



Single Inheritance Deep Dive: Examples + Code | C++ Tutorials for Beginners #38



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

Single Inheritance Deep Dive: Examples + Code | C++ Tutorials for Beginners #38

In this tutorial, we will discuss single inheritance in C++

Single Inheritance in C++

Single inheritance is a type of inheritance in which a derived class is inherited with only one base class. For example, we have two classes “employee” and “programmer”. If the “programmer” class is inherited from the “employee” class which means that the “programmer” class can now implement the functionalities of the “employee” class.

An example program to demonstrate the concept of single inheritance in C++ is shown below.

```
class Base
```

```
{
    int data1; // private by default and is not inheritable
public:
    int data2;
    void setData();
    int getData1();
    int getData2();
};

void Base ::setData(void)
{
    data1 = 10;
    data2 = 20;
}

int Base::getData1()
{
    return data1;
}

int Base::getData2()
{
    return data2;
}
```

Code Snippet 1: Base Class

As shown in a code snippet 1,

- 1st we created a “base” class which consists of private data member’s integer “data1” and public data member integer “data2”.
- 2nd the “base” class consists of three member functions “setData”, “getData1”, and “getData2”.
- 3rd the function “setData” will assign the values “10” and “20” to the data members “data1” and “data2”.
- 4th the function “getData1” will return the value of the data member “data1”.
- 5th the function “getData2” will return the value of the data member “data2”.

The derived class will inherit the base class which is shown below.

```
class Derived : public Base
{ // Class is being derived publically
    int data3;

public:
    void process();
    void display();
};

void Derived ::process()
{
    data3 = data2 * getData1();
}

void Derived ::display()
{
    cout << "Value of data 1 is " << getData1() << endl;
```

```
    cout << "Value of data 2 is " << data2 << endl;  
    cout << "Value of data 3 is " << data3 << endl;  
}
```

Code Snippet 2: Derived Class

As shown in Code snippet 2,

- 1st we created a “derived” class which is inheriting the base class publically. The “derived” class consists of private data member’s integer “data3”.
- 2nd the “derived” class consists of two public member functions “process” and “display”.
- 3rd the function “process” will multiply the values “data2” and “data1”; and store the values in the variable “data3”.
- 4th the function “display” will print the values of the data member “data1”, “data2”, and “data3”.

The main program is shown in code snippet 3.

```
int main()  
{  
    Derived der;  
    der.setData();  
    der.process();  
    der.display();  
  
    return 0;  
}
```

Code Snippet 3: Main Program

As shown in a code snippet 3,

- 1st object “der” is created of the “derived” data type.
- 2nd the function “setData” is called by the object “der”. This function will set the values of the data members “data1” and “data2”
- 3rd the function “process” is called by the object “der”. This function will multiply the values “data2” and “data1”; and store their value in the variable “data3”.
- 4th the function “display” is called by the object “der”. This function will print the values of the data member “data1”, “data2”, and “data3”.

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

```
PS D:\MyData\Business\cod...
Value of data 1 is 10
Value of data 2 is 20
Value of data 3 is 200
```

Figure 1: Program Output

[Previous](#)[Next](#)

CodeWithHarry

Copyright © 2022 CodeWithHarry.com

