

C++ Classes and Objects

C++ Class

Class is a concept of Object Oriented Programming. Class allows user to create a user defined datatype, with custom properties and functions.

In this tutorial, we shall learn how to define a class and how to use it in our C++ programs.

C++ Class Structure

Following code snippet illustrates the high level structure of a class in C++.

```
class ClassName {  
    //attributes  
    //methods  
};
```

So, class keyword is used to define a class. And a ClassName is used to identify the class. Flower braces define the body of class and a semi-colon after the body ends the class definition.

A class can contain attributes and methods. Attributes are variables and Methods are functions that belong to the class alone.

Also, we need to specify Access Specifiers for attributes and methods. This results in selective accessibility of attributes and methods to outside the class.

C++ Class Attributes

Class attributes are just variables from a general programming point of view. But when it comes to Object Oriented Programming, these class attributes define the state of the class objects.

Following class defines a class named Student, with three attributes.

```
class Student {  
    string name;  
    int rollno;  
    int section;
```

```
    ...  
};
```

C++ Class Methods

Methods of a class defines the behavior of the class objects. Class methods are functions that can be accessed within the class or on the class objects.

Following example, defines a class named `Student` with method `printDetails()`.

```
class Student {  
    //attributes  
    string name;  
    int rollno;  
    int section;  
  
    //methods  
    void printDetails(){  
        cout << "Name : " << name << endl;  
        cout << "Roll Number : " << name << endl;  
        cout << "Section : " << name << endl;  
    }  
};
```

C++ Access Specifiers

By default **private** is the access modifier of class members, be it attributes or methods. Meaning, you cannot access the class members outside the class. So, if you would like to modify the access to specific modifier(s) or methods, place the required access specifier before those variables or methods as shown below.

In the following example, we have defined a class with `name` and `rollno` attributes with **public** access. `section` attribute has a **private** access. Then the method `printDetails()` is given **public** access using the public access modifier.

```
class Student {  
    public:  
        string name;  
        int rollno;  
  
    private:  
        int section;  
  
    public:  
        void printDetails(){  
            cout << "Name : " << name << endl;  
            cout << "Roll Number : " << name << endl;  
            cout << "Section : " << name << endl;  
        }  
};
```

If access specifier is not explicitly given to an attribute or method, the program checks if access specifier has been provided for the previous class member. If so, then that access specifier is assigned to this attribute or method.

C++ Create Class Object

To create a class object, you have to declare a variable with the class type, just like you declare an integer with variable name and int datatype.

In the following, example, we shall define a class named Student and create an object for this class in the main method.

C++ Program

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

class Student {
public:
    string name;
    int rollno;
    int section;

public:
    void printDetails(){
        cout << "Name : " << name << endl;
        cout << "Roll Number : " << name << endl;
        cout << "Section : " << name << endl;
    }
};

int main() {
    //create class object
    Student student_1;
}
```

An object with variable name `student_1` of type `Student` has been created.

C++ Access Class Members

After creating object, you can access the class members using dot operator. You can either assign a value to the attribute, or read the value. When it comes to methods, you can just call them.

In the following example, we have defined a class `Student`, created an object of type `Student`, assigned values to the attributes of the object and made a call to the member function.

C++ Program

```

#include <iostream>
using namespace std;

class Student {
public:
    string name;
    int rollno;
    int section;

public:
    void printDetails(){
        cout << "Name : " << name << endl;
        cout << "Roll Number : " << rollno << endl;
        cout << "Section : " << section << endl;
    }
};

int main() {
    //create class object
    Student student_1;

    //modify class object attributes
    student_1.name = "Angel";
    student_1.rollno = 32;
    student_1.section = 3;

    //call functions on class objects
    student_1.printDetails();
}

```

Output

```

Name : Angel
Roll Number : 32
Section : 3

```

C++ Class Constructors

You can assign values to class attributes while creating a class object. This can be done using constructors of class. Constructors are methods that does not return a value and have the same name as that of Class.

C++ Program

```

#include <iostream>
using namespace std;

class Student {
public:
    string name;
    int rollno;
    int section;

    //constructor
    Student(string x, int y, int z) {
        name = x;
        rollno = y;
        section = z;
    }
};

```

```

        name = x;
        rollno = y;
        section = z;
    }

public:
void printDetails(){
    cout << "Name : " << name << endl;
    cout << "Roll Number : " << rollno << endl;
    cout << "Section : " << section << endl;
}
};

int main() {
    //create class object with call to constructor
    Student student_1("Angel", 32, 3);

    //call functions on class objects
    student_1.printDetails();
}

```

Output

```

Name : Angel
Roll Number : 32
Section : 3

```

Conclusion

In this [C++ Tutorial](#), we learned about Class in C++, structure of a class, class members: attributes and methods, access specifiers to class members, class constructors.

C++ Tutorials

- ◆ [C++ Tutorial](#)
- ◆ [C++ Hello World Program](#)
- ◆ [C++ If Else](#)
- ◆ [C++ Switch](#)
- ◆ [C++ Ternary Operator](#)
- ◆ [C++ Logical Operations](#)
- ◆ [C++ Arithmetic Operations](#)
- ◆ [C++ While Loop](#)
- ◆ [C++ Do-While Loop](#)
- ◆ [C++ For Loop](#)

- ◆ [C++ ForEach](#)
- ◆ [C++ Continue](#)
- ◆ [C++ Break](#)
- ◆ [C++ Comments](#)
- ◆ [C++ Recursion](#)
- ◆ [C++ Try Catch](#)
- ◆ [C++ String Operations](#)
- ◆ [C++ Array Operations](#)
- ◆ [C++ Vector Operations](#)
- ◆ [C++ Input Output Operations](#)
- ⇒ **C++ Class**
- ◆ [C++ Programs](#)