   | Result     |
|---|---|------------|
| Select avg(sal) from emp;                     | To find the average salary paid to an employee from emp table | 37253.3333 |
| Select avg(sal) from emp where job='MANAGER'; | To find the average salary paid to MANAGERS                   | 46800.0000 |

The avg() function can be performed on numeric (int/decimal) type only. The avg() function considers only non NULL values in the specified column.

Let us see some of the examples of sum() function.

Please refer to the emp table for the commands given in table 16.8.

**Table 16.8: Examples of sum() Function**

| Command                                       | Purpose   | Result |
|---|---|--------|
| Select sum(sal) from emp;                     | To find the sum of salary paid to all employees from emp table. | 558800 |
| Select sum(sal) from emp where job='MANAGER'; | To find the sum of salary paid to MANAGERS.                     | 140400 |

Let us see some of the examples of count() function. The count() function takes one argument which can be any column name, an expression based on a column, or an asterisk (\*).



- count(column name) - returns the number of non-NULL values in the column.
- count(\*) - returns the total number of rows satisfying the condition including NULL values.

Table emp is used for the following commands given in table 16.9:

**Table 16.9: Count(\*) and Count( )**

| Command                              | Purpose  | Result |
|--------------------------------------|--|--------|
| Select count(*) from emp;            | To count the total number of employees in the table emp. | 15     |
| Select count(distinct job) from emp; | To count the distinct jobs in the emp table.             | 5      |

**16.11 GROUP BY**

The group by clause is used to make groups of identical data present in multiple rows of the table by one or more columns. For every group of rows only one single row will be returned. It is used in **SELECT** statement and follows **WHERE** clause and precedes **ORDER BY** clause. It is often used with aggregate functions.

Let us see an example of group by clause.

**Example**

The following command will display the maximum salary paid to employees of different kinds of jobs.

```
select job, max(sal) from emp group by job;
```

| JOB       | max(sal) |
|-----------|----------|
| PRESIDENT | 95700    |
| CLERK     | 36000    |
| ANALYST   | 34200    |
| MANAGER   | 60000    |
| SALESMAN  | 15000    |

**Fig. 16.17: Using Max() Function**

Here, first all the rows with same types of job are grouped together and then maximum salary from each group is calculated. For every group only one row will be displayed as output (refer Fig. 16.17).



You can include more than one aggregate function in one single command. Let us see an example which uses two aggregate functions in one statement.

The following command will display maximum and minimum salary paid to employee of each type of department.

```
select deptno, max(sal), min(sal) from emp group by deptno;
```

| DEPTNO | max(sal) | min(sal) |
|--------|----------|----------|
| 10     | 95700    | 18000    |
| 20     | 29400    | 9600     |
| 30     | 60000    | 15000    |

Fig. 16.18: Using Max() and Min()

Here, first all the rows with same type of deptno are grouped together and then maximum and minimum salary from each group is calculated. For every group only one row will be displayed as output (refer Fig. 16.18).

### Having Clause

The having clause is used to filter groups specified by **group by** clause based on a certain condition. When the having clause is used without group by clause it behaves like a **where** clause.

Let us see an example using having clause.

### Example

The following command will display the maximum salary paid to employees of each type of job whose maximum salary is greater than 50,000 (refer Fig. 16.19).

```
select job, max(sal) from emp group by job having max(Sal)>50000;
```

| JOB       | max(sal) |
|-----------|----------|
| PRESIDENT | 95700    |
| MANAGER   | 60000    |

Fig. 16.19: Using Group BY and Having Clause

The major difference between the having and the where clause is that where clause is used to put a filter on individual rows while having clause is used to put a filter on the group of rows. You can also use both where and having clauses in the same command/query.



Notes



Notes

## 16.12 UPDATE COMMAND

Sometimes you may need to update the existing data in the table. The **Update** command will help you to modify the data present in the table.

The syntax of update command is as follows:

**Syntax:**

```
UPDATE <Table_Name>
SET <column name> = <value>,
[ <column name> = <value>, ...]
[WHERE <condn>];
```

The following command will update the salary of all the employees of emp table.

```
update emp set sal=sal+2000;
```

This command updates salary of all the employees (by sal=sal+2000) by increasing it to Rs.2000 more.

To update the salary details of a particular employee, you can use where clause in update statement. The following command will increase the salary by Rs.2000, and change department number to 20 for the employee whose employee number is 7521.

```
update emp set sal=sal+2000, deptno=20 where empno=7521;
```

Here, 2000 is added to salary and department number is set to 20 to the employee whose employee number is 7521.

## 16.13 DELETE STATEMENT

You can delete the data of the table by using DELETE command.

The syntax for using this command is:

**Syntax:**

```
DELETE FROM <TableName>[ WHERE<condn>];
```

To delete the data of those employee whose salary is less than 6000, you can use the following command

```
delete from emp where sal<6000;
```

```
mysql> Delete from emp where sal<6000;
Query OK, 0 rows affected (0.04 sec)
```

Fig. 16.20: Using Delete Command

Here, only those rows will be deleted from the table where salary is less than 6000.

To delete all the rows of table emp, simply write the statement:

**delete from emp;**

Here, this statement will delete all the rows from the emp table.



Notes

### 16.14 TRUNCATE COMMAND

It is used to delete all the rows from a table in one go. It is similar to the delete all.

The syntax is as follows:

**TRUNCATE TABLE <TableName>;**

If you want to delete the rows from the emp table then you can write the command as, truncate table emp; This command will delete all the entries/rows from the emp table, however, the structure of the table will remain. You can use the table to insert new rows again.

### 16.14 DROP COMMAND

Drop command is used to remove the structure of the table from the memory. The syntax for using drop command is as follows:

**DROP TABLE <TableName>;**

If you want to remove the structure of emp table, then you can write the command as:

**drop table emp;**

This will remove the structure of emp table.



### INTEXT QUESTIONS 16.3

1. Which command is used to modify a row of a table?
 

|                          |                             |
|--------------------------|-----------------------------|
| (a) Modify Column        | (b) Alter Table .....Modify |
| (c) Alter Table .....Add | (d) Update                  |
2. Which of the following is not an aggregate function?
 

|             |           |
|-------------|-----------|
| (a) avg()   | (b) sum() |
| (c) total() | (d) max() |



Notes

3. The SUM() function cannot be applied to which data type .....  
(a) int (b) decimal  
(c) date (d) float
4. Which of the following statements is correct?  
(a) delete from emp; (b) delete \* from emp;  
(c) delete emp; (d) delete where emp;
5. The count(\*) function counts the number of non-null values of the table .....  
(a) yes (b) no



### WHAT YOU HAVE LEARNT

- MySQL is used to store, retrieve, modify and administrate a database using SQL.
- SQL is the language used to manipulate and manage databases and tables. All the commands which are used to create, destroy, or restructure databases and tables come under DDL category. Examples of DDL commands are - CREATE, DROP, ALTER.
- All the commands which are used to manipulate data within tables come under DML category. Examples of DML queries are - INSERT, UPDATE, DELETE.
- Create Database command is used to create a new database.
- Create Table command is used to create a new table.
- The Describe command is used to view the structure of a table.
- You can change the structure of a table i.e., add, remove or change column(s) in a table, using the ALTER TABLE statement.
- The INSERT INTO command is used to insert a new row in a table.
- The SELECT statement is used to fetch data from one or more database tables.
- The keyword **Distinct** is used to eliminate redundant data from display.
- The logical operators OR and AND are used to connect relational expressions in the **Where** clause and the logical operator NOT is used to negate a condition.
- The **Like** operator is used to fetch data from the table which matches a specified pattern. It uses two wild card characters: '%' and '\_'. The percent



## Notes

(%) symbol is used to represent any sequence of zero or more characters. The underscore (\_) symbol is used to represent a single character.

- The NULL represents a value that is unavailable, unassigned, unknown or inapplicable.
- The results of the **SELECT** command can be displayed in an ascending or descending order of a single column or columns using Order By clause.
- The **UPDATE** command is used to modify existing data in a table.
- The **DELETE** command is used to delete rows from a table.
- The **DROP** command is used to remove the structure of the table from the memory.
- The aggregate functions work on multiple rows. There are 5 types of aggregate functions: max(), min(), sum(), count(), avg().



## TERMINAL EXERCISE

1. What is NULL? What happens when you perform arithmetic calculations on NULL values?
2. List the categories in which the SQL commands are categorized giving two examples of each.

Consider a table SALESMAN with the following data:

| S.No. | Sname       | Salary | Bonus | %     | Date of Join |
|-------|-------------|--------|-------|-------|--------------|
| 1     | T.K. Gupta  | 300000 | 36    | 12.00 | 19-07-2009   |
| 2     | S.P. Singh  | 500000 | 48    | 10.00 | 22-03-2008   |
| 3     | N.P. Jain   | 300000 | 36    | NULL  | 08-03-2007   |
| 4     | MohitYadav  | 800000 | 60    | 10.00 | 06-12-2008   |
| 5     | Ajay Sinha  | 200000 | 36    | 12.50 | 03-01-2010   |
| 6     | Ravi Sharma | 700000 | 60    | 12.50 | 05-06-2008   |
| 7     | K.S.Garg    | 500000 | 48    | NULL  | 05-03-2008   |

Write SQL commands for the tasks 1 to 15 and write the output.



### **Create Database and use it**

1. Create the database SALES.
2. Use the database SALES.

### **Create Table / Insert Into**

3. Create the table SALESMAN and insert tuples in it.

### **Simple Select**

4. Display the details of all the SALESMAN.
5. Display the sales man no, name, and salary of all the salesman.

### **Conditional Select using Where Clause**

6. Display the details of all the salesman with salary less than 300000.
7. Display the sno and salary of all the salesman who joined before 01-04-2009.

### **Using NULL**

8. Display the details of all the salesman whose bonus is NULL.

### **Using DISTINCT Clause**

9. Display the salary of various salesman from the table without duplicate values.

### **Using ORDER BY clause**

10. Display the details of all the salesmans in the descending order of their join date.

### **Using UPDATE, DELETE, ALTER TABLE**

11. Increase the bonus by 0.5% for all the salesman.
12. Delete the records of all the salesmans whose join date is before 2007.
13. Add another column Category of type CHAR(5) in the Salesman table.



## **ANSWERS TO INTEXT QUESTIONS**

### **16.1**

1. (a) DDL
2. (a) DDL
3. (b) USE STUDENT;
4. (c) DESC STUDENT;



Notes

**16.2**

1. (d) select ID from empsal where salary not is NULL;
2. (c) select distinct dept from empsal;
3. (d) \_
4. (a) ORDER BY

**16.3**

1. (d) UPDATE
2. (c) Total()
3. (c) date
4. (a) delete from emp;
5. (b) no

**Key Learning Outcomes**

- Be proficient in writing commands to create, alter, update and delete the tables.
- Be able to apply aggregate functions in select commands and also perform sorting and grouping of data using order by and group by clauses.

## MODULE-2

Computer Applications  
(712)



Notes

# 17

## WORKPLACE PRACTICES AND WORK MANAGEMENT

Workplace can be defined as an environment where people work to produce a product or to provide a service. When we move from the academic world to the professional world, we need some personality traits to fit ourselves in the corporate sector. Education paves the specific way for your career path, but it is important that you groom yourself into a confident and a good human being; These qualities are required in all the individuals at any workplace in order to make it a great place to work. A working individual usually spends most of his/ her time everyday at his/ her workplace; thus it becomes utterly important to have a comfortable and enthusiastic environment at the workplace. It plays an important role in enhancing the overall productivity and growth of an individual. In this lesson, we will discuss workplace practices and skills to manage work efficiently.



### Objectives

After reading this lesson, you will be able to:

- explain the importance of workplace practices;
- apply work management tips;
- manage stress better;
- take steps to avoid and manage anger.

---

### 17.1 Workplace Practices

---

The workplace culture can vary from organization to organization. Some workplaces may provide its workers the flexibility in terms of work schedule but in others there might be a strict schedule to be followed by the worker. It is important to





Notes

understand that though there are different methods of operation, none of the organizations would compromise either with the deadline for the work to be submitted or with the quality of work to be submitted. At this point, it is important that we understand to become a valued employee in the organisation. We as an employee should be vigilant and core to the rules and policies followed at the workplace. Following are the few workplace practices that are universally applicable to all the workers in any organization.

- **Be Punctual:** Being on time at your workplace will earn you respect from everyone. It will help you to manage things better at work and will ensure that you are aware of all the updates and happenings related to your work and the organization.
- **Be Aware of Your Role:** The success of an organization depends on how efficiently its employees complete their assigned tasks. Always remember you are being paid to produce an output hence, you should be aware of your role and the short term and long term goals you are expected to achieve. The ability to identify the nature and purpose of work, keeps the worker motivated and also benefits the organization.
- **Know the Organizational Chart:** An organizational chart depicts the structure of an organization. It displays a reporting or relationship hierarchy. Having a clear idea of the organizational chart can help you to understand the work flow and identify your reporting supervisors or subordinates. It also gives clarity of communication with respect to the assigned work. An example of NIOS organisation structure is shown in Fig. 17.1.

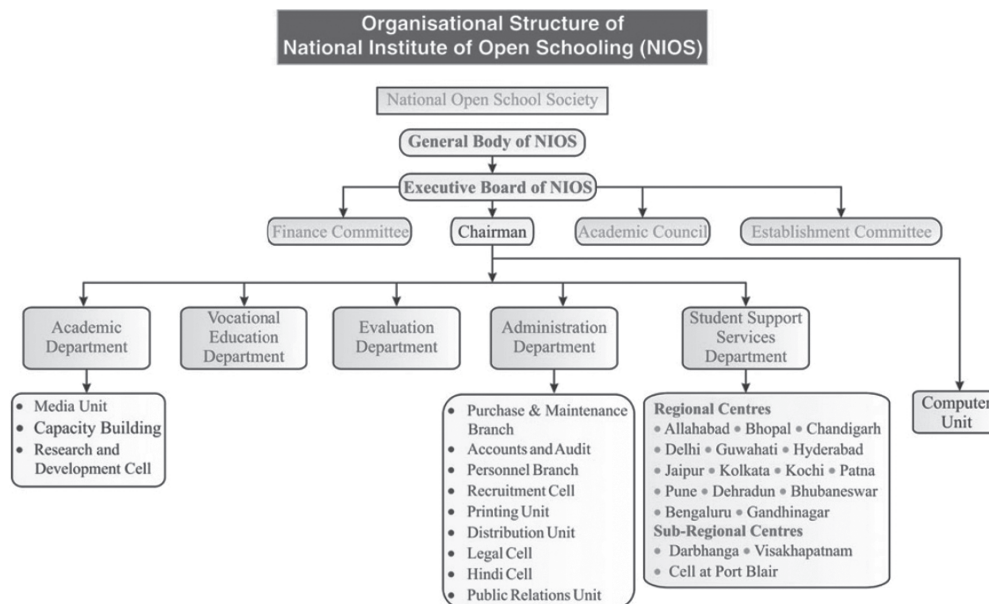


Fig. 17.1: NIOS Organisational Structure

## MODULE-2

Computer Applications  
(712)



Notes

### Workplace Practices and Work Management

- **Communication:** It plays a key role in an organization at all levels such as within colleagues, with superiors, with subordinates and the most importantly with the customer. The communication should be clear, concise and effective. It requires the understanding of the language being used at the workplace that may include certain technical terminologies. Any organization that at some level deals with customers and thus it lays down strict rules for the language to be used of which the employees are made aware of, at the time of induction. Make sure, you bring them into practice when interacting with the customer. Also, while interacting with the colleagues, it is always advisable to be alert and use words judiciously.
- **Team Building:** It is important to understand that no employee can work in isolation in an organization. So, team work plays an important role in the effective and efficient achievement of goals. A mutual trust and accountability is required on part of every worker in the organization.
- **Respect:** To earn respect, it is important that you give respect to others. An employee spends a large part of the day at workplace. Everyone has to work with each other and many a times issues such as privacy, cleanliness, and other distractions play a major role in disturbing the environment while working. At this point, it is important to respect the working environment of co-workers. This would result in nurturing of a conducive workplace.



#### Intext Questions 17.1

1. Fill in the blanks:
  - (a) Being ..... is the act of being on time always.
  - (b) An ..... is a diagram that depicts the structure of an organization, it displays a reporting or relationship hierarchy.
  - (c) Communication should be clear ..... and .....

#### 17.2 Work Management

Knowledge of workplace practices can help you be at ease with the environment but at the end of the day, it is your work management skills that can help you to enjoy your workplace and make you an efficient employee. Though each person can adopt different work management methods and still be able to deliver work with perfection on time, but there might be people who are unable to do so. This might lead to stress which eventually can lead to anger that can affect your health and career as well.



---

**Notes**

Following are few essential work management tips:

- **Plan:** Have a lot of work to do? It is always better to prepare a '**To Do**' list to keep a track of your work. Make sure that you keep updating your to do list so that it reflects how much you have achieved. This will help you to stay motivated.
- **Prioritize:** It is important to note that you cannot do everything at once. Rather, looking at the list of things sometimes sends your mind into a confused state. Hence, it is advised to prioritize work in order of the urgency and time required. Reordering the '**To Do**' list as per priority will surely help. This will help you to focus on each work, one at a time.
- **Stay Updated:** Do you need to generate a letter for 100 recipients along with mailing labels? Instead of typing each recipient's name once on the letter and then the mailing label you can make use of mail merge feature of word processor. Similarly, if you want a document to be reviewed by n number of people you can use Google docs as it allows collaborative work. Hence, keep yourself updated with the latest technology that can help you to work faster and more efficiently.
- **Take Timeout:** Giving yourself short break after a significant time would really help you to manage stress and think better. Allowing yourself time to relax will help you feel better and prepare you to handle challenges without being stressed.
- **Follow Guidelines:** Do you start working without carefully going through the details? Well that can lead to delay in work or a complete waste of work. Carefully read the instructions, study the format or understand the details before starting the work. You should have a clear picture of the outcome of your work.
- **Communicate:** Never hesitate to take help. In case you are stuck, discussing with your seniors or your colleagues will surely help and solve a complex problem in a lesser time frame.
- **Be Grateful:** Did your last work get stuck? Were you able to complete it with your colleague's advice? If yes. Did you express gratitude to him/her? You should always give credit and recognize the contribution of your colleagues in your work. This helps you to build a good rapport and earn respect.
- **Never Boast:** You should always work for perfection and not recognition. Recognition is a byproduct of sincerity and hard work. You should never boast of your achievements; rather your work should do the talking.



### Intext Questions 17.2

1. State whether True or False.
  - (a) You can do everything without planning.
  - (b) You should always express gratitude towards people who have helped you in completion of your work.
  - (c) Taking short breaks can help manage stress.

---

### 17.3 Self Management

---

Self awareness is one of the ten life skills that WHO (World Health Organization) promotes. Self awareness is the honest knowledge of self. None of us is born perfect. Everyone has strengths and weaknesses. To be able to use our strength to maximize the output, acknowledging weakness and working to convert it to strength is self awareness. It is the one of the most important requirements to manage oneself better.

#### 17.3.1 Anger Management

In today's busy world, nearly all of us come across stress in some form or other at our workplace. Many a times it gets converted into anger. The outcome of dissatisfaction is anger. It is an expression of helplessness that is projected by an emotional outburst or withdrawal. It has serious implications on our health, relationships and workplace. Have you ever given it a thought that what do we do when someone else expresses his/her anger what do we do? Sometimes we tend to ignore or many a times react instead of addressing the person's concern. The same happens when we get angry. No one has the time and patience to understand our reason of getting angry. It is better to learn to manage anger in a positive way. In order to manage anger, we can do the following:

- **Be Thoughtful:** Take time out and select your words wisely. Think before you speak as one wrong word, can cost you dearly. So, it is wise to avoid words that you would later regret using.
- **Express Yourself When in Control:** Once you are in control of your emotions you can express your dissatisfaction but in a non confrontational way. Also, keep yourself prepared to move on, in case your dissatisfaction is not addressed with a clear concluding remark.



Notes

- **Divert Your Attention:** An increasing stress can trigger anger, and you are the best judge of the situation that you are in. So, whenever you feel that you are overstressed, divert your attention to something enjoyable such as listening to music or a quick chat with someone you are comfortable with or if possible go for a brisk walk or jog. Physical activity has the capability of bringing your stress down. Getting angry is never a solution rather it will rob you of your calmness and paralyze your thinking process.
- **Avoid Blame Game:** You may tell your side of story without blaming others. Blaming others will make the other person feel offended and would create an awkward situation for both of you.
- **Deep Breathe:** Deep breathing lowers stress level, reduces anxiety and alleviates negative feeling. Thus it will help you to be in control of situation.
- **Seek Help:** Controlling anger might be a challenging task but not being able to control it might have a bigger impact on your career. Hence, if you are unable to manage anger despite sincere efforts, it is advised to seek help.



**What You have Learnt**

- Workplace can be defined as an environment where an individual works to produce a product or service.
- None of the organization would compromise with the deadline for the work to be submitted.
- If an employee is vigilant and cares for the rules and policies followed at the workplace he/she would be able to become a valued employee in the organization.
- Workplace practices should be observed and followed by employees in an organization.
- Ineffective work management might lead to stress which eventually can lead to anger that can affect your health and career as well.
- Effective work management can help to stay stress free.
- Anger is the byproduct of stress that does harm both professionally and personally.
- Anger should be managed.

## MODULE-2

Computer Applications  
(712)



Notes



### Terminal Exercise

1. What do you understand by the term workplace?
2. List and explain the workplace practices.
3. Highlight the importance of communication in workplace practices and work management?
4. List and explain work management tips.
5. Why is it important to manage anger?
6. Write and explain steps to manage anger.



### Answers to Intext Questions

#### 17.1

1. (a) Punctual      (b) Organization chart.      (c) Concise, effective

#### 17.2

1. (a) False      (b) True      (c) True

### Key Learning Outcome

- Able to maintain a healthy, safe and secure work environment.

Code 711 & 712

*Certificate*  
*in*  
**Computer Applications**  
  
**Practical Manual**



विद्याधनम् सर्वधनं प्रधानम्

**NATIONAL INSTITUTE OF OPEN SCHOOLING**

(An autonomous organisation under MoE, Govt. of India)

A-24-25, Institutional Area, Sector-62, NOIDA-201309 (U.P.)

Website: [www.nios.ac.in](http://www.nios.ac.in), Toll Free No. 18001809393

---

## ADVISORY COMMITTEE

---

**Chairman**  
NIOS, NOIDA

**Director, Vocational**  
NIOS, NOIDA

---

## CURRICULUM COMMITTEE

---

**Prof. Sunil Kumar Khatri**  
Director  
Amity Institute of Information  
Technology, NOIDA

**Dr. G. Mythili**  
Deputy Director  
IGNOU  
New Delhi

**Dr. Athiyaman**  
Scientist – F & Head  
Ministry of Earth Science  
New Delhi

**Ms. Radhika. B**  
Assistant Director (Vocational)  
NIOS, NOIDA

---

## LESSON WRITERS

---

**Ms. Purvi Kumar**  
HOD (Computer Science)  
Ganga International School  
New Delhi

**Ms. Anju Gupta**  
Former HOD Computer Science  
Rukhmini Devi Public School  
New Delhi

---

## EDITORS

---

**Ms. Mohini Arora**  
HOD (Computer Science)  
AirForce Golden Jubilee Institute  
New Delhi

**Ms. Radhika. B**  
Assistant Director (Vocational)  
NIOS, NOIDA

---

## COURSE COORDINATOR

---

**Ms. Radhika B**  
Assistant Director (Vocational)  
NIOS, NOIDA

---

## GRAPHICS/DTP

---

M/S Sri Krishna Graphics  
Delhi

---



# CONTENTS

|                        |    |
|------------------------|----|
| 1. Practical 1 .....   | 1  |
| 2. Practical 2 .....   | 3  |
| 3. Practical 3 .....   | 4  |
| 4. Practical 4 .....   | 6  |
| 5. Practical 5 .....   | 8  |
| 6. Practical 6 .....   | 10 |
| 7. Practical 7 .....   | 12 |
| 8. Practical 8 .....   | 14 |
| 9. Practical 9 .....   | 16 |
| 10. Practical 10 ..... | 18 |
| 11. Practical 11 ..... | 20 |
| 12. Practical 12 ..... | 22 |
| 13. Practical 13 ..... | 24 |
| 14. Practical 14 ..... | 26 |
| 15. Practical 15 ..... | 28 |
| 16. Practical 16 ..... | 30 |
| 17. Practical 17 ..... | 32 |
| 18. Practical 18 ..... | 33 |
| 19. Practical 19 ..... | 34 |
| 20. Practical 20 ..... | 35 |
| 21. Practical 21 ..... | 37 |
| 22. Practical 22 ..... | 39 |
| 23. Practical 23 ..... | 41 |

|     |                    |    |
|-----|--------------------|----|
| 24. | Practical 24 ..... | 42 |
| 25. | Practical 25 ..... | 43 |
| 26. | Practical 26 ..... | 44 |
| 27. | Practical 27 ..... | 46 |
| 28. | Practical 28 ..... | 47 |
| 29. | Practical 29 ..... | 48 |
| 30. | Practical 30 ..... | 49 |
| 31. | Practical 31 ..... | 50 |
| 32. | Practical 32 ..... | 51 |
| 33. | Practical 33 ..... | 53 |
| 34. | Practical 34 ..... | 55 |
| 35. | Practical 35 ..... | 56 |



# Practical 1

Visit NIOS website and in Online Course Material section, find out an e-book of your choice.

## OBJECTIVES

After completing this practical, you will be able to:

- explore the e-content freely available on the website;
- enhance your understanding in the subject with the help of e-content available on the website.

## SOFTWARE REQUIREMENT

- Web Browser
- Adobe Reader XI

## PROCEDURE/ACTIVITY

- 1: Open any web browser.
- 2: In the address bar of the browser, type the following URL of the website and press the Enter key:  
*<https://www.nios.ac.in/>*
- 3: On the Home Page → Click on “Online Course Material” link, present at the bottom of the page.
- 4: Select the required course from the given menu.
- 5: Click on the desired eTextBook link present on the next page.
- 6: Link of all the lessons present under that eTextbook will be displayed on the next page.
- 7: Click on the required lesson. That lesson will be opened up on the screen.



Notes

**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....

**Invigilator's Signature**



Notes

## Practical 2

Create your account on any social networking website and share your ideas with other members on the topic “**Healthy mind resides in a healthy body**” to spread health awareness among them.

### OBJECTIVES

After completing this practical, you will be able to:

- create your account on any social networking website;
- express and share your views with others on any particular topic.

### PROCEDURE/ACTIVITY

1. Create your account on any social networking website by providing the basic information such as first name, last name, age, password etc.
2. Connect with other members by sending them friend’s request.
3. Start the conversation on the given topic and share your ideas on the same.

### LEARNER’S OBSERVATIONS

.....

.....

.....

.....

.....

.....

.....

Invigilator’s Signature



## Practical 3

To spread awareness on the topic “**Save Water**”, create a document in any word processing software, like Writer and enter the following lines:

All the living beings like humans, animals, plants, etc. need **water** to grow, develop and live. **Water** is the only source of all lives here. We need **water** in all walks of life from morning till night like drinking, cooking, bathing, washing clothes, watering plants, etc. With water crisis rampant in many parts of the world due to lack of adequate rainfall, depletion of groundwater and scorching heat, the world is fighting a tough battle of saving water.

Save this recently created document with a suitable name like “**campaign**” for the future reference and close it.

### OBJECTIVES

After completing this practical, you will be able to:

- create a document;
- arrange data in the document;
- save it for future reference;
- close the document for better safety.

### PROCEDURE/ACTIVITY

1. Create a new file in Writer application by following the given steps:
2. Open openoffice.org application →Select “**Text Document**” section.  
A blank document will be created.
3. Write the lines given above in the practical section.

Save the file by following the given steps:



Notes

4. Click on File → Save As

Save As dialog box will open.

5. Enter suitable name like “**campaign**” in File name text field.

6. Click on the Save button.

To close the document, follow any one of the given steps:

7. Click on File → Exit

OR

Press Ctrl + Q

OR

Click on Close (X) button present at the top extreme right corner of the window.

**LEARNER’S OBSERVATIONS**

.....  
.....  
.....  
.....  
.....  
.....  
.....

**Invigilator’s Signature**



## Practical 4

Open the document “**campaign**” and add the title “**SAVE WATER**” at the top of the document as shown below with the following styles:

- Align the title in the center.
- Apply bold, italic and underline style on the title.
- Title should be in 18 font size and red in colour.

### SAVE WATER

All the living beings like humans, animals, plants, etc., need **water** to grow, develop and live. **Water** is the only source of all lives here. We need **water** in all walks of life from morning till night like drinking, cooking, bathing, washing clothes, watering plants, etc.

With water crisis rampant in many parts of the world due to lack of adequate rainfall, depletion of groundwater and scorching heat, the world is fighting a tough battle of saving water.

- Save the document again and close it.

### OBJECTIVES

After completing this practical, you will be able to:

- open an existing document;
- add title/data in the document at required place;
- align the text;
- apply Bold/Italic/Underline style on the text;
- change the font size and color as per the requirement;
- save the document for future reference;
- close the document for better safety.





Notes

**PROCEDURE/ACTIVITY**

1. Open openoffice.org application →Select **“Open a document”** section.  
An Open Dialog box will appear.
2. Select the required document i.e., **“campaign”** from the open dialog box →  
Click on the Open button. Previously created document will be opened up.
3. Bring the mouse pointer in the beginning of the first line and press the Enter  
key so that space can be created in the beginning of the document.
4. Add title **“SAVE WATER”**.
5. Click on **Center Alignment** tab present at Formatting toolbar.
6. Select the text →Click on Bold, Italic, Underline tab present at Formatting  
toolbar.
7. Select the text →Select the desired font size from Font Size box and desired  
color from the Font color box.
8. Save the file by following the given steps:  
Click on File →Save  
Document will be saved.
9. To close the document, follow any one of the given steps:  
Click on File →Exit  
OR  
Press Ctrl + Q  
OR  
Click on **Close (X)** button present at the top extreme right corner of the  
window.

**LEARNER’S OBSERVATIONS**

.....  
.....  
.....  
.....  
.....

**Invigilator’s Signature**



## Practical 5

Create a workbook in Calc to store medal records of Asian games-2018. Enter the records as shown below:

**2018 Asian Games Medal Table**

| Rank ↕ | NOC ↕  | Gold ↕ | Silver ↕ | Bronze ↕ |
|--------|--|--------|----------|----------|
| 1      |  China (CHN)            | 132    | 92       | 65       |
| 2      |  Japan (JPN)            | 75     | 56       | 74       |
| 3      |  South Korea (KOR)     | 49     | 58       | 70       |
| 4      |  Indonesia (INA)*     | 31     | 24       | 43       |
| 5      |  Uzbekistan (UZB)     | 21     | 24       | 25       |
| 6      |  Iran (IRI)           | 20     | 20       | 22       |
| 7      |  Chinese Taipei (TPE) | 17     | 19       | 31       |
| 8      |  India (IND)          | 15     | 24       | 30       |
| 9      |  Kazakhstan (KAZ)     | 15     | 17       | 44       |
| 10     |  North Korea (PRK)    | 12     | 12       | 13       |

Save it with any suitable name like “Asiad-2018” and close it.

### OBJECTIVES

After completing this practical, you will be able to:

- create a workbook in calc;
- enter the data in the workbook;
- save it for future reference;
- close the workbook for better safety.



Notes

**PROCEDURE/ACTIVITY**

1. Create a new workbook in Calc application by following the given steps:
2. Open openoffice.org application →Select “**Spreadsheet**” section.  
A blank spreadsheet file will be created.
3. Enter the records as given above in the practical section.
4. Save the workbook by following the given steps:  
Click on File →**Save/Save As/Save All**  
**Save As** dialog box will open.
5. Enter suitable name like “**Asiad-2018**” in File name text field.
6. Click on the **Save** button.  
To close the workbook, follow any one of the given steps:
7. Click on **File →Exit**  
OR  
Press Ctrl + Q  
OR  
Click on **Close (X)** button present at the top extreme right corner of the window.

**LEARNER’S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

**Invigilator’s Signature**



## Practical 6

Open the workbook created in Practical 5 and apply the following styles:

- Apply bold, italic and underline style on the heading.
- Header row should be 18 in font size and red in colour.

Save the workbook again and close it.

### OBJECTIVES

After completing this practical, you will be able to:

- open an existing workbook;
- apply Bold/Italic/Underline style on the heading;
- change the font size and color of the text as per the requirement;
- save the workbook for future reference;
- close the workbook for better safety.

### PROCEDURE/ACTIVITY

1. Open openoffice.org application →Select “**Open a document**” section.  
An Open Dialog box will appear.
2. Select the required workbook i.e., “**Asiad-2018**” from the open dialog box. Click on the Open button. Previously created workbook will be opened up.
3. Select the text →Click on Bold, Italic, Underline tab present at Formatting toolbar.
4. Select the text →Select the desired font size from Font Size box and desired color from the Font color box. Save the workbook by following the given steps:



**Notes**

5. Click on File → **Save**. Workbook will be saved.

To close the workbook, follow any one of the given steps:

6. Click on **File** → **Exit**

OR

Press Ctrl+Q

OR

Click on **Close (X)** button present at the top extreme right corner of the window.

**LEARNER'S OBSERVATIONS**

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

**Invigilator's Signature**



## Practical 7

Open “Asiad-2018” workbook created in Practical-5 and apply suitable formula in front of the record to calculate total medals won by any country. Save the workbook again and close it.

### OBJECTIVES

After completing this practical, you will be able to:

- open an existing workbook;
- add suitable formula in required cell to perform desired task;
- save the workbook for future reference;
- close the workbook for better safety.

### PROCEDURE/ACTIVITY

1. Open openoffice.org application →Select “**Open a document**” section.  
Open Dialog box will appear.
2. Select the required workbook i.e., “**Asiad-2018**” from the open dialog box. Click on the Open button. Previously created workbook will be opened up.
3. Select the cell →Type the required formula.
4. Press the Enter key. Total of all three values of first record will be displayed.
5. Bring the mouse pointer at lower bottom corner of the cell. Mouse pointer will be converted into ‘+’ sign, which is called fill handle.



**Notes**

6. Drag the fill handle down to fill the cells with required formula/value. Save the file by following the given steps:

7. Click on **File →Save**. Workbook will be saved.

To close the workbook, follow any one of the given steps:

6. Click on **File →Exit**

OR

Press Ctrl+Q

OR

Click on Close (X) button present at the top extreme right corner of the window.

**LEARNER'S OBSERVATIONS**

.....  
.....  
.....  
.....  
.....  
.....  
.....

**Invigilator's Signature**



## Practical 8

Create a new presentation through template option in Impress on the theme “**Save Earth**”, having at least three slides. Save it with any suitable name like “**Mother Earth**” and close it.

### OBJECTIVES

After completing this practical, you will be able to:

- create a presentation through template in impress.
- add one or more slides as per the requirement;
- save it for future reference;
- close the presentation for better safety.

### PROCEDURE/ACTIVITY

1. Create a new presentation in Impress application by following the given steps:
2. Open openoffice.org application → Select “**Presentation**” section. Presentation Wizard will be opened up, from which select “**From template**” option.
3. Select the desired template → Click on “**Next >>**” button. Next window of Presentation Wizard will be opened up.
4. Select suitable slide design → Click on the **Create** button. A presentation with single slide will be created.
5. Add the required content on the first slide.
6. To add more slides, click on Insert → Slide → Add the content on the slide.





7. Save the presentation by following the given steps:  
Click on **File→Save/Save As/Save All**  
Save As dialog box will open.
5. Enter suitable name like “**Mother Earth**” in File name text field.
6. Click on the Save button. To close the presentation, follow any one of the given steps:
7. Click on **File→Exit**  
OR  
Press Ctrl + Q  
OR  
Click on Close (X) button present at the top extreme right corner of the window.

**LEARNER’S OBSERVATIONS**

.....  
.....  
.....  
.....  
.....  
.....  
.....

**Invigilator’s Signature**



## Practical 9

Create a new presentation through “**Empty presentation**” option in Impress on the theme “**Save Earth**”, having at least three slides. Save it with any suitable name like “**My Mother Earth**” and close it.

### OBJECTIVES

After completing this practical, you will be able to:

- create a presentation through “**Empty presentation**” option in impress.
- add one or more slides as per the requirement;
- save it for future reference;
- close the presentation for better safety.

### PROCEDURE/ACTIVITY

Create a new presentation through “**Empty presentation**” option in Impress application by following the given steps:

1. Open openoffice.org application →Select “**Presentation**” section.  
Presentation Wizard will be opened up, from which select “**Empty presentation**” option →Click on “**Next>>**” button.
2. Select suitable slide design →Click on the **Create** button. A presentation with single slide will be created.
3. Add the required content on the first slide.
4. To add more slides, click on **Insert →Slide**. Add the content on the slide(s).
5. Save the presentation by following the given steps:  
Click on **File →Save/Save As/Save All**. Save As dialog box will open.
6. Enter suitable name like “**My Mother Earth**” in File name text field.



**Notes**

7. Click on the **Save** button.

To close the presentation, follow any one of the given steps:

8. Click on **File–Exit**

OR

Press Ctrl + Q

OR

Click on **Close (X)** button present at the top extreme right corner of the window.

**LEARNER’S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

**Invigilator’s Signature**



## Practical 10

Open the previously created presentation in Practical 9, apply suitable transition effects and close it after saving it again.

### OBJECTIVES

After completing this practical, you will be able to:

- open the previously created presentation in Impress;
- apply transition effects;
- save it for future reference;
- close the presentation for better safety.

### PROCEDURE/ACTIVITY

1. Open a previously created presentation in Impress application:

Open openoffice.org application → Select “**Presentation**” section. Presentation Wizard will be opened up, from which select “**Open existing presentation**” option → Select required previously created Presentation → Click on the Open button. Previously created presentation will be opened up.

2. Click on **Slide Show** → **Slide Transition**.

Slide Transition pane will appear at the right hand side of the window.

3. Select desired transition effects and settings Click on “**Apply to All Slides**” option. Transition effects will be applied successfully.

4. Save the presentation by following the given steps:

Click on **File** → **Save/Save All**

Presentation will be saved automatically with the previous name. To save presentation with a new name, Save As option can be used.



**Notes**

5. To close the presentation, follow any one of the given steps:

Click on **File-Exit**

OR

Press Ctrl + Q

OR

Click on **Close (X)** button present at the top extreme right corner of the window.

**LEARNER'S OBSERVATIONS**

.....  
.....  
.....  
.....  
.....  
.....  
.....

**Invigilator's Signature**



## Practical 11

Create a table named “STUDENT” in MySQL having following structure:

| Column Name | Data Type (Size) |
|-------------|------------------|
| RegNo       | INT (11)         |
| Name        | VARCHAR(20)      |
| ExamName    | VARCHAR(20)      |

### OBJECTIVES

After completing this practical, you will be able to:

- create a table in MySQL;
- use basic data types like int, varchar etc., effectively.

### PROCEDURE/ACTIVITY

1. Open MySQL Command Line Editor.
2. On the command prompt, give following SQL command to open the RESULT database in which table needs to be created:

```
USE <DataBase Name>
```

If database is not created already, then first use the following command to create RESULT database:

```
CREATE DATABASE <DataBase Name>
```

3. Using CREATE TABLE, create the STUDENT table.

**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....



**Notes**

**Invigilator's Signature**



## Practical 12

Create a table named “EMPLOYEE” in MySQL having following structure:

| Column Name | Data Type (Size) | Constraint |
|-------------|------------------|------------|
| EmpNo       | INT (11)         | UNIQUE     |
| Name        | VARCHAR(20)      | NOT NULL   |
| City        | VARCHAR(20)      | NOT NULL   |
| MobileNo    | INT (10)         | NOT NULL   |
| JoinDate    | DATE             |            |
| Salary      | INT (6)          |            |

### OBJECTIVES

After completing this practical, you will be able to:

- create a table in MySQL;
- use various data types effectively;
- put constraints on the columns.

### PROCEDURE/ACTIVITY

1. Open MySQL Command Line Editor.
2. On the command prompt, give following SQL command to open office database in which table needs to be created:

```
USE <DataBase Name>
```

If database is not created already, then first use the following command to create database:

```
CREATE DATABASE <DataBase Name>
```

3. Give the suitable SQL command to create table along with required constraints.



**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....



**Notes**

**Invigilator's Signature**



## Practical 13

Insert following records in the previously created EMPLOYEE table:

| EmpNo | Name     | City   | MobileNo   | JoinDate    | Salary |
|-------|----------|--------|------------|-------------|--------|
| 1001  | Rajan    | Delhi  | 9711516618 | 12-Feb-2017 | 10000  |
| 1002  | Sania    | Mumbai | 9101827363 | 03-Jan-2018 | 20000  |
| 1003  | John     | Delhi  | 8373544332 | 01-Apr-2019 | 25000  |
| 1004  | Harpreet | Jaipur | 7325343433 | 12-Feb-2016 | 30000  |
| 1005  | Rozy     | Surat  | 8310304527 | 03-Jan-2017 | 20000  |

### OBJECTIVES

After completing this practical, you will be able to:

- insert records in a table in MySQL.

### PROCEDURE/ACTIVITY

1. Open MySQL Command Line Editor.
2. On the command prompt, give following SQL command to open office database in which table structure was created:

USE <DataBase Name>

3. Give the suitable SQL command to insert records in the table.

Use insert command to insert records in the table

**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....



**Notes**

**Invigilator's Signature**



## Practical 14

Write the suitable query to perform the following tasks in EMPLOYEE table of office database:

- Display all the records.
- Display Employee's name and their respective City name.
- Display all the employee's detail whose name starts with 'R' alphabet.
- Display all the employee's details who has joined after 2017.
- Display all the employee's details whose belongs to Delhi city.

### OBJECTIVES

After completing this practical, you will be able to:

- perform variety of queries on the table.
- display a variety of reports based on various conditions.

### PROCEDURE/ACTIVITY

1. Open MySQL Command Line Editor.
2. On the command prompt, give following SQL command to open the database in which table structure was created:

USE <DataBase Name>

3. Write suitable SQL select statements to display the records from employee table.



**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....

**Notes**

**Invigilator's Signature**



## Practical 15

Write the suitable query to perform the following tasks using EMPLOYEE table of office database:

- Increase the salary of all the employees by Rs. 1000.
- Display Employee's name and their annual salary.
- Delete the record of those employees who belong to Delhi.

### OBJECTIVES

After completing this practical, you will be able to:

- perform variety of queries on the table;
- display a variety of reports based on various conditions.

### PROCEDURE/ACTIVITY

1. Open MySQL Command Line Editor.
2. On the command prompt, give following SQL command to open the database in which table structure was created:

USE <DataBase Name>

3. (a) Using UPDATE command to increase the salary of employees.  
(b) Using SELECT command to display the desired information of employees.  
(c) Use DELETE command to delete the records of employee belonging to Delhi.

**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....



**Notes**

**Invigilator's Signature**



## Practical 16

Evaluate the following expression(s) and find out the result and data type of result.

(i)  $\text{len}(\text{"Hello"}) = 25/5$  or  $20/10$

.....

(ii)  $3 < 5$  or  $50/(5 - (3 + 2))$

.....

(iii)  $2 * (2 * (\text{len}(\text{"01"})))$

.....

### OBJECTIVES

After completing this practical you will be able to:

- create expression;
- identify precedence of operation performed, in an expression having different type of operators;
- type expressions in Python IDLE.

### PROCEDURE/ACTIVITIES

- Open IDLE and type the expression to get result.
- Operator precedence section may be referred before working on operators.
- Use `type()` function to find out data type.
- Note and analyze your observation.





**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....

**Notes**

**Invigilator's Signature**



Notes

## Practical 17

Write a Python statement(s), to check, if the given number is even or odd.

### OBJECTIVES

After completing this practical, you will be able to:

- accept data from keyboard;
- use if statement and display appropriate message.

### PROCEDURE/ACTIVITIES

1. Open Python IDLE.
2. Use `input()` to accept data.
3. Convert it to integer using `int()`.
4. Check whether the given number is even or odd by using divisibility rule for number 2.
5. Use if statement to display appropriate message.

### LEARNER'S OBSERVATIONS

.....

.....

.....

.....

.....

.....

.....

Invigilator's Signature



Notes

# Practical 18

Write a Python statement(s) to convert given time in seconds to hours, minutes and seconds.

## OBJECTIVES

After completing this practical, you will be able to:

- apply arithmetic operators, to form expression;
- use correct operator to find quotient and remainder of integer division.

## PROCEDURE/ACTIVITIES

- Open IDLE.
- Using input() and int() accept an integer number (i.e., time in seconds).
- Divide number by 3600 to obtain quotient (which is number of hrs).
  - Divide the number by 3600 to get number of hours in the given seconds.
  - Calculate remainder of number/3600 and store in variable minutes.
  - Divide minutes by 60 to get minutes  
Calculate remainder of minutes/60 to get remaining seconds.

## LEARNER'S OBSERVATIONS

.....

.....

.....

.....

.....

.....

Invigilator's Signature



Notes

## Practical 19

Write a Python script, to accept a character. If it is in lowercase then convert it to uppercase and vice versa. Display the converted character.

### OBJECTIVES

After completing this practical, you will be able to:

- accept character input;
- apply string method / functions to get the desired output.

### PROCEDURE/ACTIVITIES

1. Open a new file to write the code.
2. Accept a character using `input()`.
3. Check if character is uppercase using `isupper()` method.
4. Check if character is lowercase using `islower()` method.
5. Use `upper()` method to convert it into uppercase.
6. Use `lower()` method to convert it into lowercase.
7. Use print statement to display the character
8. Save the code using file → Save As option

### LEARNER'S OBSERVATIONS

.....

.....

.....

.....

.....

Invigilator's Signature



## Practical 20

Write a Python script, to accept a sentence and count number of vowels appearing in it.

### OBJECTIVES

After completing this practical, you will be able to:

- accept a sentence from user using `input()`;
- compare characters using relational operator;
- traverse string.

### PROCEDURE/ACTIVITIES

1. Open a new file to write the code.
2. Accept a sentence using `input()`.
3. Define a variable `count` to value 0, for counting the number of vowels in sentence.
4. Using for loop, traverse the sentence, character wise.
5. Convert the character to any one case lowercase or uppercase.
6. Check existence of vowel ('aeiou' / 'AEIOU') using `in` operator.
7. Add 1 to count whenever it returns True.
8. Display count.
9. Save the code in "Vow count" using file →Save As option.



Notes

**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....

**Invigilator's Signature**



## Practical 21

Write a script that will ask the user to enter N integers. Find out the following:

- (i) Total number of positive integers.
- (ii) Total number of negative integers.

### OBJECTIVES

After completing this practical, you will be able to:

- accept multiple numbers using loop;
- check and distinguish between positive and negative numbers in a program.

### PROCEDURE/ACTIVITIES

1. Open a new file to write the code.
2. Define two variables NoPos and NoNeg to value 0.
3. Accept value for variable N.
4. Using for loop to accept values and check and for +ve and \_ve numbers.
5. Compare the number with 0 to check for +ve or -ve value.
6. Increment NoPos for positive value and Noneg for negative value.
7. Inside loop use input() with int() to accept next number.
8. Outside the loop display NoPos and NoNeg with appropriate message.
9. Save the code using file →Save As option.



Notes

**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....

**Invigilator's Signature**





## Practical 22

Write a Python script, to accept a list of names and a name . Check whether the name is present in the list of names or not. If a name is present in the list of names then the script should display the position of the name in a list. Otherwise it displays FALSE.

### OBJECTIVES

After completing this practical, you will be able to:

- create a list by accepting string values from user;
- apply search (linear) on a list of values;
- return appropriate value to user.

### PROCEDURE/ACTIVITIES

1. Accept list of names using eval() function.
2. Accept a name using input() function.
3. Find out the length of list for traversing it.
4. Using while loop, traverse the list, for comparing name with each element of list.
5. As soon as the name is found in the list, display the value of list index + 1 and stop the loop.
6. Display FALSE, if you reach end of loop and no match is found.



Notes

**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....

**Invigilator's Signature**



Notes

# Practical 23

Write a Python script, to accept a list of numbers and then display it in reverse order.

## OBJECTIVES

After completing this practical, you will be able to:

- create a list of numbers, by accepting values from user.
- apply list slicing to reverse a list.

## PROCEDURE/ACTIVITIES

1. Accept a list of numbers using eval() function.
2. Use slicing without start and stop value and with step -1 in print statement, to reverse the content of list for displaying only.
3. Display the result.

## LEARNER'S OBSERVATIONS

.....

.....

.....

.....

.....

.....

.....

Invigilator's Signature



Notes

## Practical 24

Write a Python script, to accept a list of numbers and then display all those elements, which are divisible by 3 and 5.

### OBJECTIVES

After completing this practical, you will be able to:

- apply indexing, to access individual element of list;
- use loop to traverse the list;

### PROCEDURE/ACTIVITIES

1. Accept a list of numbers using eval() function.
2. Find out length of list using len() function.
3. Use for loop to traverse the list.
4. While traversing, check divisibility of element by 3 and 5.
5. If the element is divisible by both, then display it using print.

### LEARNER'S OBSERVATIONS

.....

.....

.....

.....

.....

.....

Invigilator's Signature



Notes

## Practical 25

Write Python script to accept a list of numbers and then display sum of elements placed at even index value.

### OBJECTIVES

After completing this practical, you will be able to:

- accept a list of numbers, from user;
- apply len() function to work on list;
- traverse a list using range() function with different step value.

### PROCEDURE/ACTIVITIES

1. Using eval() function accept the list.
2. Initialize a variable (sum) by value 0, to store sum in it.
3. Find length of the list using len() function.
4. Traverse the list, using for loop with step as 2.
5. Keep on adding the element in variable (sum), while traversing the list.
6. Display value of the variable (sum).

### LEARNER'S OBSERVATIONS

.....

.....

.....

.....

.....

.....

Invigilator's Signature



## Practical 26

Create a text file named “**myTextfile1.txt**” by storing predefined content in it.

### OBJECTIVES

After completing this practical you will be able to:

- open a text file for writing in it;
- add EOL in a string constant;
- write a string in the file;
- close a file.

### PROCEDURE/ACTIVITIES

1. Create a file object for writing in it. File is to be saved in current working directory.
2. Write following content in file using write lines, retain the format by using EOL:

Just at dawn,  
When everyone's asleep,  
And you can hear the wind,  
That the trees cast at you,

3. Close file.

**LEARNER'S OBSERVATIONS**

.....

.....

.....

.....

.....

.....

.....

.....



**Notes**

**Invigilator's Signature**



Notes

# Practical 27

Open a text file – “mytextfile.txt1” for reading and displaying content on monitor.

### OBJECTIVES:

After completing this practical you will be able to:

- open a text file for reading from it;
- use a loop to read data from file;
- read a string, terminated by EOL (End Of Line), from the file.
- close a file.

### PROCEDURE/ACTIVITIES:

1. Create file object to read from the file.

#### Syntax:

```
file object=open(“filename”,mode)
```

2. Read file content.
3. Using loop to read a line from the file using readline.
4. Display them on screen.
5. Close file.

### LEARNER’S OBSERVATIONS

.....

.....

.....

.....

.....

**Invigilator’s Signature**





Notes

# Practical 28

Write a Python program to accept input (string) from the user and store it in – “myTextfile2.txt”. Add statement(s) to read content from this file and display on screen.

## OBJECTIVES

After completing this practical you will be able to:

- open a text file for writing in it;
- add EOL in a string constant;
- close a file.

## PROCEDURE/ACTIVITIES

1. Create file object to write and read from the file.
2. Use input() to accept data (i.e., string) at run time.
3. Write it in file using write line. Do not forget to add EOL whenever required.
4. Use loop to add multiple strings in the file.
5. Bring file handle to the beginning of the file, for reading from it.
6. Read content from the file using readline, use loop to read the complete file.
7. Close file.

## LEARNER’S OBSERVATIONS

.....

.....

.....

.....

.....

Invigilator’s Signature



Notes

# Practical 29

Write a Python program to create a binary file “**myBinaryfile.dat**”, for storing Student’s data such as Name, Enrolment number, Marks obtained in 4 subjects / papers.

## OBJECTIVES

After completing this practical, you will be able to:

- use pickle module to work with binary file;
- open a binary file for writing in it;
- use dictionary data type to store record of students in the program;
- write data in binary file;
- close a file.

## PROCEDURE/ACTIVITIES

1. Use dictionary to accept above mentioned data fields of 5 students. Enrolment number can be used as key.
2. Import pickle module.
3. Create a file object to write in binary file.
4. Using **dump()** to write students data (i.e. dictionary) in the file.
5. Close file.

## LEARNER’S OBSERVATIONS

.....

.....

.....

.....

.....

Invigilator’s Signature



Notes

# Practical 30

Write a Python program to read data from “**myBinaryfile.dat**” (binary file) and display Student’s data on screen.

## OBJECTIVES

After completing this practical you will be able to

- use pickle module in a program;
- open a binary file for reading from it;
- read record of students from binary file, and display them on screen;
- close a file.

## PROCEDURE/ACTIVITIES

1. Import pickle module.
2. Create file object to read data from binary file.
3. Using load() method read file content in a dictionary data type.
4. Close the file.
5. Display file content on screen by printing dictionary.

## LEARNER’S OBSERVATIONS

.....

.....

.....

.....

.....

.....

Invigilator’s Signature



Notes

# Practical 31

Using data stored in “myBinaryfile.dat”, calculate total and percentage of marks obtained by each student. Display Name, Enrolment number, total marks, percentage for each student.

## OBJECTIVES

After completing this practical you will be able to:

- open a binary file for reading from it;
- read records of students from the file;
- manipulate data read from the file;
- close a file.

## PROCEDURE/ACTIVITIES

1. Import pickle module.
2. Create a file object to read binary file.
3. Using **load()** method read file content in a dictionary.
4. Close the file.
5. Using dictionary manipulation, calculate total & percentage of each student and display all other information also.

## LEARNER’S OBSERVATIONS

.....

.....

.....

.....

.....

Invigilator’s Signature



Notes

## Practical 32

A list L is used to implement stack, show the status (created using L) of stack after following operations are performed on it.

Push(53.5)

Display()

.....  
.....

Pop()

Display()

.....  
.....

Pop()

Display()

.....  
.....

Push(21.79)

Display()

.....  
.....

### OBJECTIVE

After completing this practical you will be able to use list to implement stack.



Notes

**PROCEDURE/ACTIVITIES**

1. Use function written in book to complete practical. For showing status of stack, keep on writing the element(s) of stack or the message displayed while performing operation.
2. Perform first operation by tracing Push operation statement.
3. Trace pop operation statement to perform next two operations.
4. Trace display statement for performing last operation.

**LEARNER'S OBSERVATIONS**

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

**Invigilator's Signature**



Notes

## Practical 33

Given the content of stack (created using list), write down the sequence of operation performed on the stack. Stack top is on right side of text

Stack : 65

.....  
.....

Stack : 65, 90

.....  
.....

Stack : 65

.....  
.....

Stack : 65, 98, 43

.....  
.....

Stack : 65, 98

.....  
.....

Stack : 65

.....  
.....

Stack :

.....  
.....



Notes

Stack :

.....  
.....

**OBJECTIVE**

After completing this practical you will be able to use list to implement stack..

**LEARNER'S OBSERVATIONS**

.....  
.....  
.....  
.....  
.....  
.....  
.....

**Invigilator's Signature**





# Practical 34

Write a Python function, to accept a positive integer(N) as parameter and return sum of all the numbers lying between 1 and N .

## OBJECTIVES

After completing this practical, you will be able to:

- learn and apply modular approach to programming;
- write function with parameters, for accepting value(s);
- calculate and return value(s) from function.

## PROCEDURE/ACTIVITIES

1. Open a new file to write the function.
2. Accept N as parameter in the function.
3. Define variable sum to 0.
4. Use a loop (for) to repeat N times.
5. Inside loop add loop variable to sum.
6. Return value of sum, once loop execution finishes.
7. Save your function in file “Add Name” by using option file →Save As

## LEARNER’S OBSERVATIONS

.....  
.....  
.....  
.....  
.....  
.....

Invigilator’s Signature



Notes

## Practical 35

Write a Python function `calc_pow(num1,num2)`, to calculate and display, cube of all the numbers starting from num1 to num2.

### OBJECTIVES

After completing this practical, you will be able to:

- apply modular approach to solve the problem;
- learn to test code in small fragment(s);
- apply exponentiation operator.

### PROCEDURE/ACTIVITIES

1. Open a new file to write the function.
2. Accept two parameters in function header.
3. Use for loop from num1 to num2+1.
4. Calculate cube of each value using `**` operator.
5. Display result inside loop.
6. Save your code using file →Save As option in “Funct 1”

### LEARNER’S OBSERVATIONS

.....  
.....  
.....  
.....  
.....  
.....

**Invigilator’s Signature**