Static, final keywords on methods

Inheritance

Method overloading

1. Method signature should be different(method name+ arguments) where name should be same as per concept
2. Change in no. of argument

Type of arguments

1. Order of arguments
2. Return type may be same or not
3. Access specifier may be same or different
4. Overloading done at compile time
5. Overloading with in the class
6. Compile time polymorphism

Method Overriding

1. Method signature should be same.

Same name+arguments(type,number,order)

1. Return types are primitives then they should be same
2. Return type is reference variable then use same return type mention for method as in super class or use sub classes reference of return type of method mention in parent class
3. Overriding between the classes which has inheritance.
4. Run time polymorphism.
5. Access specifier may be same or higher scope (private<default<protected<public)

Access specifier:

Where all these are applied

Variables methods classes

Public yes yes yes

Protected yes yes no

[Default] yes yes yes

Private yes yes no

Scope:

Public protected default private

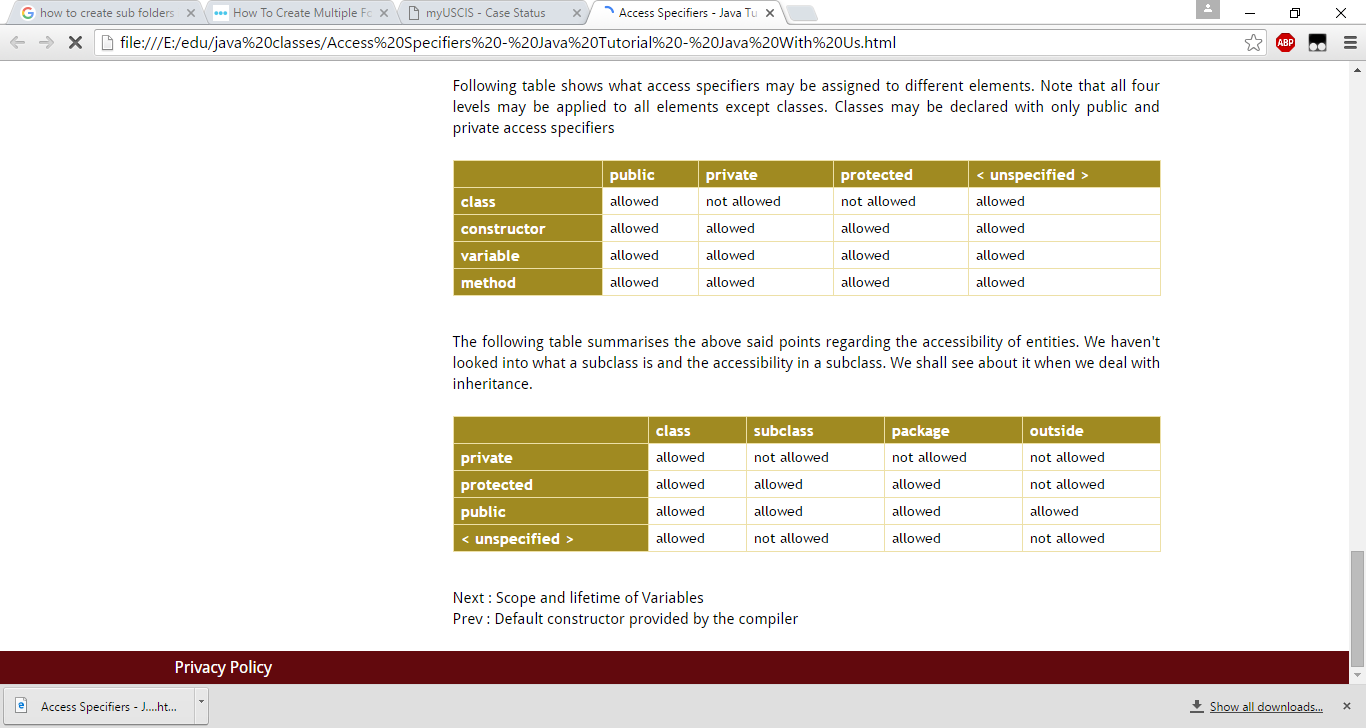
With in the class same package yes yes yes yes

With int different class in same package yes yes yes no

With in derived class in same package yes yes yes no

With in the class outside package yes no no no

With in derived class outside package yes yes no no



Constructors:

1. Same as class name
2. No return type/not even void
3. Special type of method that handled by jvm
4. If there is no constructor in the class then default constructor (constructor without arguments) is created by JVM.
5. If a constructor is created by user then jvm consider it as default constructor.
6. If a constructor with argument based is added then JVM doesn’t add default constructor.

Super:

1. Jvm automatically insert super() method as first statement until we mention any super statement
2. If we mention super then jvm doesn’t add any super to it.
3. The super statement is added at compile time.
4. It is used to call super class (parent class) constructor.

Where the keywords applied on variables

Instance local

Static yes no

Final yes yes

Public yes no

Private yes no

Protected yes no

unspecified yes yes

**Exercise:**

1. Write switch case for display mon to sun based on entered numbers 1 to 7 and other than this return wrong number entered
2. Html page exercise
3. Multilevel with passing 4 data types and calling super with datatype values 2 to parent another 2 to its parent