

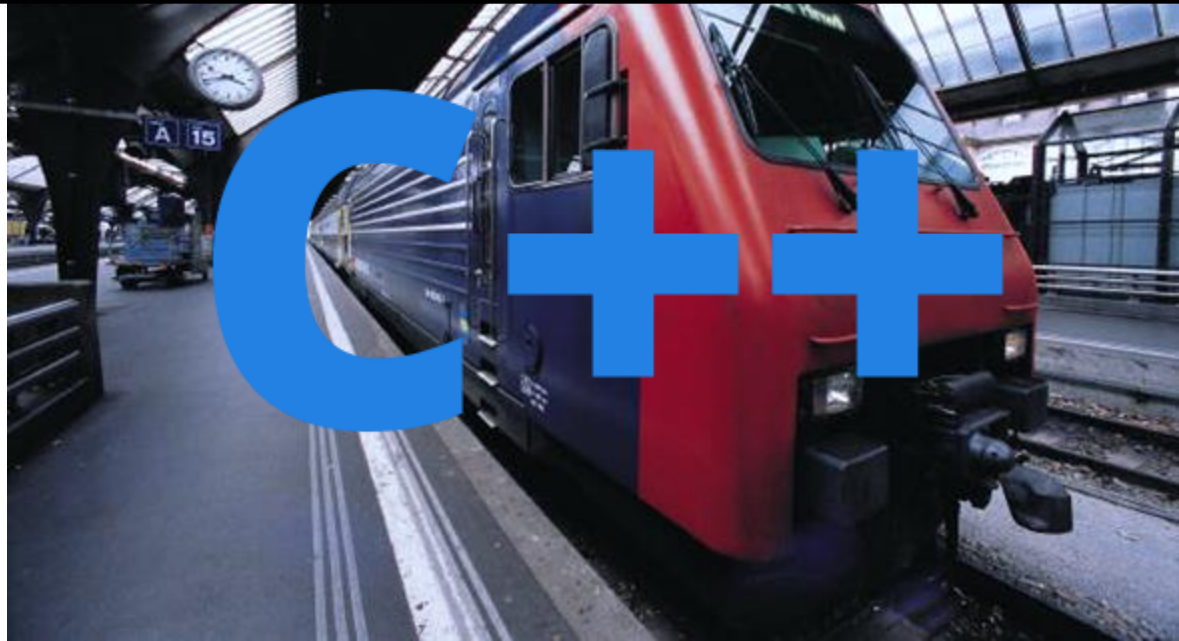


**Build Your IT Skill**

2020

## ណែនាំស្តាប់ពី Inheritance Of OOP

Single Inheritance      Multiple Inheritance



## ណែនាំអោយស្គាល់ពី Inheritance

### I. ដូចម្តេចទៅដែលហៅថា Inheritance?

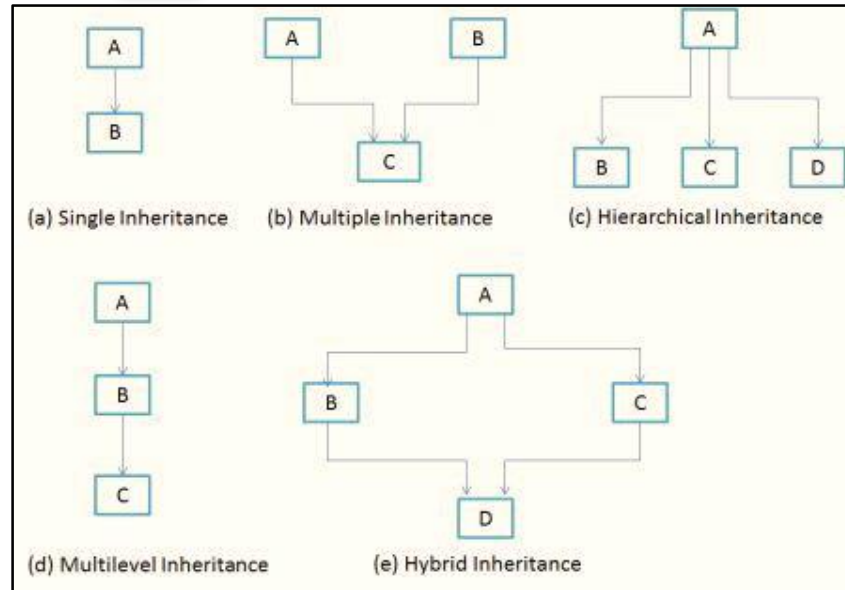
Inheritance គឺជាដំណើរនៃការកកើត Class ថ្មីមួយចេញពី Class ដែលមានស្រាប់ ដែល Class ថ្មី គេហៅថា Sub Class ឬ Derived Class និង Class មានស្រាប់គេហៅថា Base Class ឬ Super Class។

ទំរង់ដូច្នេះ

```
1
2 class Super_Class{
3
4     Data Member
5
6     Function Member
7     .....
8
9 };
10
11 class Sub_Class: public/protected/private{
12
13     Data Member
14
15     Function Member
16     .....
17 };
18
19
```

នៅក្នុង Inheritance គេបែងចែកជា ២ប្រភេទគឺ៖

- ១) Single Inheritance
- ២) Multiple Inheritance
- ៣) Hierarchical
- ៤) Multilevel
- ៥) Hybrid

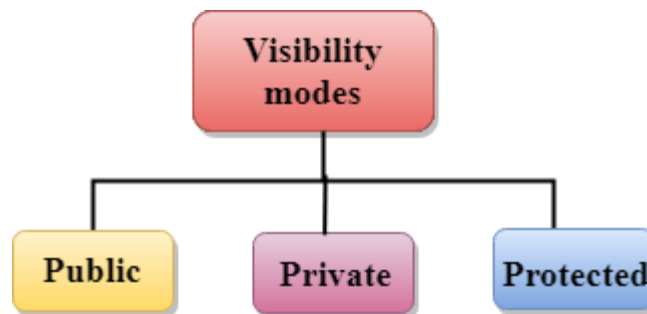


តែការបែងចែកជាចំណែកធំៗនៃ Inheritance គេបែងចែកជាពីរសំខាន់គឺ៖

១) Single Inheritance

២) Multiple Inheritance

ខាងក្រោមគឺជាកំរិតនៃ Level របស់ Inheritance៖



អ្នកត្រូវចាំថាកំរិតនៃការ Accessing Private គឺមិនអាចប្រាស់បានសំរាប់ទំរង់ទាំងអស់ នៃ Inheritance ៖

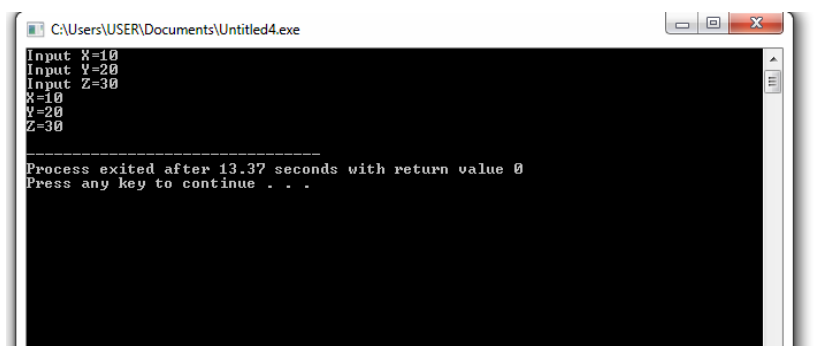
Base class visibility	Derived class visibility		
	Public	Private	Protected
Private	Not Inherited	Not Inherited	Not Inherited
Protected	Protected	Private	Protected
Public	Public	Private	Protected

1.1. Single Inheritance: គឺជាប្រភេទ Inheritance ដែលមាន Base Class មួយ និង Sub Class អាចមាន១ ឬ ច្រើន។

ឧទាហរណ៍ ១

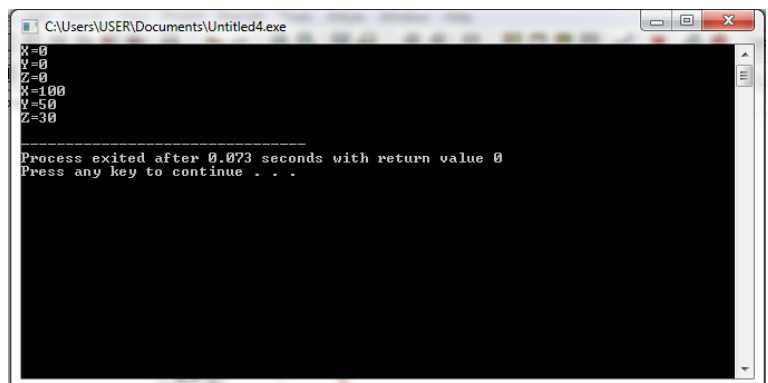
```
1  #include<iostream>
2  using namespace std;
3  class Test1{
4      protected:
5          int x;
6          int y;
7  };
8  class Test2:public Test1{
9      private:
10         int z;
11     public:
12         void Input()
13         {
14             cout<<"Input X="<<cin>>x;
15             cout<<"Input Y="<<cin>>y;
16             cout<<"Input Z="<<cin>>z;
17         }
18         void Output()
19         {
20             cout<<"X="<<x<<endl;
21             cout<<"Y="<<y<<endl;
22             cout<<"Z="<<z<<endl;
23         }
24     };
25     int main()
26     {
27         Test2 obj2;
28         obj2.Input();
29         obj2.Output();
30     }
```

លទ្ធផលទទួលបាន៖



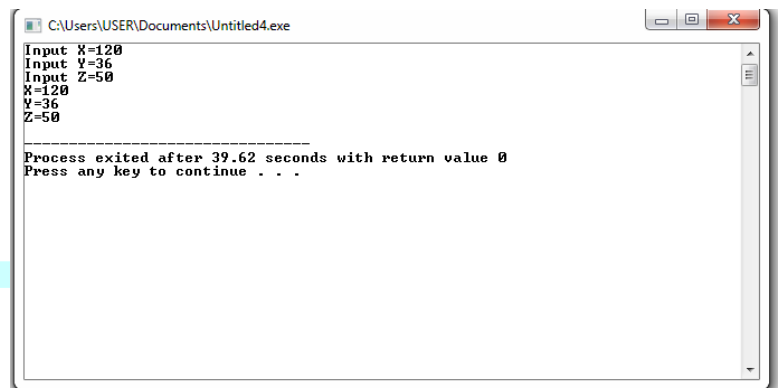
## ឧទាហរណ៍ ២៖

```
1  #include<iostream>
2  using namespace std;
3  class Test1{
4      protected:
5          int x;
6          int y;
7      public:
8          Test1()
9          {
10             x=0;
11             y=0;
12         }
13         Test1(int a,int b)
14         {
15             x=a;
16             y=b;
17         }
18     };
19     class Test2:public Test1{
20     private:
21         int z;
22     public:
23         Test2()
24         {
25             Test1:Test1();
26             z=0;
27         }
28         Test2(int a,int b,int c)
29         {
30             x=a;
31             y=b;
32             z=c;
33         }
34         void Input()
35         {
36             cout<<"Input X=";<<cin>>x;
37             cout<<"Input Y=";<<cin>>y;
38             cout<<"Input Z=";<<cin>>z;
39         }
40         void Output()
41         {
42             cout<<"X="<<x<<endl;
43             cout<<"Y="<<y<<endl;
44             cout<<"Z="<<z<<endl;
45         }
46     };
47     int main()
48     {
49         Test2 obj2;
50         obj2.Output();
51         Test2 obj3(100,50,30);
52         obj3.Output();
53     }
```



## ឧទាហរណ៍ ៣៖

```
1  #include<iostream>
2  using namespace std;
3  class Test1{
4      protected:
5          int x;
6          int y;
7      public:
8          Test1()
9          {
10             x=0;
11             y=0;
12         }
13         Test1(int a,int b)
14         {
15             x=a;
16             y=b;
17         }
18         void Input()
19         {
20             cout<<"Input X=";<<cin>>x;
21             cout<<"Input Y=";<<cin>>y;
22         }
23         void Output()
24         {
25             cout<<"X="<<x<<endl;
26             cout<<"Y="<<y<<endl;
27         }
28     };
29     class Test2:public Test1{
30     private:
31         int z;
32     public:
33         Test2()
34         {
35             Test1::Test1();
36             z=0;
37         }
38         Test2(int a,int b,int c)
39         {
40             x=a;
41             y=b;
42             z=c;
43         }
44         void Input()
45         {
46             Test1::Input();
47             cout<<"Input Z=";<<cin>>z;
48         }
49         void Output()
50         {
51             Test1::Output();
52             cout<<"Z="<<z<<endl;
53         }
54     };
55     int main()
56     {
57         Test2 obj2;
58         obj2.Input();
59         obj2.Output();
60     }
61 }
```



## ឧទាហរណ៍ ៤៖

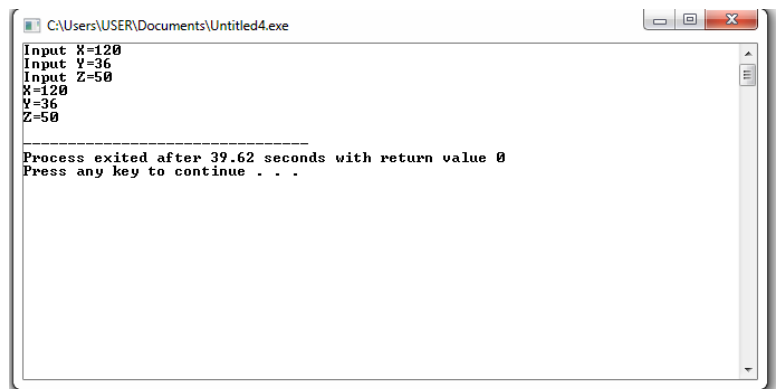
```
1  #include<iostream>
2  using namespace std;
3  class Test1{
4      protected:
5          int x;
6          int y;
7      public:
8          Test1()
9          {
10             x=0;
11             y=0;
12         }
13         Test1(int a,int b)
14         {
15             x=a;
16             y=b;
17         }
18         void Input()
19         {
20             cout<<"Input X=";<<cin>>x;
21             cout<<"Input Y=";<<cin>>y;
22         }
23         void Output()
24         {
25             cout<<"X="<<x<<endl;
26             cout<<"Y="<<y<<endl;
27         }
28     };
29     class Test2:public Test1{
30     private:
31         int z;
32     public:
33         Test2()
34         {
35             Test1::Test1();
36             z=0;
37         }
38         Test2(int a,int b,int c)
39         {
40             x=a;
41             y=b;
42             z=c;
43         }
44         void Input()
45         {
46             Test1::Input();
47             cout<<"Input Z=";<<cin>>z;
48         }
49         void Output()
50         {
51             Test1::Output();
52             cout<<"Z="<<z<<endl;
53         }
54     };
```

```

55 class Test3:private Test1{
56     private:
57         int z1;
58     public:
59         Test3()
60     {
61         Test1::Test1();
62         z1=0;
63     }
64         Test3(int a,int b,int c)
65     {
66         x=a;
67         y=b;
68         z1=c;
69     }
70     void Input()
71     {
72         Test1::Input();
73         cout<<"Input Z1=";cin>>z1;
74     }
75     void Output()
76     {
77         Test1::Output();
78         cout<<"Z1="<<z1<<endl;
79     }
80 };
81 int main()
82 {
83     Test3 obj3;
84     obj3.Output();
85     obj3.Input();
86     obj3.Output();
87 }

```

លទ្ធផលទទួលបាន៖



```

C:\Users\USER\Documents\Untitled4.exe
Input X=120
Input Y=36
Input Z=50
X=120
Y=36
Z=50

-----
Process exited after 39.62 seconds with return value 0
Press any key to continue . . .

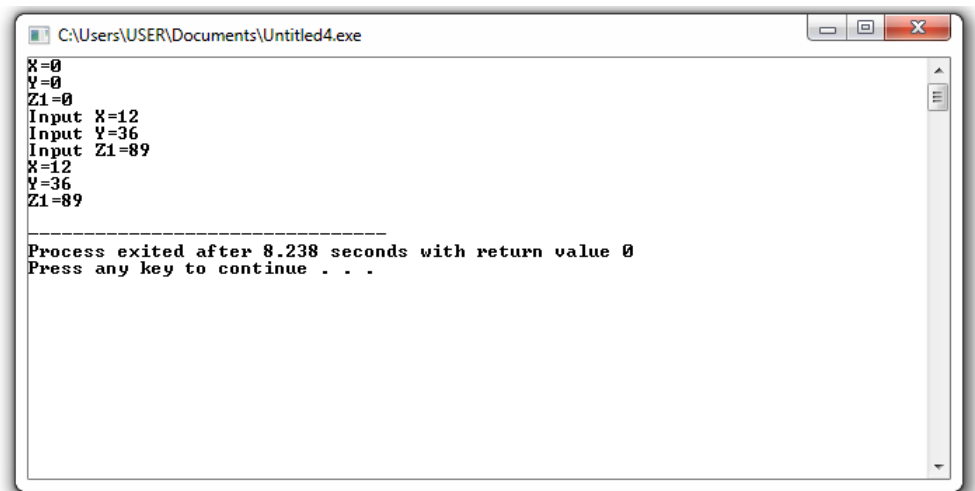
```



## ឧទាហរណ៍ ៥៖

```
1  #include<iostream>
2  using namespace std;
3  class Test1{
4      protected:
5          int x;
6          int y;
7      public:
8          Test1()
9          {
10             x=0;
11             y=0;
12         }
13         Test1(int a,int b)
14         {
15             x=a;
16             y=b;
17         }
18         void Input()
19         {
20             cout<<"Input X=";<<cin>>x;
21             cout<<"Input Y=";<<cin>>y;
22         }
23         void Output()
24         {
25             cout<<"X="<<x<<endl;
26             cout<<"Y="<<y<<endl;
27         }
28     };
29     class Test2:public Test1{
30     private:
31         int z;
32     public:
33         Test2()
34         {
35             Test1::Test1();
36             z=0;
37         }
38         Test2(int a,int b,int c)
39         {
40             x=a;
41             y=b;
42             z=c;
43         }
44         void Input()
45         {
46             Test1::Input();
47             cout<<"Input Z=";<<cin>>z;
48         }
49         void Output()
50         {
51             Test1::Output();
52             cout<<"Z="<<z<<endl;
53         }
54     };
```

```
55 class Test3:private Test2{
56     private:
57         int z1;
58     public:
59         Test3()
60         {
61             Test1::Test1();
62             z1=0;
63         }
64         Test3(int a,int b,int c)
65         {
66             x=a;
67             y=b;
68             z1=c;
69         }
70         void Input()
71         {
72             Test1::Input();
73             cout<<"Input Z1=";cin>>z1;
74         }
75         void Output()
76         {
77             Test1::Output();
78             cout<<"Z1="<<z1<<endl;
79         }
80     };
81     int main()
82     {
83         Test3 obj3;
84         obj3.Output();
85         obj3.Input();
86         obj3.Output();
87     }
```



```
C:\Users\USER\Documents\Untitled4.exe
X=0
Y=0
Z1=0
Input X=12
Input Y=36
Input Z1=89
X=12
Y=36
Z1=89

Process exited after 8.238 seconds with return value 0
Press any key to continue . . .
```



## នៅក្នុង Single Inheritance គេបែងចែកជា ពីរប្រភេទទៀត គឺ៖

១) Direct Class(Sub Class to direct Base Class)

២) InDirect Class(Sub Class to Sub Class to Base Class)

1.2. Multiple Inheritance: គឺជាប្រភេទ Inheritance មួយប្រភេទទៀតដែលគេអាចបង្កើតស្តង់ដារ Base Class ពីរ ឬច្រើន បន្ទាប់មក Sub Class មួយ ច្រើនអាចទាញយកទិន្នន័យពី Base Class ទំនើបបាន។

ឧទាហរណ៍ ១៖ ចូរបង្កើតស្តង់ដារ Base Class ពីរ និង Sub Class ចំនួន ១ដោយទាញទិន្នន័យពី Base Class ទាំងពីរ?

```

1  #include<iostream>
2  using namespace std;
3  class Test1{
4      protected:
5      int a;
6      int b;
7  };
8  class Test2{
9      protected:
10     float c;
11     float d;
12 };
13 class Test3:private Test1,private Test2{
14     public:
15     Test3()
16     {
17         a=0;
18         b=0;
19         c=0;
20         d=0;
21     }
22     Test3(int a1,int b1,float c1,float d1)
23     {
24         a=a1;
25         b=b1;
26         c=c1;
27         d=d1;
28     }
29     void Output()
30     {
31         cout<<"A="<<a<<endl;
32         cout<<"B="<<b<<endl;
33         cout<<"C="<<c<<endl;
34         cout<<"D="<<d<<endl;
35     }
36 };
37 int main()
38 {
39     Test3 obj3;
40     obj3.Output();
41     cout<<".....\n";
42     Test3 obj4(12,52,100,60);
43     obj4.Output();
44 }

```

```

C:\Users\USER\Documents\Untitled4.exe
A=0
B=0
C=0
D=0
.....
A=12
B=52
C=100
D=60
.....
Process exited after 0.01664 seconds with return value 0
Press any key to continue . . .

```

## លំហាត់អនុវត្ត

១) ចូរបង្កើត Class មួយឈ្មោះ Person ដែលមាន Data member ដូចជា ID(int), Name(String), Sex(String), DOB(String), Constructor ពីរគឺ Person(), Person(\_\_\_\_\_) និង Method ចំនួនពីរ ដូចជា void Input() និង void Output() បន្ទាប់មកបង្កើត object មួយ?

២) តពីលេខ១ ចូរបង្កើត class ចំនួនពីរទៀតគឺ Employee និង Students ហើយហៅ ទិន្នន័យពី Class Person មកប្រើបន្ទាប់មកបង្កើត Object នៃ Class ទាំងពីរ មកប្រើប្រាស់វា។

គេអោយ Base Class ដូចខាងក្រោម៖

```
1  #include<iostream>
2  using namespace std;
3  class Person{
4  protected:
5      int code;
6      string name;
7      string sex;
8      string dob;
9  public:
10     Person()
11     {
12         code=0;
13         name="N/A";
14         sex="N/A";
15         dob="dd/mm/yyyy";
16     }
17     Person(int i,string n,string s,string d)
18     {
19         code=i;
20         name=n;
21         sex=s;
22         dob=d;
23     }
24     void Input()
25     {
26         cout<<"Input ID=";<<cin>>code;
27         cout<<"Input name=";<<cin>>name;
28         cout<<"Input Sex=";<<cin>>sex;
29         cout<<"Input DOB=";<<cin>>dob;
30     }
31     void Output()
32     {
33         cout<<"ID="<<code<<endl;
34         cout<<"Name="<<name<<endl;
35         cout<<"Sex="<<sex<<endl;
36         cout<<"DOB="<<dob<<endl;
37     }
38 };
```

1.3. Hybrid Inheritance: គឺជាប្រភេទ នៃ Inheritance មួយបែបទៀតរបស់ OOP ក្នុង C++ ដែលវាអាចអនុញ្ញាតិអោយមានការទាញយកទិន្នន័យពី Base Class បន្តគ្នាពេលពី Base Class មួយទៅកាន់ Base Class មួយផ្សេងទៀត រហូតដល់ Derived Class។

```
1 Syntax:
2
3
4 class A
5 {
6     | .....
7 };
8 class B : public A
9 {
10    | .....
11 };
12 class C
13 {
14    | .....
15 };
16 class D : public B, public C
17 {
18    | .....
19 };
20
```

ឧទាហរណ៍ ១៖

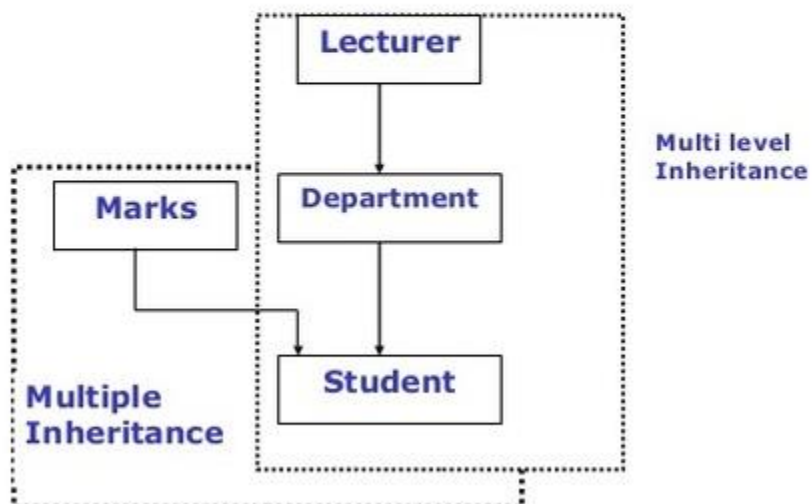
```
1 #include <iostream>
2 using namespace std;
3 class A
4 {
5     public:
6     int x;
7 };
8 class B : public A
9 {
10    public:
11    B()    //constructor to initialize x in base class A
12    {
13        x = 10;
14    }
15 };
16 class C
17 {
18    public:
19    int y;
20    C()    //constructor to initialize y
21    {
22        y = 4;
23    }
24 };
```

```
25 class D : public B, public C    //D is derived from class B and class C
26 {
27     public:
28     void sum()
29     {
30         cout << "Sum= " << x + y;
31     }
32 };
33
34 int main()
35 {
36     D obj1;                    //object of derived class D
37     obj1.sum();
38     return 0;
39 }                               //end of program
```

### លំហាត់អនុវត្ត

## Hybrid Inheritance

**Hybrid is nothing but the combination of  
Multilevel and multiple Inheritance**



Good Luck!