Difference Between Base Class and Derived Class in C++

We will see the individual meaning of base class and derived class, after understanding the difference between base class and derived class in C++. The major difference between base class and derived class in object-oriented programming are listed in the table below. You can also check the difference between procedural and object-oriented programming.

Key Difference Between Base Class and Derived Class

Base Class	**Derived Class**
It is defined as the class from which properties are acquired.	It is defined as the class which acquires the properties.
Cannot acquire or inherit properties from any other class.	Can inherit properties and methods from the base class.
Also called Parent class or Superclass.	Also called Child class or Subclass.

What is Base Class in C++?

A base class is any existing class from which other classes can be derived in C++ or object-oriented programming. A base class is sometimes alternately called a parent class or a superclass. Other non-base classes can acquire the members and functions of a base class.

The syntax of the base class is the same as any other regular class syntax. The syntax is as follows:

```
class base_class

{

//class_members

//class_member_functions

}
```

Note: C++ is a case-sensitive language, so the keyword class is always in lower case; however, to name a class, we can use any notation. In the above example, snake notation has been used. One can also use camel notation for the class name.

What is Derived Class in C++?

A derived class in object-oriented programming is defined as one that can inherit or acquire properties of an already existing class called the base class. The base class can be easily accessed by a derived class. One of the striking difference between base class and derived class in C++ is that a derived class provides more functionality with respect to the base class.

Derived classes use access modifiers to access the members of the base class. If the modifier is not specified, then by default, the functionality of the private access modifier is applied to the derived class. The syntax for creating a derived class from the base class is as follows:

```
class base_class

{

//class_members

//class_member_functions

}

class derived_class: public base_class

{

//class_members

//class_member_functions

}
```

Note: The derived classes make use of access modifiers. Access modifiers are used to provide accessibility to the members of the class. There are three types of access modifiers in C++, they are:

- Public
- Private
- Protected