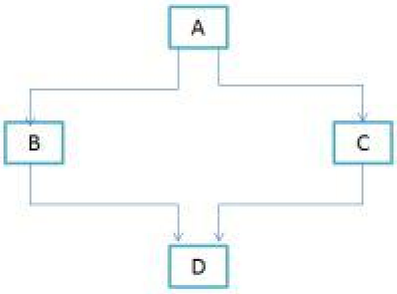
Does Java support Multiple inheritance?

Earlier we discussed types of inheritance. In this post we will discuss why java doesn't support **multiple inheritance**. We would also see how to achieve this using interfaces in Java.

Why Java doesn't support multiple inheritance?

C++ , Common lisp and few other languages supports multiple inheritance while java doesn't support it. It is just to **remove ambiguity**, because **multiple inheritance** can cause ambiguity in few scenarios. One of the most common scenario is **Diamond problem.**

What is diamond problem?
Consider the below diagram which shows multiple inheritance as Class D extends both Class B & C. Now lets assume we have a method in `class A` and `class B` & `C` overrides that method in their own way. **Wait!! here the problem comes** – Because D is extending both B & C so if D wants to use the same method which method would be called (the overridden method of B or the overridden method of C). Ambiguity. That's the main reason why Java doesn't support multiple inheritance.



How to achieve multiple inheritance in Java using interfaces?

```
interface X
{
   public void myMethod();
}
interface Y
{
   public void myMethod();
}
class Demo implements X, Y
{
   public void myMethod()
   {
       System.out.println(" Multiple inheritance example using interfaces");
   }
}
```

As you can see that the class implemented two interfaces. A class can implement any number of interfaces. In this case there is no ambiguity even though both the interfaces are having same method. Why? Because methods in an interface are always abstract by default, which doesn't let them to give their implementation (or method definition) in interface itself.