



## Multiple Inheritance Deep Dive with Code Example in C++ | C++ Tutorials for Beginners #41



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

## Multiple Inheritance Deep Dive with Code Example in C++ | C++ Tutorials for Beginners #41

In this tutorial, we will discuss multiple inheritances in C++

### Multiple Inheritances in C++

Multiple inheritances are a type of inheritance in which one derived class is inherited with more than one base class. For example, we have three classes “employee”, “assistant” and “programmer”. If the “programmer” class is inherited from the “employee” and “assistant” class which means that the “programmer” class can now implement the functionalities of the “employee” and “assistant” class. The syntax of inheriting multiple inheritances is shown below.

```
// class DerivedC: visibility-mode base1, visibility-mode base2
// {
//     Class body of class "DerivedC"
// };
```

### Code Snippet 1: Multiple inheritances syntax

As shown in a code snippet 1,

- After writing the class keyword we have to write the derived class name and then put a “:” sign.
- After “:” sign we have to write the visibility mode and then write the base class name and again we have to write the visibility mode and write another base class name.

An example program is shown below to demonstrate the concept of multiple inheritances in C++.

```
class Base1{
protected:
    int base1int;

public:
    void set_base1int(int a)
    {
        base1int = a;
    }
};
```

```
class Base2{
protected:
```

```
    int base2int;

public:
    void set_base2int(int a)
    {
        base2int = a;
    }
};

class Base3{
protected:
    int base3int;

public:
    void set_base3int(int a)
    {
        base3int = a;
    }
};
```

### Code Snippet 2: Base Classes

As shown in Code snippet 2,

- 1<sup>st</sup> we created a “Base1” class which consists of protected data member integer “base1int”.
- 2<sup>nd</sup> the “Base1” class consists of a public function “set\_base1int”. This function will set the value of the data member “base1int”.
- 3<sup>rd</sup> we created a “Base2” class which consists of protected data member integer “base2int”.

- 4<sup>th</sup> the “Base2” class consists of a public function “set\_base2int”. This function will set the value of the data member “base2int”.
- 5<sup>th</sup> we created a “Base3” class which consists of protected data member integer “base3int”.
- 2<sup>nd</sup> the “Base3” class consists of a public function “set\_base3int”. This function will set the value of the data member “base3int”.

The code for the “Derived” class is shown below. “Derived” class will inherit all the base classes.

```
class Derived : public Base1, public Base2, public Base3
{
    public:
        void show(){
            cout << "The value of Base1 is " << base1int<<endl;
            cout << "The value of Base2 is " << base2int<<endl;
            cout << "The value of Base3 is " << base3int<<endl;
            cout << "The sum of these values is " << base1int + base2int + base3int << endl;
        }
};
```

### Code Snippet 3: Derived Class

As shown in a code snippet 3,

- 1<sup>st</sup> we created a “Derived” class which is inheriting “Base1”, “Base2”, and “Base3” classes in public mode.
- 2<sup>nd</sup> the “Derived” class consists of the public member function “show”.
- 4<sup>th</sup> the function “show” will first print the values of “base1int”, “base2int”, and “base3int” individually and then print the sum of all three values.

It can be clearly seen that the class “Derived” is inheriting class “Base1”, “Base2”, and “Base3”. This is an

example of multiple inheritances. The code main program is shown below.

```
int main()
{
    Derived harry;
    harry.set_base1int(25);
    harry.set_base2int(5);
    harry.set_base3int(15);
    harry.show();

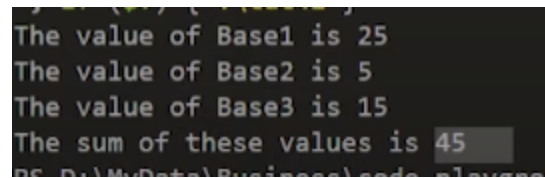
    return 0;
}
```

#### Code Snippet 4: Main Program

As shown in Code snippet 4,

- 1<sup>st</sup> object “harry” is created of the “Derived” data type.
- 2<sup>nd</sup> the function “set\_base1int” is called by the object “harry” and the value “25” is passed.
- 3<sup>rd</sup> the function “set\_base2int” is called by the object “harry” and the value “5” is passed.
- 4<sup>th</sup> the function “set\_base3int” is called by the object “harry” and the value “15” is passed.
- 4<sup>th</sup> the function “show” is called by the object “harry”.

The output for the following program is shown in figure 1.



```
The value of Base1 is 25
The value of Base2 is 5
The value of Base3 is 15
The sum of these values is 45
```

**Figure 1:** Program Output

[Previous](#)

[Next](#)



CodeWithHarry

Copyright © 2022 CodeWithHarry.com

