Name:-Mizanul Hoque.
ID;-C221101.

Topic:-Inheritance.

///Single inheritance...

```
#include<bits/stdc++.h>
using namespace std;
class A
{
public:
   void show()
   {
      cout<<"Class A shoe"<<endl;
   }
};
class B : public A
{
public:
};
int main()
{
   B obj;
   obj.show();

}
```

///ambuguity.....

```
#include<bits/stdc++.h>
using namespace std;
class A
{
public:
   void show()
   {
      cout<<"Class A shoe"<<endl;
   }
};
class C
{
public:
   void show()
   {
```

```
        cout<<"Class C shoe"<<endl;
    }
};
class B : public A,public C
{
public:
    //show-A
    //show-C
};
int main()
{
    B obj;
    obj.show();

}
```

///multiple inheritance...

```
#include<bits/stdc++.h>
using namespace std;
class A
{
public:
    void show()
    {
        cout<<"Class A shoe"<<endl;
    }
};
class C
{
public:
    void show()
    {
        cout<<"Class C shoe"<<endl;
    }
};
class B : public A,public C
{
public:
    //show-A
    //show-C
};
int main()
{
```

```
    B obj;
    obj.A::show();
    obj.C::show();

}
```

/// ambugity(Multilevel inheritance).

```
#include<bits/stdc++.h>
using namespace std;
class A
{
public:
    void show()
    {
        cout<<"Class A shoe"<<endl;
    }
};
class B : public A
{
public:

};
class C : public A
{
public:

};
class D : public B,public C
{
public:

};

int main()
{
    D obj;
    obj.show();

}
```

///Multilevel inheritance(1)...

```
#include<bits/stdc++.h>
using namespace std;
class A
{
public:
   void show()
   {
      cout<<"Class A shoe"<<endl;
   }

};
class B : virtual public A
{
public:

};
class C : virtual public A
{
public:

};

class D : public B,public C///ekta class er jinish ekbar asar jonno virtual likte hoi..
{
public:                  ///(A<B<D)---(A<C<D)....ekta copy asbe....
                         ///jekono ekta asbe virtual likle...
                         ///virtual likle oo check korbe B<C same class teke inherite hoise kina..
                         ///same class teke inherit korle ekta copy asbe.

};

int main()
{
   D obj;
   obj.show();

}
```

///Multilevel inheritance(2)...

```
#include<bits/stdc++.h>
using namespace std;
class A
```

```
{
public:
   void show()
   {
      cout<<"Class A shoe"<<endl;
   }
};
class B : virtual public A
{
public:
   void showb()
   {
      ///eigula o cole jabe(D) te...
   }

};
class C : virtual public A///(class C : public virtual A)--->>eirkm likle o hbe...
{
public:
   void showc()
   {
      ///eigula o cole jabe(D) te...
      //cout<<"YYY"<<endl;
   }

};
class D : public B,public C///ekta class er jinish ekbar asar jonno virtual likte hoi..
{
public:                  ///(A<B<D)---(A<C<D)....ekta copy asbe....
                        ///jekono ekta asbe virtual likle...
                        ///virtual likle oo check korbe B<C same class teke inherite hoise kina..
                        ///same class teke inherit korle ekta copy asbe.

};

int main()
{
   D obj;
   obj.show();
   //obj.showc();

}
```

///inheritance default constructor...

```
#include<bits/stdc++.h>
using namespace std;

///BAse class(parent class)--->A
class A
{
public:
    A()
    {
        cout<<"Class A shoe"<<endl;
    }

};
///BAse class(parent class)--->B
class B
{
public:
    B()
    {
        cout<<"Class B shoe"<<endl;
    }

};
/// child class(derived class)-->C
class C : public A,public B
{
public:
    C()
    {
        cout<<"Class C shoe"<<endl;
    }

};

int main()
{
    C obj;
    ///inheritance er ketre aghe base class er constructor gula print hobe..
   /// eiketre jeta age ashe seta aghe print hobe...

   ///jodi eirkm hoi  ( class C : public B,public A)
   ///output :
   ///Class B shoe
```

```
   ///Class A shoe
  /// Class C shoe
  ///jodi eirkm hoi  ( class C : public A,public B)
   ///output :
   ///Class A shoe
   ///Class B shoe
  /// Class C shoe

}
```

///inheritance parameterrized constructor(1)...

```
#include<bits/stdc++.h>
using namespace std;

///BAse class(parent class)--->A
class A
{
public:
    A(int a)
    {
        cout<<"Class A shoe"<<endl;
        cout<<a<<endl;
    }

};
///BAse class(parent class)--->B
class B
{
public:
    B(int b)
    {
        cout<<"Class B shoe"<<endl;
        cout<<b<<endl;
    }

};
/// child class(derived class)-->C
class C : public A,public B
{
public:
    C (int value,int value2) : A(value),B(value2) ///A(1000),B(50) -->eibabe o value pass kora jabe
    {
        cout<<"Class C shoe"<<endl;
```

```
    }
//    C (int value,int value2) : A(100),B(20)
//    {
//        cout<<"Class C shoe"<<endl;
//    }


};

int main()
{
    C obj(10,100);

}
```

///inheritance parameterrized constructor(2)...

```
#include<bits/stdc++.h>
using namespace std;

///BAse class(parent class)--->A
class A
{
public:
    A(int a)
    {
        cout<<"Class A shoe"<<endl;
         cout<<a<<endl;
    }

};
/// child class(derived class)-->C
class C : public A
{
public:
    C(int value) : A(value)
    {
       // A obj(a);
        //cout<<c<<endl;
        cout<<"Class C shoe"<<endl;
    }

};
```

```
int main()
{
    C obj(10);

}
```