OOP using Java

Trainer: Mr. Rohan Paramane

Static Field & its Initialization

- **Static Field**
  - Static field do not get space inside instance rather all the instances of same class share single copy of it.
  - Static Field is also called as class variable. It gets space once per class.
  - Static Field gets space once per class during class loading on method area .
  - Static Field can be accessed using object reference but it is designed to access using class name and dot operator.
- **Static Initializer block**
  - It is used to initialize the static fields
  - A static initialization block is a normal block of code enclosed in braces, { }, and preceded by the static keyword. Here is an example: static { // code to write }
  - A class can have any number of static initialization blocks, and they can appear anywhere in the class body.
  - The runtime system guarantees that static initialization blocks are called in the order that they appear in the source code.

Static Method

- To access non static members of the class, we should define non static method inside class.
- Non static method/instance method is designed to call on instance.
- To access static members of the class, we should define static method inside class.
- static method/class level method is designed to call on class name.
- static method do not get this reference:
- If we call, non static method on instance then method get this reference.
- Static method is designed to call on class name.
- Since static method is not designed to call on instance, it doesn't get this reference.

Final

- In java we do not get const keyword. But we can use final keyword.
- After storing value, if we don’t want to modify it then we should declare variable final.
- We can provide value to the final variable either at compile time or run time.
- once initialized, if we don't want to modify state of any field inside any method of the class (including constructor body) then we should declare final field.
- If we want to declare any field final then we should declare it static also.
- In Java, we can declare reference final.
- But we can not declare instance final.
- We can declare method final. It is not allowed to override final method in sub class.
- We can declare class final. It is not allowed to extend final class.

Static Import

- If static members belonging to the different class then use of type name and dot operator is mandatory.
- There are situations where you need frequent access to static final fields (constants) and static methods from one or two classes.
- Prefixing the name of these classes over and over can result in cluttered code.
- The static import statement gives you a way to import the constants and static methods that you want to use so that you do not need to prefix the name of their class.

Thank you!

Rohan Paramane

rohan.paramane@sunbeaminfo.com