C++ Classes and Objects

In this tutorial, we will learn about objects and classes and how to use them in C++ with the help of examples. In previous tutorials, we learned about functions and variables. Sometimes it's desirable to put related functions and data in one place so that it's logical and easier to work with.

Suppose, we need to store the length, breadth, and height of a rectangular room and calculate its area and volume.

To handle this task, we can create three variables, say, length, breadth, and height along with the functions calculateArea() and calculateVolume().

However, in C++, rather than creating separate variables and functions, we can also wrap these related data and functions in a single place (by creating objects). This programming paradigm is known as object-oriented programming.

But before we can create objects and use them in C++, we first need to learn about classes.

C++ Class

A class is a blueprint for the object.

We can think of a class as a sketch (prototype) of a house. It contains all the details about the floors, doors, windows, etc. Based on these descriptions we build the house. House is the object.

Create a Class

A class is defined in C++ using keyword class followed by the name of the class.

The body of the class is defined inside the curly brackets and terminated by a semicolon at the end.

```
class className {
   // some data
   // some functions
};
```

For example,

```
class Room {
    public:
        double length;
        double breadth;
        double height;

        double calculateArea(){
            return length * breadth;
        }

        double calculateVolume(){
            return length * breadth * height;
        }

};
```

Here, we defined a class named Room.

The variables length, breadth, and height declared inside the class are known as data members. And, the functions calculateArea() and calculateVolume() are known as member functions of a class.

C++ Objects

When a class is defined, only the specification for the object is defined; no memory or storage is allocated. To use the data and access functions defined in the class, we need to create objects.

Syntax to Define Object in C++

```
className objectVariableName;
```

We can create objects of Room class (defined in the above example) as follows:

```
// sample function
void sampleFunction() {
    // create objects
    Room room1, room2;
}

int main(){
    // create objects
    Room room3, room4;
}
```

Here, two objects room1 and room2 of the Room class are created in sampleFunction(). Similarly, the objects room3 and room4 are created in main().

As we can see, we can create objects of a class in any function of the program. We can also create objects of a class within the class itself, or in other classes.

Also, we can create as many objects as we want from a single class.