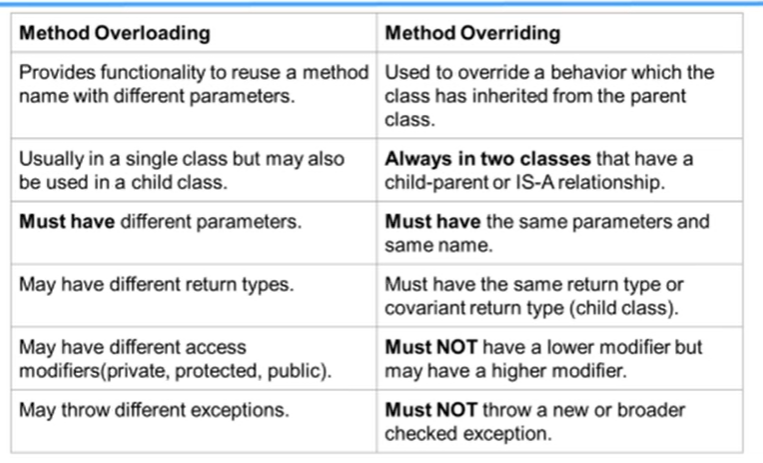
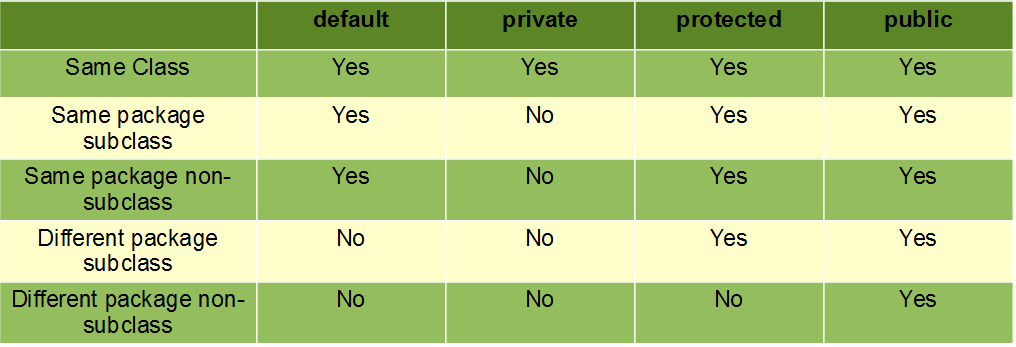
@ Method Overloading / Overriding





*Q) as we know we can’t override static methods only instance methods can override. So, What is instance method and static method?*

Ans>

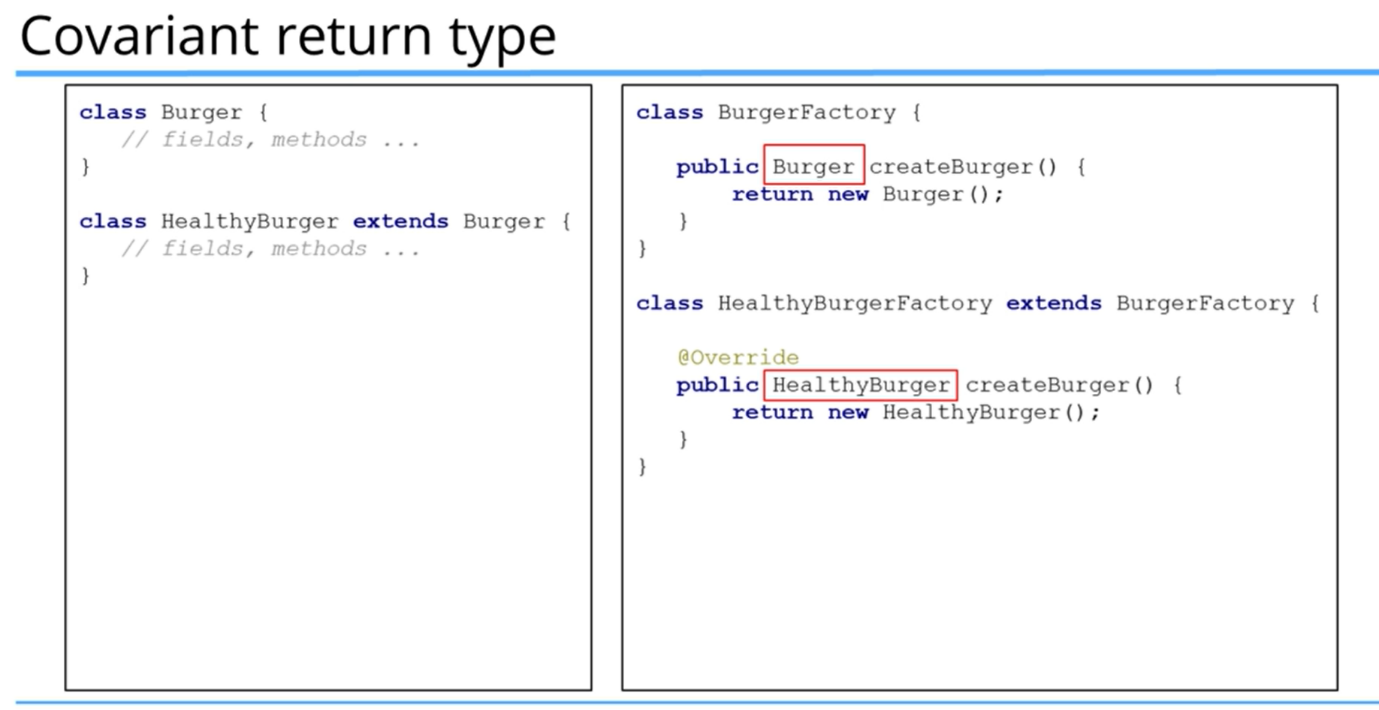
instance methods are methods that belong to a specific instance of a class whereas static methods  belong to the class itself (and are declared using the modifier **'static'**) .

To use an instance method you have to first instantiate the class and access it throught the object you have created : **objectName.instanceMethodName() ;**

To access static methods ,you refer to the class where they are created , in conjonction with the name of the static method : **ClassName.staticMethodName() ;**

I hope that was clear enough .

*Q) Explain Covariant return type?*



Ans>

Overriding methods only work in classes with inheritance.  Since class HealthyBurger extends Burger {, Burger is the superclass and HealthyBurger is the subclass.  If they both have a method with the same name and parameters, in this case createBurger(), then the methods are overriding.  The returns can be the same in both methods.  Or the return in the subclass (HealthyBurger) may be a subclass type of the return (Burger) in the superclass method (called a covariant return).

The return new HealthyBurger(); is an anonymous return because it does not have a reference.  It just instantiates a HealthyBurger object using the HealthyBurger constructor.  Since the constructor is not explicitly typed into the class code, the JVM will supply a no-arg constructor which is the empty parentheses ( ) at the end.  A constructor also has ( ) at the end because it is a special type of method.