Repo Created: <https://github.com/perakampraveen2003/java-practice.git>

Static: Implies static member will be shared among all instances of the class, so we will only create one instance of it that can be shared across all objects.

It can be act as a variable, method, block, and a class.

1. Static as Variable:
   1. The static variable gets memory only once in the class area at the time of class loading.
   2. This can be referred to as common property of all objects.
2. Static as Method:
   1. Static method belongs to the class rather than the object of a class. This method will call with class reference with creating instance.
   2. Static method uses only static context which means non static members cannot be accessed inside static method.
   3. This and super cannot be called in static context.
3. Static as block:
   1. Is used to initialize the static data member.
   2. It is executed before the main() at the time of class loading.

\*Handson Required\*

Git Link:

Abstraction: It’s a process of hiding the method implementation.

We can achieve this in two ways:

1. Abstract class.
2. Interface.

Abstract class:

1. When a class is declared with abstract keyword then it acts as Abstract class.
   1. abstract class AbstractDemo {}
2. It can hold both abstract and non-abstract methods.
   1. Abstract methods implies when a method is declared as abstract and does not has any implementation.
   2. Non-abstract methods imply usual java methods where it has its own implementation.
3. Abstract classes cannot be instantiated.
4. It can have constructors, static context and final methods (it will force the subclass not to change the final method definition)

Abstraction Handson: Must cover abstract method, non-abstract method, constructor usage and static usage as well.

Git Link:

Interface:

1. It’s used to achieve abstraction and support multiple inheritance.
2. It cannot be instantiated like abstract class.
3. Interface supports default, static and private methods since java 8.
4. It provides total abstraction which means all the methods in an interface are declared with the empty body which are abstract methods, and all the fields are public, static, and final by default if nothing is specified before data type.

Marker Interface: An interface which has no members is known as a marker interface.   
 Ex: Serializable, Cloneable

Interface Handson: Must cover abstract method, default method, constructor usage and static usage as well.

Git Link: