

Assignment – Interface

1] **Because we** don't have the correct access modifiers for the methods. in the interface , the method is implicitly public , so we need to make sure the implementation in the class is also public.

Code with corrections:-

```
interface X {
    void methodM1 ();
}
class y implements X{
    public void methodM1 () {
        System.out.println("This is meththodM1");
    }
}
```

2] No, the provided code will not compile successfully. Because When a class implements an interface with constants, it implicitly becomes **static and final**. Therefor the constant `int I = 111;` in the interface can not be overridden or modified in the implementing class.

3]No , because class extends another class only and interface can extends multiple interfaces .

4] 1--> QQQQPPPP

2-->PPPPQQQQ

5]Yes , the above program will run correctly

Output:- 4

6] **Interface:-**

→ Before java 8 an interface was defined as the collection of public abstract methods.

→ From java 8 an interface is not just a collection of public abstract methods rather it also contains non-abstract methods like default and static methods.

→ We can not create object of an interface directly. But we create object of an interface using anonymous inner types.

7] Only public abstract and default access modifiers are allowed in interface. When we give private and protected modifiers to the methods interface it gives compilation error.

8] Though we can not create the object of an interface directly. We can certainly create the reference directly. we create object of an interface using anonymous inner types.

9] In java one interface can not implement another interface. Because interface contains unimplemented methods. In java we can extend one interface from another interface.