Finalize() method

Object’s memory:-

* An object is created in the memory using new operator. Constructor is used to initialize the properties of that object
* When an object is no more required, it must be removed from the memory so that memory can be reused for other objects

In java:-

* There is no destructors in java
* There exists better mechanism to handle the garbage collection
* JVM implicitly sweeps out abandoned objects from the memory

Finalize method:-

Finalize() method is a protected and non-static method of java.lang.Object class.

Protected void finalize() throws Throwable

{

//code

}

Garbage Collector Thread:-

Garbage collector thread before sweeping out an abandoned object, it calls finalize() method of that object.

Explicit call to finalize():-

* You can finalize() method explicitly on an object before it is abandoned.
* When you call, only operations kept in finalize() method are performed on an object
* Object will not destroyed from the memory

Finalize method:-

* That means clean up operations which you have kept in the finalize() method are executed before an object is destroyed from the memory
* Garbage collector thread calls finalize() method only once for one object

No exception in finalize:-

* Exceptions occurred in finalize() method are not propagated. They are ignored by the Garbage Collector
* finalize() methods are not chained like constructors i.e three is no calling statement to super class finalize() method inside the finalize() method of subclass you need to explicitly call super class finalize() method

Force execution of finalize():-

* you call make finalize() method to be executed forcefully using
* Runtime.getRuntime().runFinalization()
* Runtime.runFinalizersOnExit(true)
* But both the methods have disadvantage Runtime.getRuntime().runFinalization() makes the just best effort to execute finalize() method. It is not grunted that it will execute finalize() method
* Runtime.runFinalilzersOnExit(true) is deprecated in JDK because same time it runs finalize() method on live object also