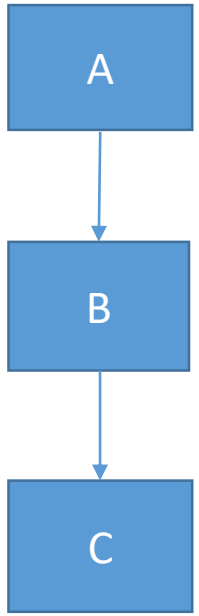
Multilevel Inheritance

- A class can be derived from another derived class.
- For example, class **A** serves as a base class for derived class **B**, which in turn serves as a base class for the derived class **C**.



- The class **B** is known as *intermediate* base class since it provides a link for the inheritance between **A** and **C**.
- The chain **ABC** is known as *inheritance path*.
- A derived class with multilevel inheritance is declared as follows:

```
class A{…..};
class B: public A{…..};
class C: public B{…..};
```

- This process can be extended to any number of levels.

```
#include<iostream>
using namespace std;
class student
{
  protected:
            int roll_number;
  public:
            void get_number(int);
            void put_number(void);
};
void student:: get_number(int a)
{
            roll_number=a;
}
void student::put_number()
{
            cout<<“Roll number: ”<<roll_number<<“\n”;
}
```

```
class test: public student
{
  protected:
                float sub1;
                float sub2;
  public:
                void get_marks(float, float);
                void put_marks(void);
};
void test :: get_marks(float x, float y)
{
  sub1 = x;
  sub2 = y;
}
void test:: put_marks()
{
 cout<<“Marks in SUB1 = “<< sub1<<“\n”;
 cout<<“Marks in SUB2 = “<< sub2<<“\n”;
}
```

```
class result: public test
{
  float total;
  public:
                void display(void);
};
void result::display(void)
{
   total = sub1 + sub2;
  put_number();
  put_marks();
  cout<<“Total = “<<total<<“\n”;
}
```

```
 int main()
 {
    result student1;
    student1.get_number(111);
    student1.get_marks(75.0, 59.5);
    student1.display();
    return 0;
}
```

OUTPUT

```
Roll Number: 111
Marks in SUB1 = 75
Marks in SUB2 = 59.5
Total = 134.5
```