Multilevel Inheritance Deep Dive with Code Example in C++ | C++ Tutorials for Beginners #40

Multilevel Inheritance Deep Dive with Code Example in C++ | C++ Tutorials for Beginners #40

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

Multilevel Inheritance in C++

Multilevel inheritance is a type of inheritance in which one derived class is inherited from another derived class. For example, we have three classes “animal”, “mammal” and “cow”. If the “mammal” class is inherited from the “animal” class and “cow” class is inherited from “mammal” which means that the “mammal” class can now implement the functionalities of “animal” and “cow” class can now implement the functionalities of “mammal” class.

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

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

class Student
{
protected:
    int roll_number;

public:
    void set_roll_number(int);
    void get_roll_number(void);
};

void Student ::set_roll_number(int r)
{
    roll_number = r;
}

void Student ::get_roll_number()
{
    cout << "The roll number is " << roll_number << endl;
}
```

Code Snippet 1: Student Class

As shown in a code snippet 1,

- 1st we created a “student” class which consists of protected data member integer “roll_number”.
- 2nd the “student” class consists of a public function “set_roll_number” and “get_roll_number”
- 3rd the function “set_roll_number” will set the value of the data member “roll_number”.
- 4th the function “get_roll_number” will print the value of the data member “roll_number”.

The code for the “exam” class is shown below which is inheriting the “student” class

```
class Exam : public Student
{
protected:
    float maths;
    float physics;

public:
    void set_marks(float, float);
    void get_marks(void);
};

void Exam ::set_marks(float m1, float m2)
{
    maths = m1;
    physics = m2;
}

void Exam ::get_marks()
{
    cout << "The marks obtained in maths are: " << maths << endl;
```

```
    cout << "The marks obtained in physics are: " << physics << endl;
}
```

Code Snippet 2: Exam Class

As shown in Code snippet 2,

- 1st we created an “exam” class that is inheriting “student” class in public mode.
- 2nd the “exam” class consists of protected data members float “math” and float “physics”.
- 3rd the “exam” class consists of public member functions “set_marks” and “get_marks”.
- 4th the function “set_marks” will set the value of the data members “math” and “physics”.
- 5th the function “get_marks” will print the value of the data members “math” and “physics”.

The code for the “result” class is shown below which is inheriting the “exam” class

```
class Result : public Exam
{
    float percentage;

public:
    void display_results()
    {
        get_roll_number();
        get_marks();
        cout << "Your result is " << (maths + physics) / 2 << "%" << endl;
    }
};
```

Code Snippet 3: Result Class

As shown in a code snippet 3,

- 1st we created a “Result” class which is inheriting the “Exam” class in public mode.
- 2nd the “Result” class consists of private data member’s float “percentage”.
- 3rd the “exam” class consists of the public member function “display_results”.
- 4th the function “display_results” will call the “get_roll_number” and “get_marks” functions, and add the values of “math” and “physics” variables then divide that value with “2” to get a percentage and prints it.

It can be clearly seen that the class “Exam” is inheriting class “student” and class “Results” is inheriting class “Exam”; which is an example of multilevel inheritance. The code main program is shown below.

```
int main()
{
    Result harry;
    harry.set_roll_number(420);
    harry.set_marks(94.0, 90.0);
    harry.display_results();
    return 0;
}
```

Code Snippet 4: Main Program

As shown in Code snippet 4,

- 1st object “harry” is created of the “Result” data type.
- 2nd the function “set_roll_number” is called by the object “harry” and the value “420” is passed.
- 3rd the function “set_marks” is called by the object “harry” and the values “94.0” and “90.0” are passed.
- 4th the function “display_results” is called by the object “harry”.

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

```
PS D:\MyData\Business\code playground\C
```

```
The roll number is 420
The marks obtained in maths are: 94
The marks obtained in physics are: 90
Your percentage is 92%
PS D:\MyData\Business\code playground\C
```

Figure 1: Program Output

Previous

Next

CodeWithHarry

Copyright © 2022 CodeWithHarry.com