

```
class Base1 {
public:
    Base1()
    { cout << " Base1's constructor called" << endl; }
};
class Base2 {
public:
    Base2()
    { cout << "Base2's constructor called" << endl; }
};
class Derived: public Base1, public Base2 {
public:
    Derived()
    { cout << "Derived's constructor called" << endl; }
};
int main()
{ Derived d; return 0; }
```

```
class Base1 {
public:
    ~Base1() { cout << " Base1's destructor" << endl; }
};
class Base2 {
public:
    ~Base2() { cout << " Base2's destructor" << endl; }
};
class Derived: public Base1, public Base2 {
public:
    ~Derived() { cout << " Derived's destructor" << endl; }
};
int main()
{
    Derived d;
    return 0;
}
```

```
class P {  
public:  
    void print() { cout <<" Inside P"; }  
};  
class Q : public P {  
public:  
    void print() { cout <<" Inside Q"; }  
};  
class R: public Q { };  
int main(void)  
{  
    R r;  
    r.print();  
    return 0;  
}
```

```
class Base {
private:
    int i, j;
public:
    Base(int i1 = 0, int j1 = 0): i(i1), j(j1) { }
};
class Derived: public Base {
public:
    void show(){
        cout<<" i = "<<i<<" j = "<<j;
    }
};
int main(void) {
    Derived d;
    d.show();
    return 0;
}
```

```
class Base
{
public:
    void fun() { cout << "Base::fun() called"; }
    void fun(int i) { cout << "Base::fun(int i) called"; }
};

class Derived: public Base
{
public:
    void fun() { cout << "Derived::fun() called"; }
};

int main()
{
    Derived d;
    d.fun(5);
    return 0;
}
```

```
class Base {
public:
    void fun()      { cout << "Base::fun() called"; }
    void fun(int i) { cout << "Base::fun(int i) called"; }
};

class Derived: public Base {
public:
    void fun() { cout << "Derived::fun() called"; }
};

int main() {
    Derived d;
    d.Base::fun(5);
    return 0;
}
```

```
class Base
{ public :
    int x, y;
    Base(int i, int j){ x = i; y = j; }
};
class Derived : public Base
{ public:
    Derived(int i, int j):x(i), y(j) {}
    void print() {cout << x <<" "<< y; }
};
int main(void)
{
    Derived q(10, 10);
    q.print();
    return 0;
}
```

```
class Base1
{ public:
    char c;
};
class Base2
{ public:
    int c;
};
class Derived: public Base1, public Base2
{ public:
    void show() { cout << c; }
};
int main(void)
{ Derived d;
  d.show();
  return 0;
}
```

```

class Base
{ protected:
    int a;
    public:
        Base() {a = 0;}
};

class Derived1: public Base
{ public:
    int c;
};

class Derived2: public Base
{ public:
    int c;
};

class Derived: public Derived1, public Derived2
{ public:
    void show() { cout << a; }
};

int main(void)
{
    Derived d;
    d.show();
    return 0;
}

```

```
class Base
{
    public:
    Base() { cout << "Base"; }
};
class Derived : public Base
{
    public:
    Derived(int i) { cout << i; }
};
int main()
{
    Derived d2(10);
    return 0;
}
```

```
class A
{
    public:
    int a;
    void change(int i)
    {   a=i; }
    void dis()
    { cout<<a; }
};
class B:public A
{
    int a=15;
    public:
    void print()
    { cout<<a; }
};
```

```
int main()
{
    B b1;
    b1.change(10);
    b1.print();
    b1.dis();
    return 0;
}
```

```
class A {  
    int data;  
public:  
    void f(int arg) { data = arg; }  
    int g() { return data; }  
};  
class B {  
public:  
    A x;  
};  
int main() {  
    B obj;  
    obj.f(20);  
    cout << obj.g() << endl;  
}
```

```
class B1 {  
public:  
    int i;  
    int j;  
    void g(int) { }  
};  
class B2 {  
public:  
    int j;  
    void g() { }  
};  
class D : public B1, public B2 {  
public:  
    int i;  
};
```

```
int main() {  
    D dobj;  
    D *dptr = &dobj;  
    dptr->i = 5;  
    dptr->j = 10;  
    dobj.g();  
}
```

```
class A {  
public: int x; };
```

```
class B {  
public: int y; };
```

```
class C: public A, virtual public B { };
```

```
class D: public A, virtual public B {  
    public:  int x;  
            int y;  
};
```

```
class E: public C, public D { };
```

```
int main() {  
    E e;  
    e.x = 1;  
    e.y = 2;  
}
```