

Aim:

To understand the concept of Hybrid Inheritance and implement it with the help of an example.

Theory:

One of the most important concepts in object-oriented programming is that of inheritance. Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and fast implementation time. When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the base class, and the new class is referred to as the derived class. Hybrid Inheritance is a type of inheritance. It is implemented by combining more than one type of inheritance. For example: Combining Hierarchical inheritance and Multiple Inheritance.

Syntax :

```
class A {  
  
};  
class B: public A {  
  
};  
class C: {  
  
};  
class D: public C public B {  
  
};
```

Code :

```
#include<iostream>  
using namespace std;  
  
class A {  
    protected:  
        int a , b ;  
    public:  
        void getab() {  
            cout<<"Enter a and b value : ";  
            cin>>a>>b;  
        }  
};  
class B:public A {  
    protected:  
        int c ;  
    public:  
        void getc() {  
            cout<<"Enter c value : ";
```

```

        cin>>c;
    }
};
class C {
protected:
    int d ;
public:
    void getd() {
        cout<<"Enter d value : ";
        cin>>d;
    }
};
class D:public B,public C {
protected:
    int e ;
public:
    void result() {
        geta();
        getb();
        getc();
        getd();
        e=a+b+c+d;
        cout<<"Addition is : "<<e<<endl;
    }
};
int main() {
    D d1;
    d1.result();
    return 0;
}

```

Output :

```

PS D:\College\OOPS> .\hybrid
Enter a and b value : 12 14
Enter c value : 12
Enter d value : 12
Addition is : 50

```

Discussion :

In the above program there are four classes which are showing hybrid inheritance. Classes A and B show multilevel and classes B,C and D show multiple inheritance. So it is a type of hybrid inheritance having a combination of multilevel and multiple inheritance. However, the invoking method is same as of previous program.

Learning Outcomes :

Inheritance is the major feature of object oriented programming we learn the following terms from inheritance: -

- Reusability : Facility to use public methods of base class without rewriting the same
- Extensibility : Extending the base class logic as per business logic of the derived class

- Data hiding : Base class can decide to keep some data private so that it cannot be altered by the derived class
- Overriding : With inheritance, we will be able to override the methods of the base class so that meaningful implementation of the base class method can be designed in the derived class.