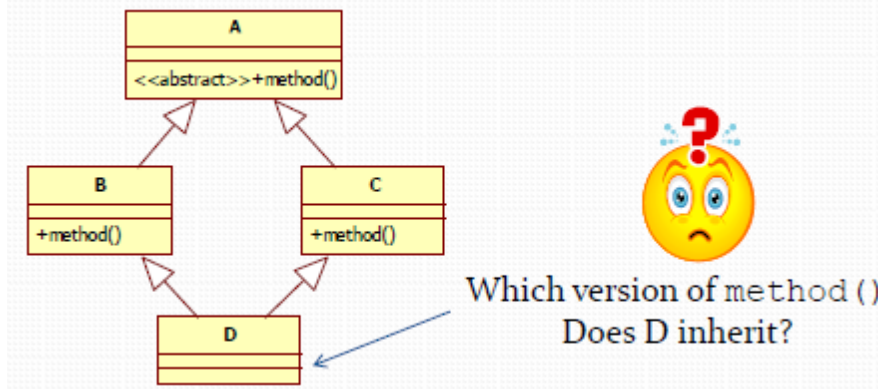


- E. The Diamond Problem that must be handled by any language that supports multiple inheritance. Java SE 8 now supports “behavioral” multiple inheritance (but not “data” multiple inheritance). Explain how features of Java 8 handle the Diamond Problem by considering two scenarios:



I. **When the type D is a class:**

Duplicate default methods named method are inherited from the types B and C. Therefore, class D should override the method or override using B or C’s method. Otherwise class D does not know which default method should implement. For example:

```
public class classD implements InterfaceB, InterfaceC {
    @Override
    public int method(int x) {
        return InterfaceB.super.method(x);
    }
}
- OR -
```

```
public class classD implements InterfaceB, InterfaceC {
    @Override
    public int method(int x) {
        return x+1;
    }
}
```

II. **When the type D is an interface:**

Duplicate default methods named method are inherited from the types B and C. Implementation or redeclaration is required for type D because there are multiple method with same signature. For example:

```
public interface InterfaceD extends InterfaceB, InterfaceC {
    //override default method in InterfaceB
    @Override
    default int method(int x) {
        return InterfaceB.super.method(x);
    }
}
- OR -
```

```
public interface InterfaceD extends InterfaceB, InterfaceC {
    //implementation required
    abstract int method(int x);
}
```