

We know that we can not have multiple inheritance like this. Where we have class A and class B, than class C trying to inherit from both two classes A and B.

Also we have two same method in both classes A and B

So when we make an object of class C, as: C c = new C( );

Then when we do c.fun( )

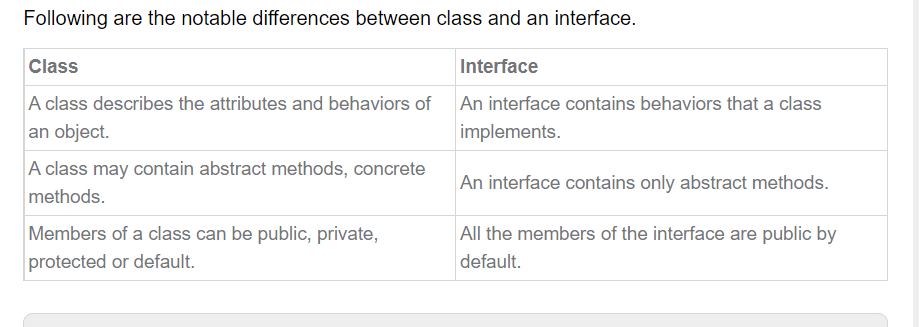
Java will get confused which fun ( )to run

To solve this we have something called interfaces

Interfaces provides absolute abstraction

It specifies what a class must do but it does not specifies how it should do, it does not provides the implementation aspect of functions.

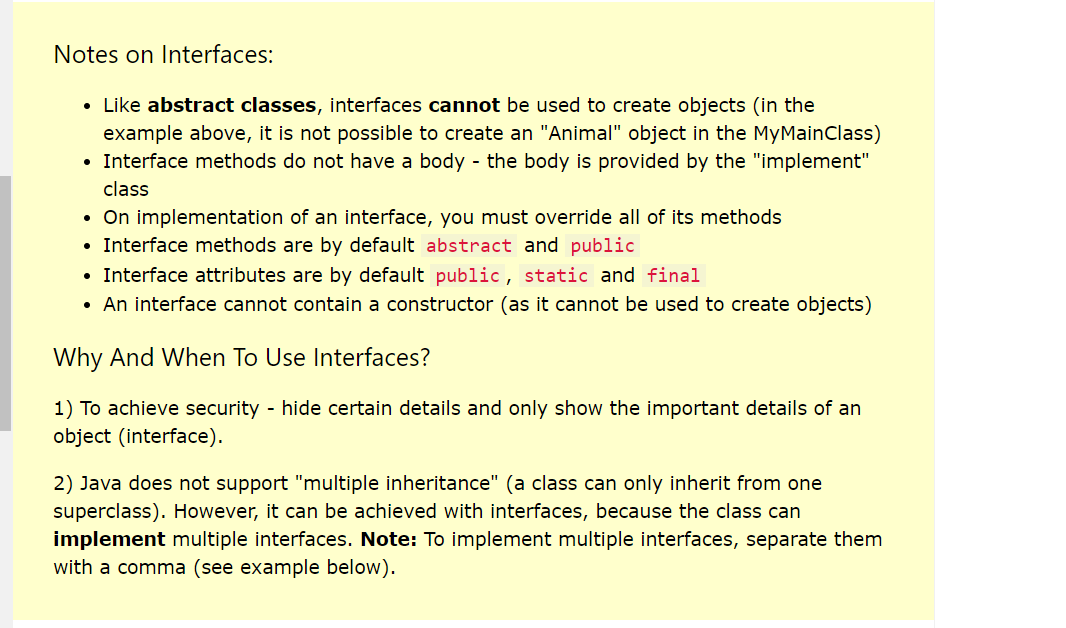
The implementation of functions will be specified by the child classes by overriding the abstract functions of the interfaces

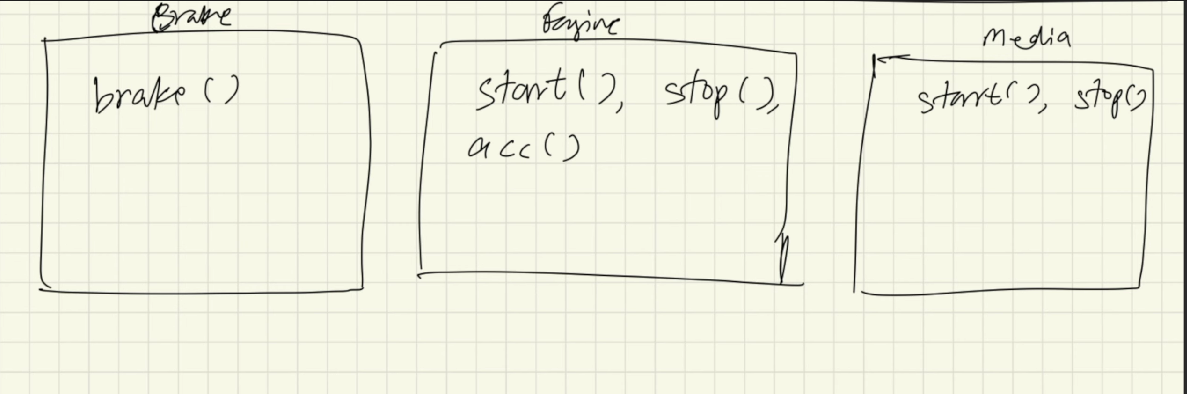


Some rules

1. An interface can only have abstract methods, and all the methods are public by default
2. An interface has no constructor, we cannot make objects
3. An interface only has static variables and those variables are also public and final, it makes sense because even if we have non static variables there is no any way of instantiating those variables since interfaces does not have constructors and they only have abstract methods, and abstract methods do not have body and implementation part (since, they will be overridden by the child classes and they will get their body and implementation part there). So, variables are static in interfaces
4. Generally, interfaces are famous for grouping all the essential methods, without bothering about the implementation part inside one entity. Variables are not that widely seen inside any interfaces

But if there needed to put any variable that is public, static and final by default. Variable is final to ensure that it is not changed over the course of program by anything

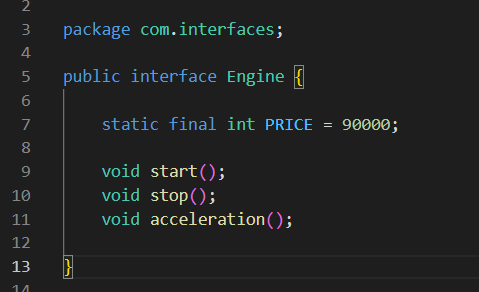


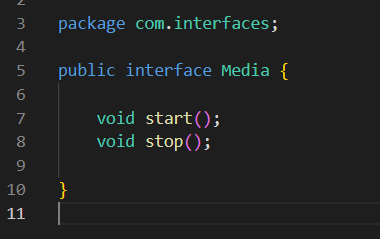


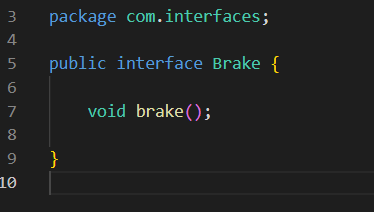
Let say we are creating a car and that car needs to have all these three things in it

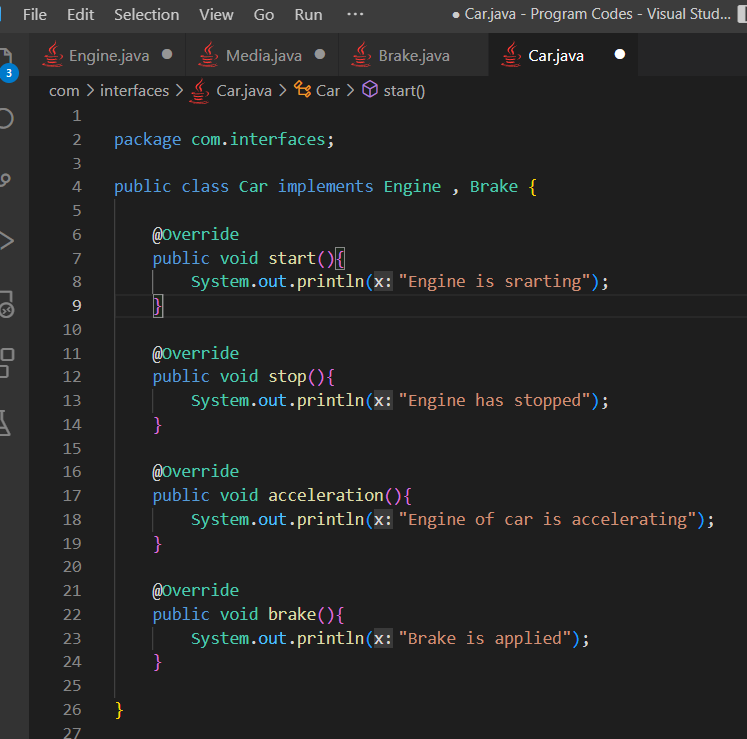
Now multiple inheritance is not allowed in java

So we have interfaces in java







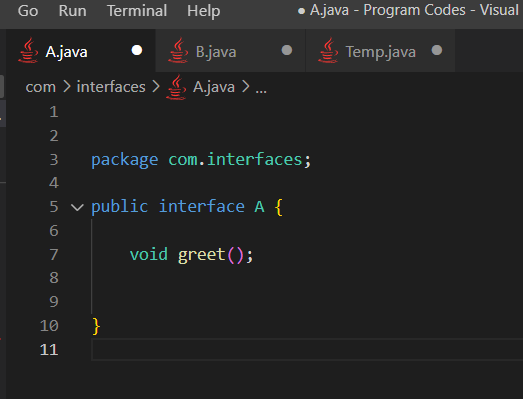


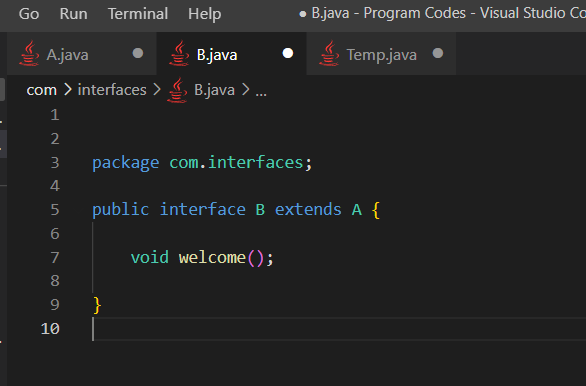
See here we are kind of inheriting from all the interfaces using implements keyword, also we are overriding the methods defined in the interfaces here in the child Car class

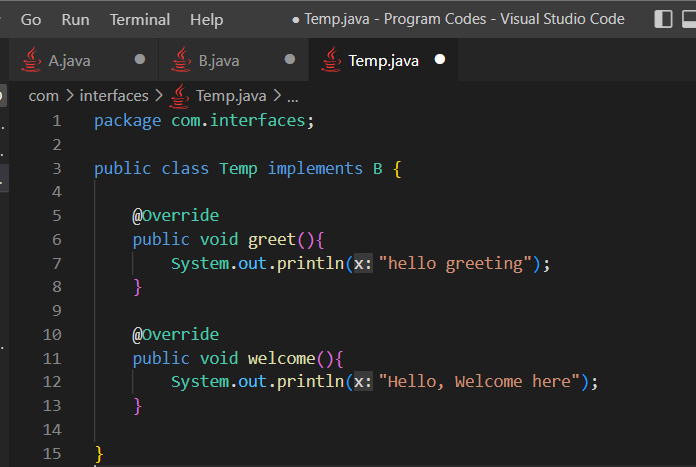
**Actually, this way of working with the interfaces is actually really poor, weak and bad practice. We generally have a separate class for a interface or we create a class for an interface. See the interfaces folder here in this folder**

**Facilitating multiple inheritance is just one of the uses of interfaces, there are mountains of uses of interfaces, if dive deep in to it, see the bookmarkes posts in stack exchange and stack overflow?**

We can also extend the interfaces

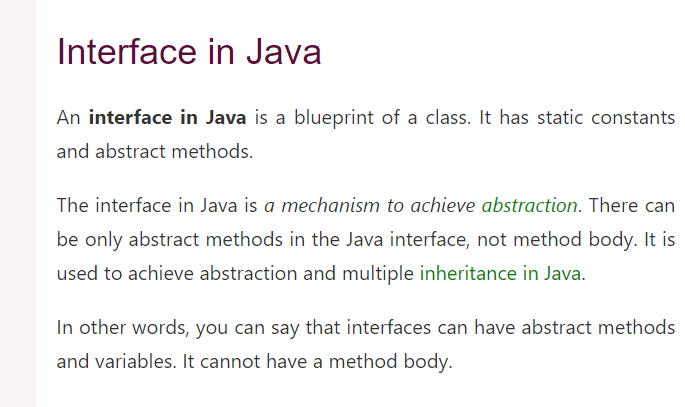




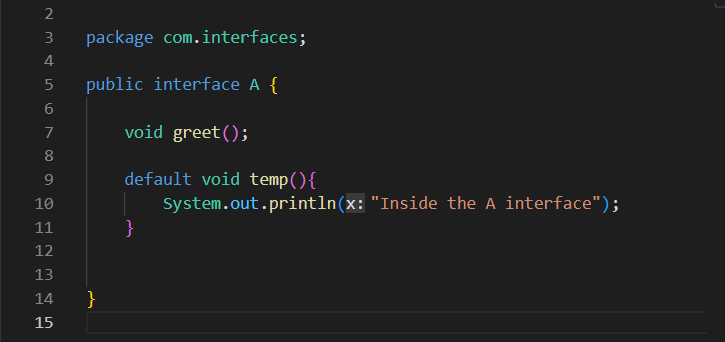


So here we need to override both the methods present in both the interfaces because B extends A

**Class is a blur print for objects, and it seems like Interface is kind of blue print for classes**

****

Starting from JDK 8 we can actually define a function with all it’s body in an interface via default keyword



But I recommend not using this