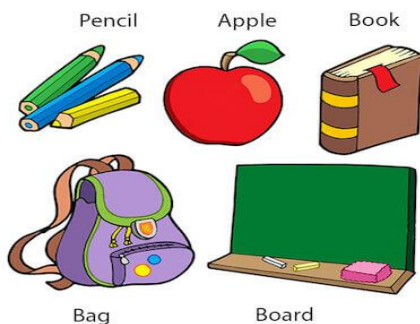


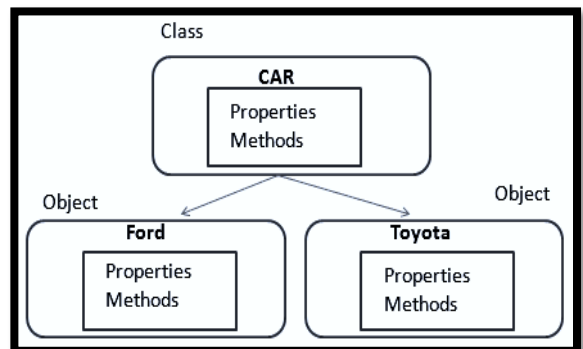
CLASSES AND OBJECTS WITH CONSTRUCTORS/ DESTRUCTORS

- **CLASS:** A class in C++ is the building block, which leads to Object-Oriented programming.
- It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.
- A C++ class is like a blueprint for an object.
- Data members are the data variables and member functions are the functions.
- Data members and member functions defines the properties and behavior of the objects in a Class.
- For Example: Let's Consider the Class of **Cars**. There may be many cars with different names and brand but all of them will share some common properties like they will have 4 wheels, Speed Limit, Mileage range etc.
- **OBJECT:** An **Object** is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

Objects: Real World Examples



- **RELATIONSHIP BETWEEN A CLASS AND OBJECT:** Let's take an example of class Car.



In the above example, for the class Car, there are 2 objects created named Ford & Toyota which have some properties and methods.

DEFINING CLASS AND DECLARING OBJECTS

A class is defined in C++ using keyword class followed by the name of class. The body of class is defined inside the curly brackets and terminated by a semicolon at the end

```

keyword      user-defined name
↓            ↓
class ClassName
{ Access specifier: //can be private,public or protected
  Data members;    // Variables to be used
  MemberFunctions() {} //Methods to access data members
};                // Class name ends with a semicolon
    
```

- By default, the members of a class are private.
- Private data members and private functions can be accessed only by member functions of a class.
- Public members can be accessed from outside of the class.

- **DECLARING OBJECTS:** When a class is defined, no memory or storage is allocated. To use the data and access functions defined in the class, you need to create objects.

Syntax:

ClassName ObjectName;

- **ACCESSING DATA MEMBERS AND MEMBER FUNCTIONS:**

ObjectName.MemberFunction ();

- **DEFINING MEMBER FUNCTION:** Member function can be defined in two ways:

- Inside the class
- Outside the class.

Inside Class: When a member function is defined inside a class, it is considered to be **inline** by default.

Outside Class: When a function is large then it should be defined outside the class declaration. The operator ‘:.’ is known as scope resolution operator and is used to associate member functions to their corresponding class.

Syntax: return_type Class_Name::
function_Name;

Example of a program to get and displaying student data:

```
# include < iostream.h >
```

```
class student {
```

```
private :
```

```
char name [ 80 ];
```

```
int rn ;
```

```
float marks ;
```

```
private :
```

```
void getdata ( ) ;
```

```
void putdata ( ) ;
```

```
};
```

```
void student :: getdata ( )
```

```
{ cin >> name >>rn >> marks ;
```

```
}
```

```
void student :: putdata ( ) {
```

```
cout << name << rn << marks ;
```

```
}
```

```
void main ( ) {
```

```
student st ;
```

```
st.getdata ( ) ;
```

```
st.putdata ( ) ;
```

```
}
```

- **NESTING OF MEMBER FUNCTION:** A member function can be called by using its name inside another member function of the same class. This is known as nesting of member functions.

- **ARRAY OF OBJECTS:**

```
class emp {
```

```
char name [ 30 ] ;
```

```
int empno ; public :
```

```
void getdata ( ) ;
```

```
void putdata ( ) ;
```

```
};
```

```
void main ( )
```

```
{
```

```
emp e [ 10 ] ; //Array of Objects
```

```
for ( i = 0; i < 10; i ++)
```

```
e [ i ]. getdata ( ) ;
```

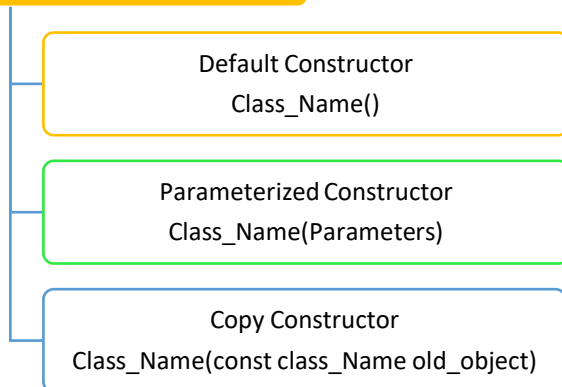
```
for ( i = 0; i < 10; i ++ )
```

```
e [ i ].putdata ( ) ;
```

```
}
```

- **CONSTRUCTOR:** A constructor is a special type of member function of a class which initializes objects of a class.
- Constructor has same name as the class .
- Constructors don't have return type.
- A constructor is automatically called when an object is created.
- It must be placed in public section of class.
- If we do not specify a constructor, C++ compiler generates a default constructor for object (expects no parameters and has an empty body).

Constructor



Default Constructor: Default constructor is the constructor which doesn't take any argument. It has no parameters.

```
#include <iostream>
using namespace std;
```

```
class construct{
public:
    int a, b;

    // Default Constructor
    construct()
    {
        a = 10;
        b = 20;
    }
};
```

```
int main()
```

```
{
// Default constructor called automatically
// when the object is created
construct c;
cout << "a: " << c.a << endl
    << "b: " << c.b;
return 1;
}
```

Output: a: 10

B: 20

Parameterized Constructor: It is possible to pass arguments to constructors. Typically, these arguments help initialize an object when it is created.

```
#include <iostream>
using namespace std;
```

```
class Point
{
private:
    int x, y;

public:
    Point(int x1, int y1) // Parameterized
    Constructor {
        x = x1;
        y = y1;
    }
}
```

```
int getX(){
    return x;
}
int getY()
{
    return y;
}
};
```

```
int main()
{
// Constructor called
Point p1(10, 15);

// Access values assigned by constructor
cout << "p1.x = " << p1.getX() << ", p1.y
= " << p1.getY();

return 0;
}
```

Output:

p1.x = 10, p1.y = 15

COPY CONSTRUCTOR: A copy constructor is a member function which initializes an object using another object of the same class.

- **DESTRUCTOR:** It is used to destroy the objects that have been created by a constructor.
- The destructor is a member function whose name is the same as the class name.
- Syntax- `~class_Name () ;`
- It never takes any argument nor does it return any value.

CHECK YOURSELF

1. Data members and member functions of a class in C++ program are by default:

- A) Protected C) Public
B) Private D) None

2. Which is used to define the member function of a class externally?

- A) : C) ::
B) # D) None

3. When you create an object of a class A like A obj ; then which one will be called automatically

- A) Constructor C) Copy constructor
B) Destructor D) Dot operator

4. Which of the following is a valid class declaration?

- A) Class A { int x; }; C) public class A{ }
B. Class B { } D) object A{int x;}

5. Constructor is executed when _____.

- A. An object goes out of scope.
B. A class is declared
C. An object is created
D. An object is used

STRETCH YOURSELF

1. Differentiate between structure and class?
2. Briefly explain about class and object by giving a suitable example.
3. Define constructor and types of constructors with example?

ANSWERS

Answers to Check Yourself:

1. B
2. C
3. A
4. A
5. C