Class

Class

A class is a logical entity used to define a new data type. A class is a user-defined type that describes what a particular kind of object will look like. A class contains variables(data members), methods, and constructors.

Class is a blueprint or a set of instructions to build a specific type of object. It is a fundamental concept of Object-Oriented Programming which revolves around real-life entities. Class determines how an object will behave and what the object will contain.

Data encapsulation is supported with “class”. The class consists of both data and functions. The data in a class is called a member, while functions in the class are called methods.

Data Members:- The variables which are declared in any class by using any fundamental data types (like int, char, float, etc.) or derived data types (like class, structure, pointer, etc.) are known as Data Members.

Methods:- A method is the equivalent of a function in object-oriented programming that is used inside classes. The methods are the actions that perform operations. A method accepts parameters as arguments, manipulates these, and then produces an output when the method is called on an object.

Constructor:- Constructors are special class functions that perform the initialization of every object. In C++, the constructor is automatically called when an object is created. It is a special method of the class because it does not have any return type. It has the same name as the class itself.

Syntax to define a class:-

```
class class_name{
        //class body
```

```
               //data_members
               //constructor (optional)
               //methods
};
```

Here,

- ❖ class: class keyword is used to create a class in C++.
- ❖ class_name: The name of the class.
- ❖ class body: Curly braces surround the class body.
- ❖ After closing curly braces, a semicolon(;) is used.

Classes are very useful in programming. Consider the example of where you don't want to use just one Smartphone but 100 smartphones. Rather than describing each one in detail from scratch, you can use the same smartphone class to create 100 objects of the type ‘smartphones’. You still have to give each one a name and other properties, but the basic structure of what a smartphone looks like is the same.

The smartphone class would allow the programmer to store similar information unique to each car (different models, maybe different colors, etc.) and associate the appropriate information with each smartphone.

Example of smartphone class:-

```
class smartphone{
  //class body

    //Data Members(Properties)
    string model;
    int year_of_manufacture;
    bool _5g_supported;

    //Constructor
    smartphone(string mod, int manu, bool _5g_supp){
      //initialzing data members
      model = mod;
      year_of_manufacture = manu;
      _5g_supported = _5g_supp;
    }

    //methods
```

```
        void print_details(){
            cout << "Model : " << model << endl;
            cout << "Year of Manufacture : " << year_of_manufacture << endl;
            cout << "5g Supported : " << _5g_supported << endl;
        }
};
```