

C++ Inheritance

Reusability is one of the important characteristics of Object Oriented Programming (OOP). Instead of trying to write programs repeatedly, using existing code is a good practice for the programmer to reduce development time and avoid mistakes. In C++, reusability is possible by using Inheritance.

Table of Contents

- [1. What is Inheritance?](#)
- [2. What are Base class and Derived class?](#)
- [3. Forms of Inheritance](#)
- [4. Single Inheritance](#)
- [5. Multiple Inheritance](#)
- [6. Hierarchical Inheritance](#)
- [7. Program for Hierarchical Inheritance](#)
- [8. Multilevel Inheritance](#)
- [9. Program for Multilevel Inheritance](#)
- [10. Hybrid Inheritance](#)

What is Inheritance?

The technique of deriving a new class from an old one is called inheritance. The old class is referred to as base class and the new class is referred to as derived class or subclass. Inheritance concept allows programmers to define a class in terms of another class, which makes creating and maintaining application easier. When writing a new class, instead of writing new data member and member functions all over again, programmers can make a bonding of the new class with the old one that the new class should inherit the members of the existing class. A class can get derived from one or more classes, which means it can inherit data and functions from multiple base classes.

Here is the syntax how inheritance is performed in C++:

```
class derived-class: visibility-mode base-class
```

Visibility mode is used in the inheritance of C++ to show or relate how base classes are viewed with respect to derived class. When one class gets inherited from another, visibility mode is used to inherit all the public and protected members of the base class. Private members never get inherited and hence do not take part in visibility. By default, visibility mode remains "private".

What are Base class and Derived class?

The existing class from which the derived class gets inherited is known as the base class. It acts as a parent for its child class and all its properties i.e. public and protected members get inherited to its derived class.

A derived class can be defined by specifying its relationship with the base class in addition to its own details, i.e. members.

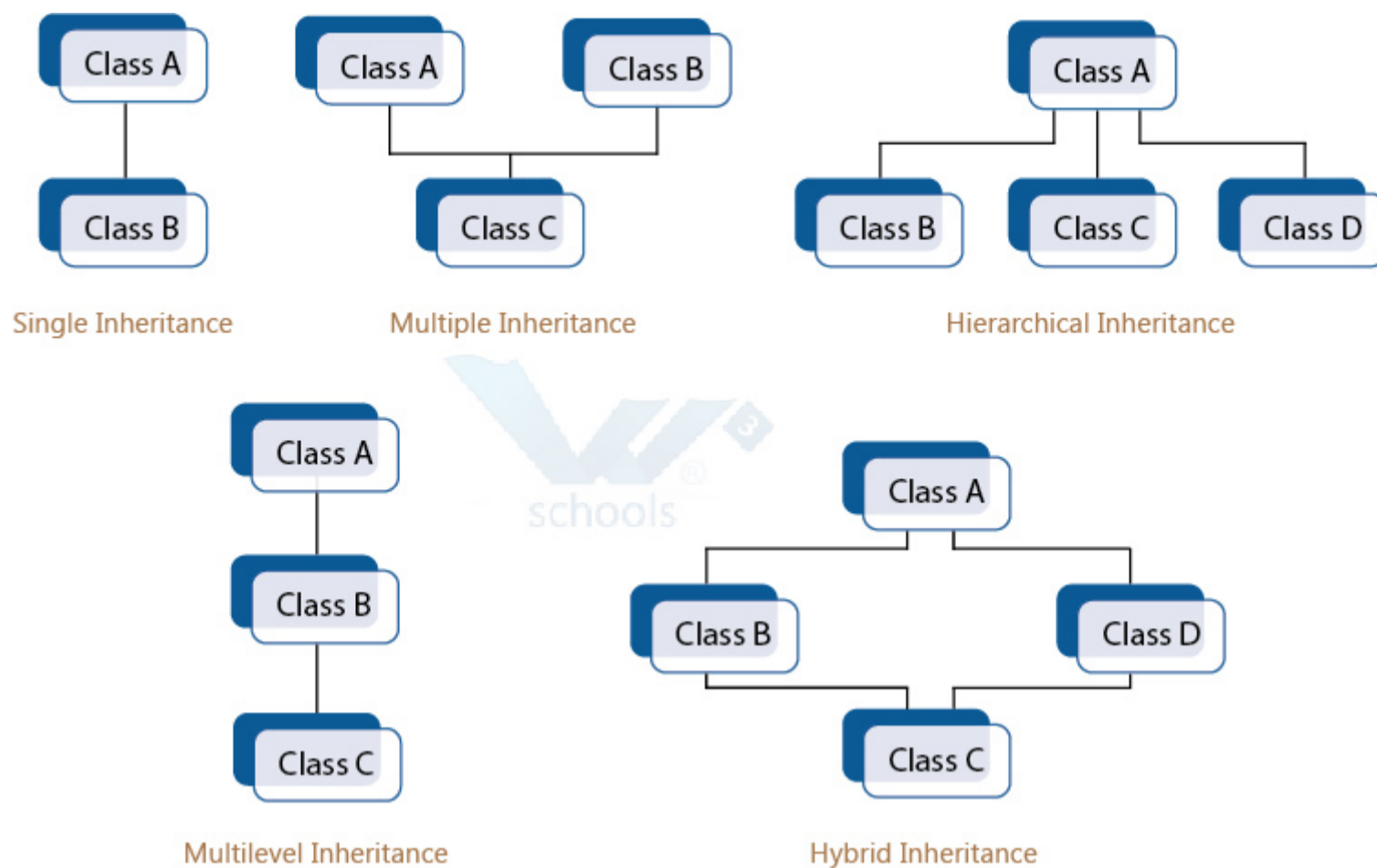
The general form of defining a derived class is:

```
class derived-class_name : visivility-mode base-class_name
{
    . . . . // members of the derived class
    . . . .
};
```

Forms of Inheritance

C++ offers five types of Inheritance. They are:

- Single Inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Multilevel Inheritance
- Hybrid Inheritance (also known as Virtual Inheritance)



Single Inheritance

In single inheritance, there is only one base class and one derived class. The Derived class gets inherited from its base class. This is the simplest form of inheritance. In the above figure, fig(a) is the diagram for single inheritance.

Multiple Inheritance

In this type of inheritance, a single derived class may inherit from two or more base classes. In the above list of figures, fig(b) is the structure of Multiple Inheritance.

Program for Multiple Inheritance:

Example:

```
public:
    void getsm()
    {
        cout << "\nEnter the mark for Extra Curricular Activities: "; cin >> xm;
    }
};
class output : public stud, public extracurricular {
    int tot, avg;

public:
    void display()

    {
        tot = (m1 + m2 + xm);
        avg = tot / 3;
        cout << "\n\n\tRoll No      : " << roll << "\n\tTotal      : " << tot;
        cout << "\n\tAverage      : " << avg;
    }
};
int main()
{
    output o;
    o.get();
    o.getsm();
    o.display();
}
```

Output:

```

Q:\example\example.exe
Enter the Roll No.: 225
Enter the two highest marks: 35
40
Enter the mark for Extra Curricular Activities: 36

Roll No    : 225
Total     : 111
Average   : 37_

```

Hierarchical Inheritance

In this type of inheritance, multiple derived classes get inherited from a single base class. In the above list of figures, fig(c) is the structure of Hierarchical Inheritance.

Syntax:

```

class base_classname {
    properties;
    methods;
};
class derived_class1 : visibility_mode base_classname {
    properties;
    methods;
};
class derived_class2 : visibility_mode base_classname {
    properties;
    methods;
};
... ..
... ..
class derived_classN : visibility_mode base_classname {
    properties;
    methods;
};

```

Program for Hierarchical Inheritance

Example:

```

#include <iostream>
#include <string.h>
using namespace std;

class member {
    char gender[10];
    int age;

public:
    void get()
    {
        cout << "Age: "; cin >> age;
        cout << "Gender: "; cin >> gender;
    }
    void disp()
    {
        cout << "Age: " << age << endl;
        cout << "Gender: " << gender << endl;
    }
};

class stud : public member {
    char level[20];

public:
    void getdata()

```

Output:

```

Q:\example\example.exe
Student
Enter data
Age: 12
Gender: M
Class: 6

Displaying data
Age: 12
Gender: M

Staff Data
Enter data
Age: 32
Gender: F
Salary: Rs.6000

Displaying data
Age: 32
Gender: F

```

Multilevel Inheritance

The classes can also be derived from the classes that are already derived. This type of inheritance is called multilevel inheritance.

Program for Multilevel Inheritance

Example:

```

#include <iostream>
using namespace std;

class base {
public:
    void display1()
    {
        cout << "\nBase class content.";
    }
};

class derived : public base {
public:
    void display2()
    {
        cout << "1st derived class content.";
    }
};

class derived2 : public derived {
    void display3()
    {
        cout << "\n2nd Derived class content.";
    }
};

```

Output:

```

Q:\example\example.exe
1st derived class content.
Base class content._

```

Hybrid Inheritance

This is a Mixture of two or More Inheritance and in this Inheritance, a Code May Contains two or Three types of inheritance in Single Code. In the above figure, the fig(5) is the diagram for Hybrid inheritance.

Next Chapter : [C++ Friend Function](#) >

