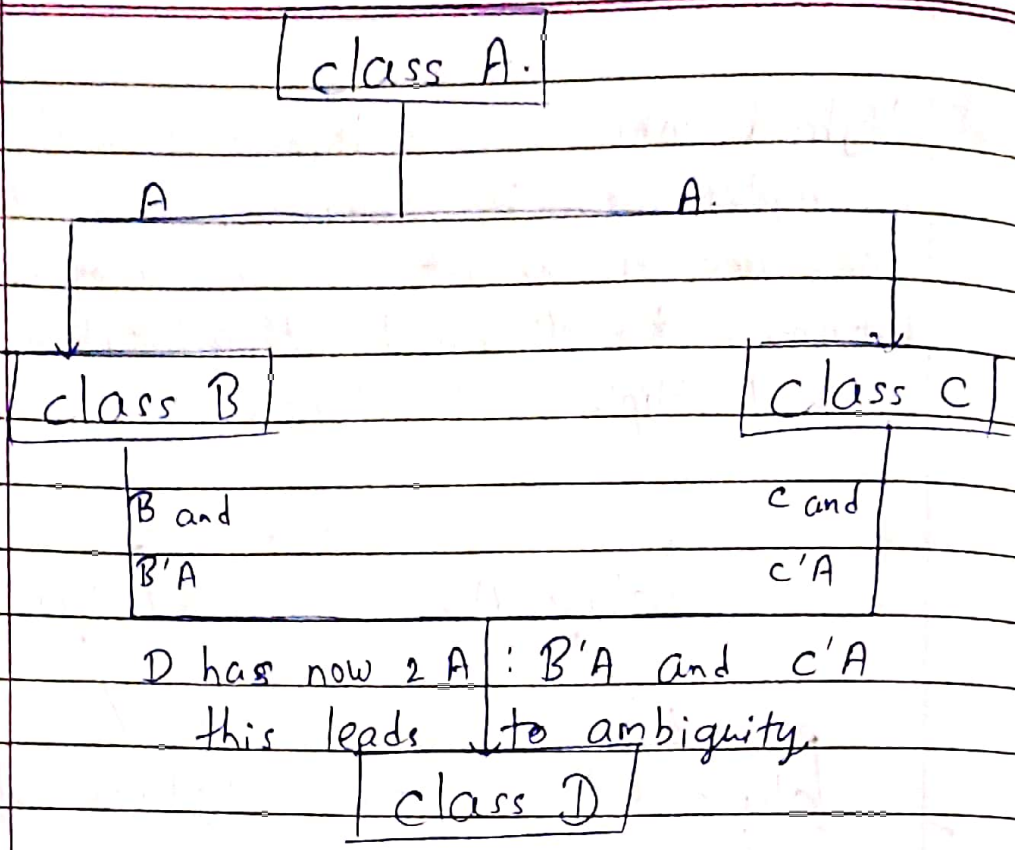
Virtual base class :-

Virtual base classes are used in hybrid (virtual) inheritance in a way of preventing multiple "instances" of a given class appearing in an inheritance hierarchy when using multiple inheritance.

Need for virtual base classes :-

consider the situation where we have one class A. this class A is inherited by two other classes B and C. Both these classes are inherited into another in a new class D as shown in figure



As we can see from the figure that data members/function of class A are inherited twice to class D. One through class B and second through class C. When any data member/function of class A. is accessed by an object of class D. ambiguity arises as to which data member/function would be called? One inherited through B or the other inherited through C. this confuses compiler and it displays error.

Example :- To show the need of virtual base classes in C++.

```
#include <iostream>
using namespace std;

class A {
    public:
        void show() {
            cout<<" Hello from A "<<endl;
        }
};

class B : public A {
};

class C : public A {
};

class D : public B, public C {
};

int main() {
    D object;
    object.show();
}
```

O/P :- compiler Error (ambiguity).

Date: YOUVA

How to resolve this issue?
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 :-

syntax for virtual Base classes :-
Syntax 1 :-
class B : virtual public A {
};

Syntax 2 :-
class C : ~~virtual~~ public virtual A {
};

Note : virtual can be written before or after the public.

Now only one copy of data / function member will be copied to class C and class B and class A becomes the virtual base class. virtual base classes offer a way to save space and avoid ambiguities in class hierarchies that use multiple inheritances. When a base class is specified as a virtual base, it can act as an indirect base more than once without duplication of its data members. A single data members is shared by all the

Page No.:
Date:
youva

base classes that use virtual base.

```
#include <iostream>
using namespace std;

class A {
    public :
        void show() {
            cout << " Hello from A" << endl;
        }
};

class B : public virtual A {
};

class C : public virtual A {
};

class D : public B, public C {
};

int main() {
    D object;
    object.show();
}
```

O/P :- Hello From A.