

C++ Classes and Objects

[< Previous](#)[Next >](#)

C++ Classes/Objects

C++ is an object-oriented programming language.

Everything in C++ is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an **object**. The car has **attributes**, such as weight and color, and **methods**, such as drive and brake.

Attributes and methods are basically **variables** and **functions** that belongs to the class. These are often referred to as "class members".

A class is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects.

Create a Class

To create a class, use the **class** keyword:

Example

Create a class called "**MyClass**":

```
class MyClass {           // The class
public:                   // Access specifier
    int myNum;            // Attribute (int variable)
```



Example explained

- The `class` keyword is used to create a class called `MyClass`.
- The `public` keyword is an **access specifier**, which specifies that members (attributes and methods) of the class are accessible from outside the class. You will learn more about access specifiers later.
- Inside the class, there is an integer variable `myNum` and a string variable `myString`. When variables are declared within a class, they are called **attributes**.
- At last, end the class definition with a semicolon `;`.

Create an Object

In C++, an object is created from a class. We have already created the class named `MyClass`, so now we can use this to create objects.

To create an object of `MyClass`, specify the class name, followed by the object name.

To access the class attributes (`myNum` and `myString`), use the dot syntax (`.`) on the object:

Example

Create an object called "`myObj`" and access the attributes:

```
class MyClass {           // The class
    public:                // Access specifier
        int myNum;         // Attribute (int variable)
        string myString;   // Attribute (string variable)
};

int main() {
    MyClass myObj;        // Create an object of MyClass

    // Access attributes and set values
    myObj.myNum = 15;
```



```
// Print attribute values
cout << myObj.myNum << "\n";
cout << myObj.myString;
return 0;
}
```

[Try it Yourself »](#)

Multiple Objects

You can create multiple objects of one class:

Example

```
// Create a Car class with some attributes
class Car {
public:
    string brand;
    string model;
    int year;
};

int main() {
    // Create an object of Car
    Car carObj1;
    carObj1.brand = "BMW";
    carObj1.model = "X5";
    carObj1.year = 1999;

    // Create another object of Car
    Car carObj2;
    carObj2.brand = "Ford";
    carObj2.model = "Mustang";
    carObj2.year = 1969;

    // Print attribute values
    cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year <<
"\n";
    cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year <<
"\n";
}
```