C++ Classes and Objects

Introduction

C++ is an object-oriented programming language where classes and objects are fundamental components. Classes, the building blocks of C++, are user-defined data types that serve as blueprints for creating objects. Each class contains its own data members and member functions, which can be accessed through instances of that class. This concept is central to understanding C++ Classes and Objects.

For example: Consider a class of students. There can be many students, but all of them will share some common properties like all of them will have a name, age, marks, roll no. So a student is a class having name, age, marks, roll no as its properties.

C++ Classes

C++ classes are user-defined data types that serve as a blueprint for creating objects. It consists of data members, which are data variables, and member functions, which are functions used to manipulate these variables. Together, data members and member functions define the behavior and properties of the objects created from the class.
For example, in a class named Student, the data members might include name, roll_no, age, and marks. The member functions could include updateMarks(), getAge(), and so on.

C++ Object

C++ objects are any real-world entity, such as a chair, pencil, or eraser. It is defined as an entity that has both state and behavior and is an instance of a class. When a class is defined, no memory is initially allocated. However, memory is allocated when an object of that class is created.
For example: if two objects naming student-1 and student-2 of Student class are created, then both of them will have separate memory allocated to them, both of them will have all the properties of the Student class.

Creating a Class and Declaring Objects

In order to define a class in C++, the ‘class’ keyword must be used, followed by the className. The data members and member functions which constitute the body of the class are defined inside curly brackets. A class is always terminated by a semicolon at the end.

```
class className
{
    Access specifier // can be private, protected, public
    Data members  // variables
    Member Functions() // methods to access data members
}; // Class ends with a semicolon
```

Size of an empty class is non -zero(1 byte) in C++.

Declaring Objects

When a class is defined, only the specifications or the blueprint for the object is defined, no memory is allocated. In order to use the data and access the functions defined inside a class, we need an object. In order to access the data members and member functions of a class, the dot (‘.’) operator is used along with the object name. For example, if student1 is the object name and we want to access the getMarks() member function, we will write student1.getMarks() .

Syntax

```
className objectName;
```