Inheritance Syntax & Visibility Mode in C++ | C++ Tutorials for Beginners #37

In this tutorial, we will discuss inheritance syntax and visibility mode in C++

Inheritance Syntax and Visibility mode in C++

Inheritance is a process of inheriting attributes of the base class by a derived class. The syntax of the derived class is shown below.

```
 // Derived Class syntax
class {{derived-class-name}} : {{visibility-mode}} {{base-class-name}}
{
    class members/methods/etc...
```

```
}
```

Code Snippet 1: Derived Class 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.

Note:

- Default visibility mode is private
- Public Visibility Mode: Public members of the base class becomes Public members of the derived class
- Private Visibility Mode: Public members of the base class become private members of the derived class
- Private members are never inherited

An example program is shown below to demonstrate the concept of inheritance.

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

// Base Class
class Employee
{
public:
    int id;
    float salary;
    Employee(int inpId)
    {
        id = inpId;
```

```
        salary = 34.0;
    }
    Employee() {}
};

// Creating a Programmer class derived from Employee Base class
class Programmer : public Employee
{
public:
    int languageCode;
    Programmer(int inpId)
    {
        id = inpId;
        languageCode = 9;
    }
    void getData(){
        cout<<id<<endl;
    }
};
```

Code Snippet 2: Inheritance Example Program

As shown in Code snippet 2,

- 1st we created an “employee” class which consists of public data member’s integer “id” and float “salary”.
- 2nd the “employee” class consists of a parameterized constructor that takes an integer “inpid” parameter and assigns its value to the data member “id”. The value of variable “salary” is set to “34”.
- 3rd the “employee” class also consists of default constructor.

- 4th we created a “programmer” class that is inheriting “employee” class. The main thing to note here is that the “visibility-mode” is “public”.
- 5th the “programmer” class consists of public data member’s integer “languageCode”.
- 6th the “programmer” class consists of a parameterized constructor that takes an integer “inpid” parameter and assigns its value to the data member “id”. The value of variable “languageCode” is set to “9”.
- 7th “programmer” class consists of a function “getData” which will print the value of the variable “id”.

The main program is shown in code snippet 3.

```
int main()
{
    Employee harry(1), rohan(2);
    cout << harry.salary << endl;
    cout << rohan.salary << endl;
    Programmer skillF(10);
    cout << skillF.languageCode<<endl;
    cout << skillF.id<<endl;
    skillF.getData();
    return 0;
}
```

Code Snippet 3: Main Program

As shown in a code snippet 3,

- 1st objects “harry” and “rohan” is created of the “employee” data type. Object “harry” is passed with the value “1” and the object “rohan” is passed with the value “2”.
- 2nd the “salary” of both objects “rohan” and “harry” are printed.

- 3rd object “skillF” is created of the “programmer” data type. Object “skillF” is passed with the value “10”.
- 4th the “languageCode” and “id” of both object “skillF” is printed.
- 5th the function “getData” is called by the “skillF” object. This will print the “id”.

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

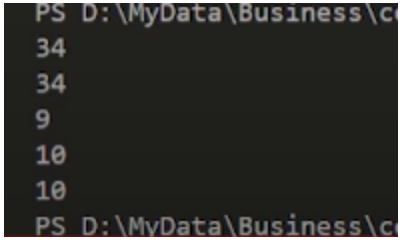

Figure 1: Program Output

Previous Next