Topic 2 [Types of Inheritance]: Learn and Test different types of inheritance in C++. In

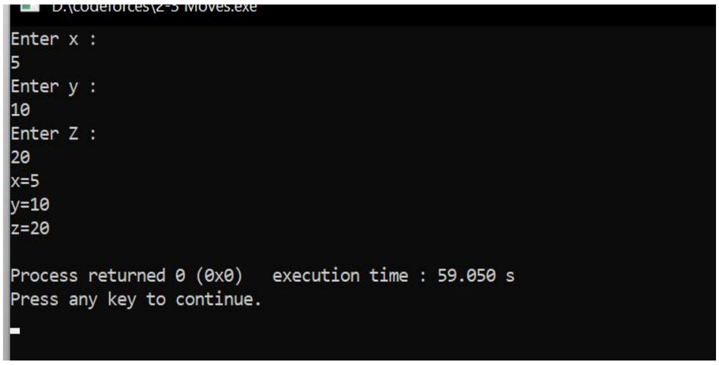
each inheritance draw the class diagram with class chain and try to access the data

members of bases classes from child classes.

1.Single inheritance:

```
#include<iostream>

using namespace std;

class A{

private:

   int x;

protected:

   int y;

public:

   int z;

   void setx(){

   cout<<"Enter x : "<<endl;

   cin>>x;

   }

   int getx(){

   return x;

   }

};

class B:public A{

public:

   void sety(){
   cout<<"Enter y : "<<endl;
   cin>>y;
   }
   void printx(){
   cout<<"x="<<getx()<<endl;
   }
   void printy(){
   cout<<"y="<<y<<endl;
   }

};
int main(){
B obj;
obj.setx();
obj.sety();
cout<<"Enter Z : "<<endl;
cin>>obj.z;
obj.printx();
obj.printy();
cout<<"z="<<obj.z<<endl;
}
```

Output:



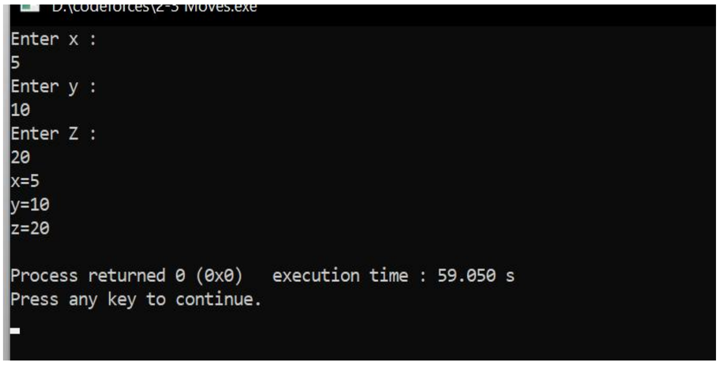
```
Enter x :
5
Enter y :
10
Enter Z :
20
x=5
y=10
z=20

Process returned 0 (0x0)   execution time : 59.050 s
Press any key to continue.
```

2.Multilevel inheritance:

Code:

```
#include<iostream>

using namespace std;

class A

{

private:

    int x;

protected:

    int y;

public:

    int z;

    void setx()

    {

        cout<<"Enter x : "<<endl;

        cin>>x;

    }

    int getx()
    {
        return x;
    }
};
class B:public A
{
public:
    void sety()
    {
        cout<<"Enter y : "<<endl;
        cin>>y;
    }
    int gety()
    {
        return y;
    }
};
class C:public B
{
public:
    void printx()
    {
        cout<<"x="<<getx()<<endl;
    }
    void printy()
    {
        cout<<"y="<<gety()<<endl;
    }
};
int main()
{
    C obj;
    obj.setx();
    obj.sety();
    cout<<"Enter z : "<<endl;
    cin>> obj.z;
    obj.printx();
    obj.printy();
    cout<<"z="<<obj.z<<endl;
}
```

Output:

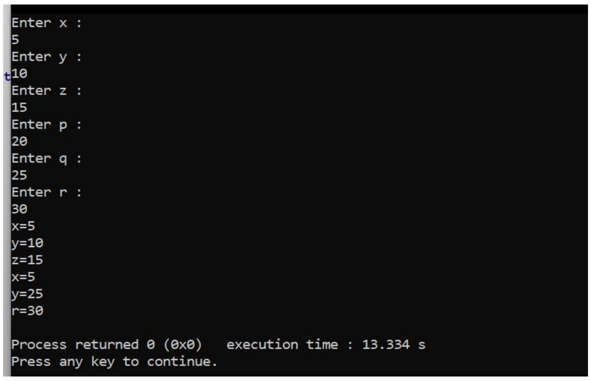


```
Enter x :
5
Enter y :
10
Enter Z :
20
x=5
y=10
z=20

Process returned 0 (0x0)   execution time : 59.050 s
Press any key to continue.
```

3.Multiple inheritance:

```
#include<iostream>

using namespace std;

class A

{

private:

    int x;

protected:

    int y;

public:

    int z;

    void setx()

    {

        cout<<"Enter x : "<<endl;

        cin>>x;

    }

    int getx()

    {

        return x;

    }

};


class B

{

private:
}

    int p;

protected:

    int q;

public:

    int r;

    void setp()

    {

        cout<<"Enter p : "<<endl;

        cin>>p;

    }

    int getp()

    {

        return p;

    }

};

class C:public A,public B

{

public:

    void sety()

    {

        cout<<"Enter y : "<<endl;

        cin>>y;

    }

    void setq()
    {
        cout<<"Enter q : "<<endl;
        cin>>q;
    }
    void printx()
    {
        cout<<"x="<<getx()<<endl;
    }
    void printy()
    {
        cout<<"y="<<y<<endl;
    }
    void printp()
    {
        cout<<"x="<<getx()<<endl;
    }
    void printq()
    {
        cout<<"y="<<q<<endl;
    }
};
int main()
{
    C obj;
    obj.setx();
    obj.sety();
    cout<<"Enter z : "<<endl;
    cin>>obj.z;
    obj.setp();
    obj.setq();
    cout<<"Enter r : "<<endl;
    cin>>obj.r;
    obj.printx();
    obj.printy();
    cout<<"z="<<obj.z<<endl;
    obj.printp();
    obj.printq();
    cout<<"r="<<obj.r<<endl;


}
```

Output:



```
Enter x :
5
Enter y :
10
Enter z :
15
Enter p :
20
Enter q :
25
Enter r :
30
x=5
y=10
z=15
x=5
y=25
r=30

Process returned 0 (0x0)   execution time : 13.334 s
Press any key to continue.
```

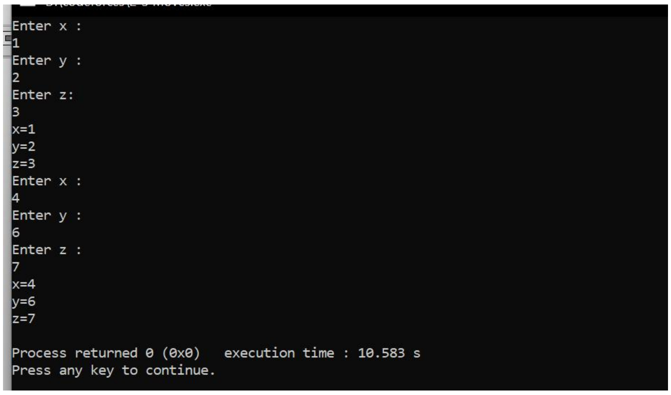
4.Hierarchical inheritance:

Code:

```
#include<iostream>

using namespace std;

class A

{

private:

   int x;

protected:

   int y;

public:

   int z;

   void setx()

   {

      cout<<"Enter x : "<<endl;

      cin>>x;

   }

   int getx()

   {

      return x;

   }

};

class B:public A

{
```

```
public:
    void sety()
    {
        cout<<"Enter y : "<<endl;
        cin>>y;
    }
    void printx()
    {
        cout<<"x="<<getx()<<endl;
    }
    void printy()
    {
        cout<<"y="<<y<<endl;
    }


};
class C:public A
{
public:
    void sety()
    {
        cout<<"Enter y : "<<endl;
        cin>>y;
    }
    void printx()
    {
        cout<<"x="<<getx()<<endl;
    }
    void printy()
    {
        cout<<"y="<<y<<endl;
    }


};
int main()
{
    B obj;
    obj.setx();
    obj.sety();
    cout<<"Enter z: "<<endl;
    cin>>obj.z;
    obj.printx();
    obj.printy();
    cout<<"z="<<obj.z<<endl;
    C obj2;
    obj2.setx();
    obj2.sety();

    cout<<"Enter z : "<<endl;

    cin>>obj2.z;

    obj2.printx();

    obj2.printy();

    cout<<"z="<<obj2.z<<endl;

}
```

Output :



```
Enter x :
1
Enter y :
2
Enter z:
3
x=1
y=2
z=3
Enter x :
4
Enter y :
6
Enter z :
7
x=4
y=6
z=7

Process returned 0 (0x0)   execution time : 10.583 s
Press any key to continue.
```

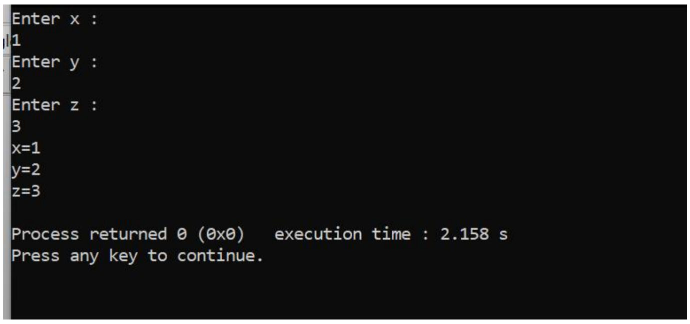
5.Hybrid inheritance:

Code:

```
#include<iostream>

using namespace std;

class A

{

private:

    int x;

protected:

    int y;
```

```
public:
    int z;
    void setx()
    {
        cout<<"Enter x : "<<endl;
        cin>>x;
    }
    int getx()
    {
        return x;
    }
};

class B:virtual public A
{


};
class C:virtual public A
{


};
class D:public B,public C
{
public:
    void sety()
    {
        cout<<"Enter y : "<<endl;
        cin>>y;
    }

    void printx()
    {
        cout<<"x="<<getx()<<endl;
    }
    void printy()
    {
        cout<<"y="<<y<<endl;
    }

};
int main()
{
    D obj;
    obj.setx();
    obj.sety();
    cout<<"Enter z : "<<endl;
    cin>>obj.z;
    obj.printx();
    obj.printy();

    cout<<"z="<<obj.z<<endl;

}
```

Output :



```
Enter x :
1
Enter y :
2
Enter z :
3
x=1
y=2
z=3

Process returned 0 (0x0)   execution time : 2.158 s
Press any key to continue.
```