CSE201: Object Oriented Programming C++

Inheritance

Inheritance in Object Oriented Programming can be described as a process of creating new classes from existing classes. New classes inherit some of the properties and behavior of the existing classes. An existing class that is "parent" of a new class is called a base class and the new class is call the child class. It is a technique of code reuse.

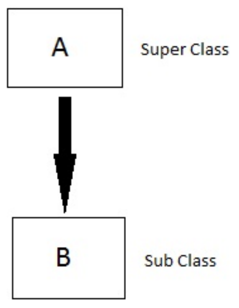
Types of Inheritance

There are 5 different types of Inheritance. Namely,

1. Single Inheritance
2. Multiple Inheritance
3. Hierarchical Inheritance
4. Multilevel Inheritance
5. Hybrid or Virtual Inheritance

Single Inheritance

In this type of inheritance one derived class inherits from only one base class. It is the simplest form of Inheritance.

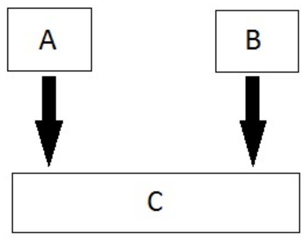


class name of the derived class: Access specifier Name of the Base class

{ Body of the Derived class;}

Multiple Inheritance

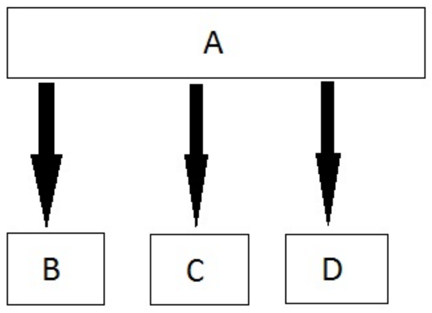
In this type of inheritance, a single derived class may inherit from two or more than two base classes.



class Derived class name: Access specifier Base class1 name, Access specifier Base class2 name……… Access specifier Base class n name { Body of the Derived class}

Hierarchical Inheritance

In this type of inheritance, multiple derived classes inherit from a single base class.



Class Derived class 1 name: Access Specifier Base class name { Body of Derived class 1;}

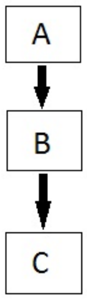
Class Derived class 2 name: Access Specifier Base class name { Body of Derived class 2;}

…..

Class Derived class n name: Access Specifier Base class name { Body of Derived class n;}

Multilevel Inheritance

In this type of inheritance, the derived class inherits from a class, which in turn inherits from some other class. The Super class for one, is sub class for the other.

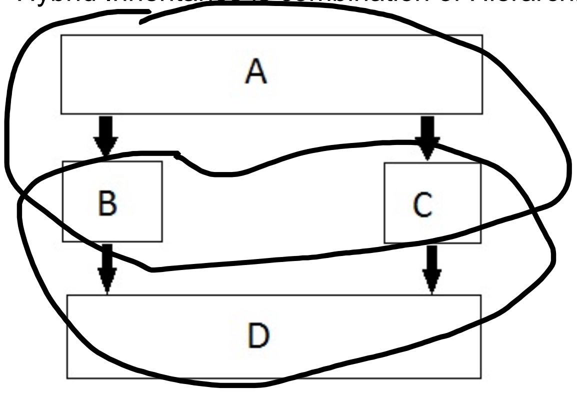


Class Derived class 1 name: Access Specifier Base class name { Body of Derived class 1;}

Class Derived class 2 name: Access Specifier Derived class 1 { Body of Derived class 2;}

Hybrid or Virtual Inheritance

Hybrid Inheritance is combination of Hierarchical and Multilevel Inheritance.



Example of Inheritance

a. *Single Inheritances*

When a **single** class is derived from a **single** parent class, it is called **Single inheritance.**

b. *Multiple Inheritances*

A C++ class can inherit members from more than one class and here is the extended syntax:

```
class derived-class: access baseA, access baseB....
```

```
#include <iostream>

using namespace std;

// Base class Shape
class Shape {
   public:
      void setWidth(int w) {
         width = w;
      }

      void setHeight(int h) {
         height = h;
      }

   protected:
      int width;
      int height;
};

// Base class PaintCost
class PaintCost  {
   public:
      int getCost(int area) {
         return area * 70;
      }
};

// Derived class
class Rectangle: public Shape, public PaintCost {
   public:
      int getArea() {
         return (width * height);
      }
};

int main() {
   Rectangle Rect;
   int area;

   Rect.setWidth(5);
   Rect.setHeight(7);

   area = Rect.getArea();

   // Print the area of the object.
   cout << "Total area: " << Rect.getArea() << endl;

   // Print the total cost of painting
   cout << "Total paint cost: $" << Rect.getCost(area) << endl;

   return 0;
}
```

c. Multi-level inheritance

In this inheritance, a derived class is created from another derived class.

d. Hierarchical inheritance

In this inheritance, more than one derived classes are created from a single base.

e. **Hybrid or Virtual inheritance**

This is combination of more than one inheritance. Hence, it may be a combination of Multilevel and Multiple inheritance or Hierarchical and Multilevel inheritance or Hierarchical and Multipath inheritance or Hierarchical, Multilevel and Multiple inheritance.