**Finalization – Part\_01**

* **Finalization:**

Just before destroying an object GC calls finalize method to perform cleanup activities.

Once finalize method completes automatically GC destroys that object.

finalize() present in Object class to with the following declaration.

protected void finalize() throws Throwable

We can override finalize() method in our class to define our own cleanup activities.

* **Case\_01:**

Just before destroying an object GC calls finalize method on the object which is eligible for GC. Then the corresponding class finalize method will be executed. For example, if String object eligible for gc, then String class finalized method will be executed, but not Test class finalize method.

Example:

class Test{

public static void main(String[] args){

String s = new String(“durga”);

s = null;

System.gc();

System.out.println(“End of main”);

}

public void finalize(){

System.out.println(“Finalize method called”);

}

}

Note:

In the above example, String object eligible for GC and hence String class finalize method got executed, which has empty implementation and hence the output is

End of main.

If we replace String object with Test object then Test class finalize method will be executed. In this case the output is

End of main

Finalize method called

Or

Finalize method called

End of main

Note: Till the line System.gc() there will be only one Thread, which is main thread, but after System.gc() there will be two

Threads as GC is a daemon thread.

* **Case\_02:**

Based on our requirement we can call finalize() method explicitly then it will be executed just like a normal method call and object won’t be destroyed.

Example:

class Test{

public static void main(String[] args){

Test t = new Test();

t.finalize();

t.finalize();

t = null;

System.gc();

System.out.println(“End of main”);

}

public void finalize(){

System.out.println(“Finalize method called”);

}

}

In the above program finalize method will be executed 3 times. In that two times explicitly by the programmer and one time by the GC. In this case output is

Finalize method called

Finalize method called

End of main

Finalize method called

Note:

If we are calling finalize() method explicitly then it will be executed like a normal method call and object won’t be destroyed. If GC calls finalize method then object will be destroyed.

Note:

init(), service() and destroy() methods are considered as life cycle methods of servlet. Just before destroying servlet

object web container calls destroy() method to perform cleanup activities. But based on our requirement we can call destroy method from init() and service() methods, then destroy method will be executed just like a normal method call and servlet object won’t be destroyed.