

Subject Name: Object Oriented Programming Using C++

Subject Code: BCA-301 N

Subject Topic: Hybrid Inheritance and Virtual Base Class

Abhishek Dwivedi

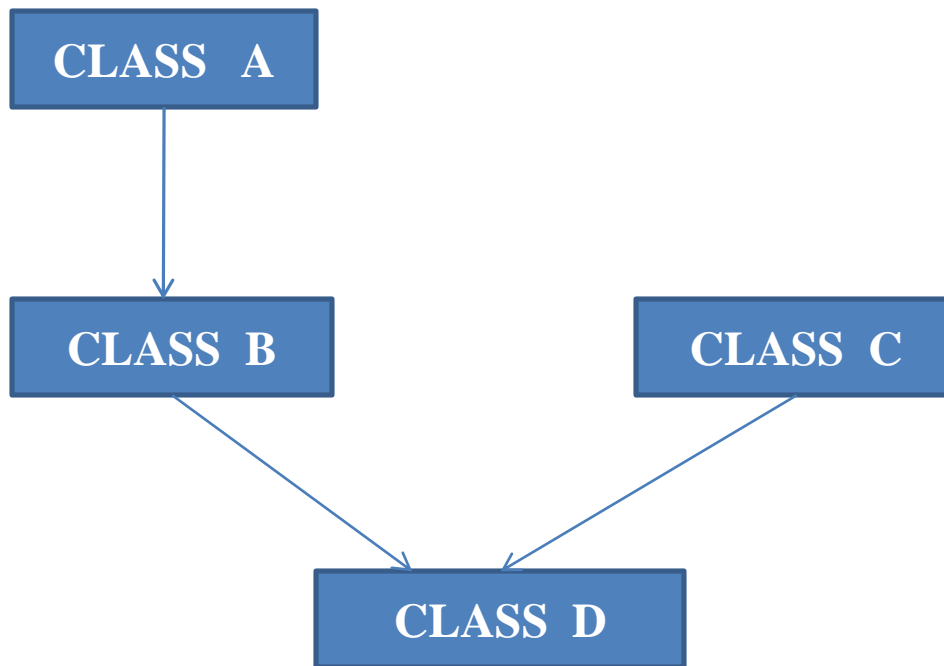
Assistant Professor

Department of Computer Application

UIET, CSJM University, Kanpur

Hybrid Inheritance

- Hybrid inheritance is a combination of Single, Multiple, Hierarchical and Multilevel type of inheritance.



Example

```
class A
```

```
{  
    protected:  
    int a;  
    public:  
    void get_a()  
    {  
        cout << "Enter the value of 'a' : " <<endl;  
        cin>>a;  
    }  
};
```

```
class B : public A
```

```
{
```

```
    protected:
```

```
    int b;
```

```
    public:
```

```
    void get_b()
```

```
{
```

```
    cout << "Enter the value of 'b' : " <<endl;
```

```
    cin>>b;
```

```
}
```

```
};
```

```
class C
```

```
{
```

```
    protected:
```

```
    int c;
```

```
    public:
```

```
    void get_c()
```

```
    {
```

```
        cout << "Enter the value of c is : " <<endl;
```

```
        cin>>c;
```

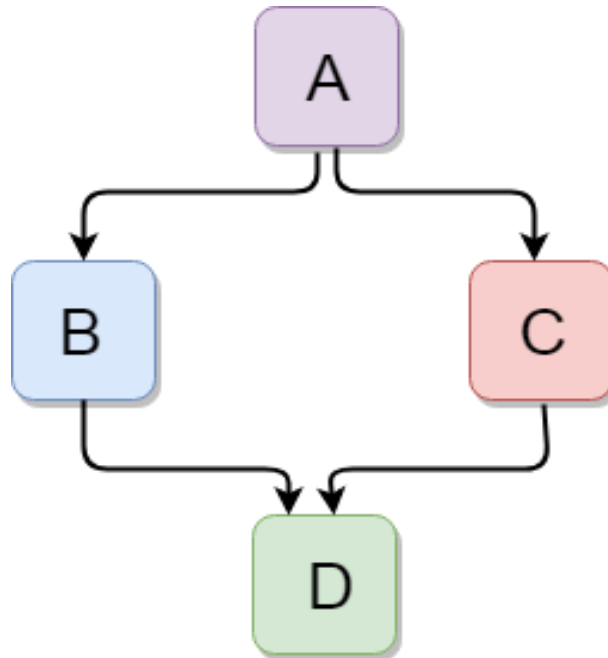
```
    }
```

```
};
```

```
class D : public B, public C
{
    protected:
    int d;
    public:
    void mul()
    {
        get_a();
        get_b();
        get_c();
        cout << "Multiplication of a,b,c is : " <<a*b*c<<endl;
    }
};
int main()
{
    D obj;
    obj.mul();
    return 0;
}
```

Virtual base class

- Virtual base classes are used in virtual inheritance in a way of preventing multiple “instances” of a given class appearing in an inheritance hierarchy.



```
class A {  
public:  
    void show()  
    {  
        cout << "Hello form A \n";  
    }  
};
```

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

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

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

```
int main()  
{  
    D obj;  
    obj.show();  
    return 0  
}
```


- To resolve this ambiguity when class **A** is inherited in both class **B** and class **C**, it is declared as **virtual base class** by placing a keyword **virtual** as :

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

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

```
class A {  
public:  
    int a;  
    A()  
    {  
        a = 10;  
    }  
};
```

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

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

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

```
int main()  
{  
    D obj;  
    cout << "a = " << obj.a << endl;  
  
    return 0;  
}
```

References:

- www.studytonight.com
- www.tutorialpoint.com
- www.geeksforgeeks.org
- “Object oriented programming in C++” Robert Lafore
- “Object oriented programming with C++”, E.Balagurusamy