**Garbage Collection – Part\_02**

* **The ways to make an object eligible for GC.**

Even though programmer is not responsible to destroy useless objects, it is highly recommended to make an object eligible for GC. If it is no longer required.

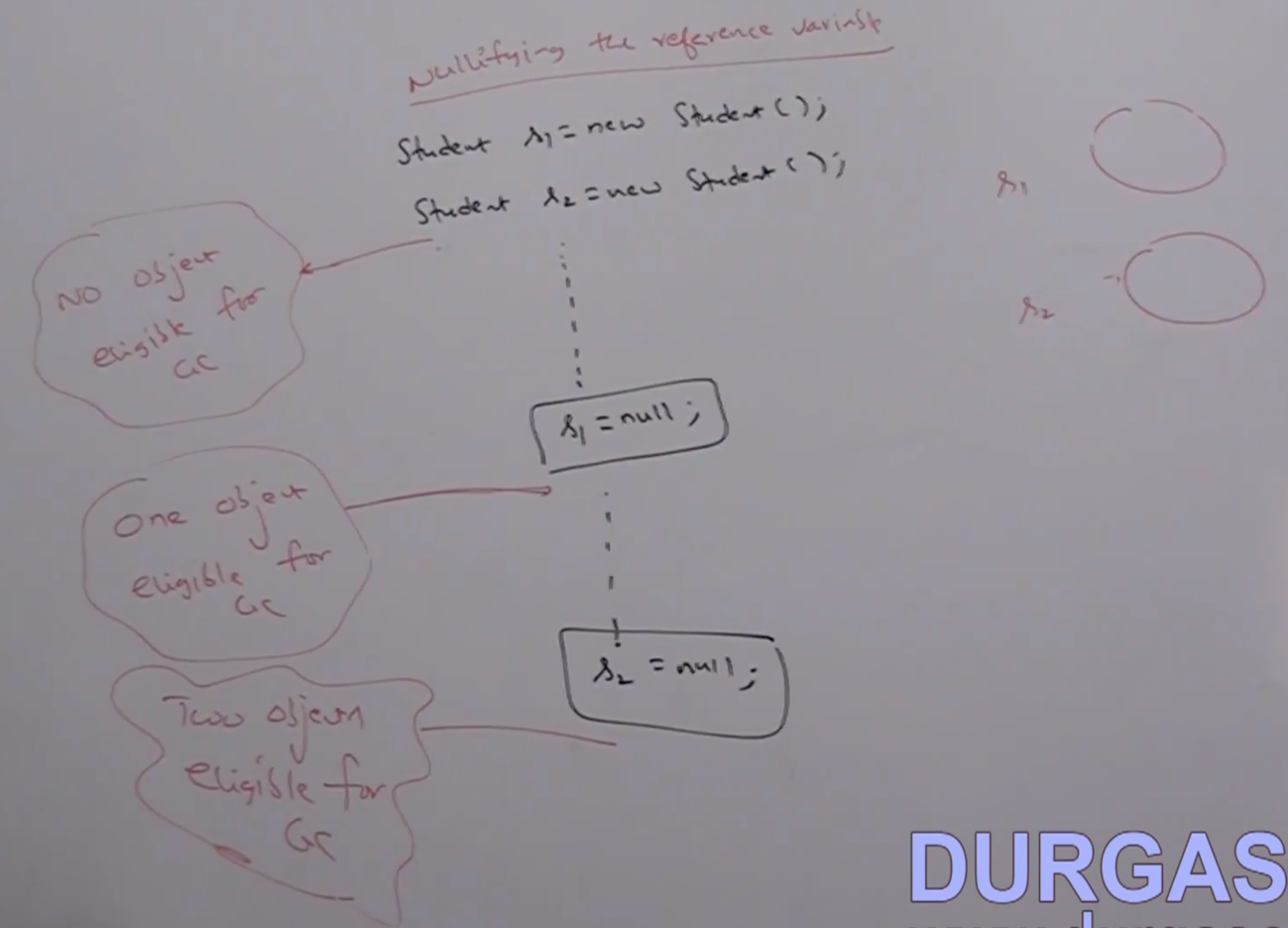
An object is said to be eligible for GC, if and only if it doesn’t contain any reference variable.

The following are various ways to make an object eligible for GC.

* **Way\_01: Nullifying the reference variable:**

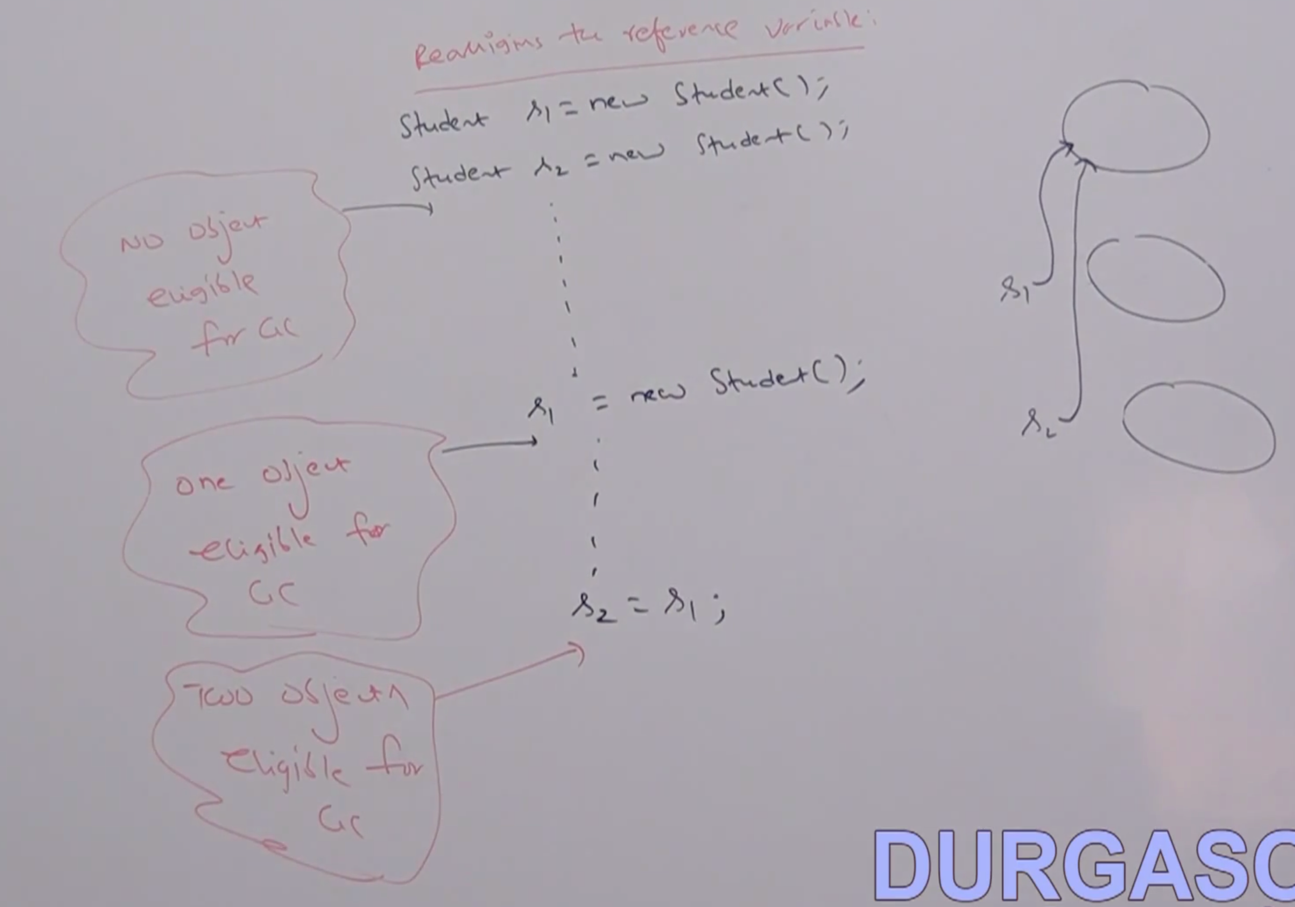
If an object no longer required then assign null to all its reference variables then that object automatically eligible for garbage collection.

This approaching is nothing by nullifying the reference variable.



* **Way\_02: Reassigning the reference variable:**

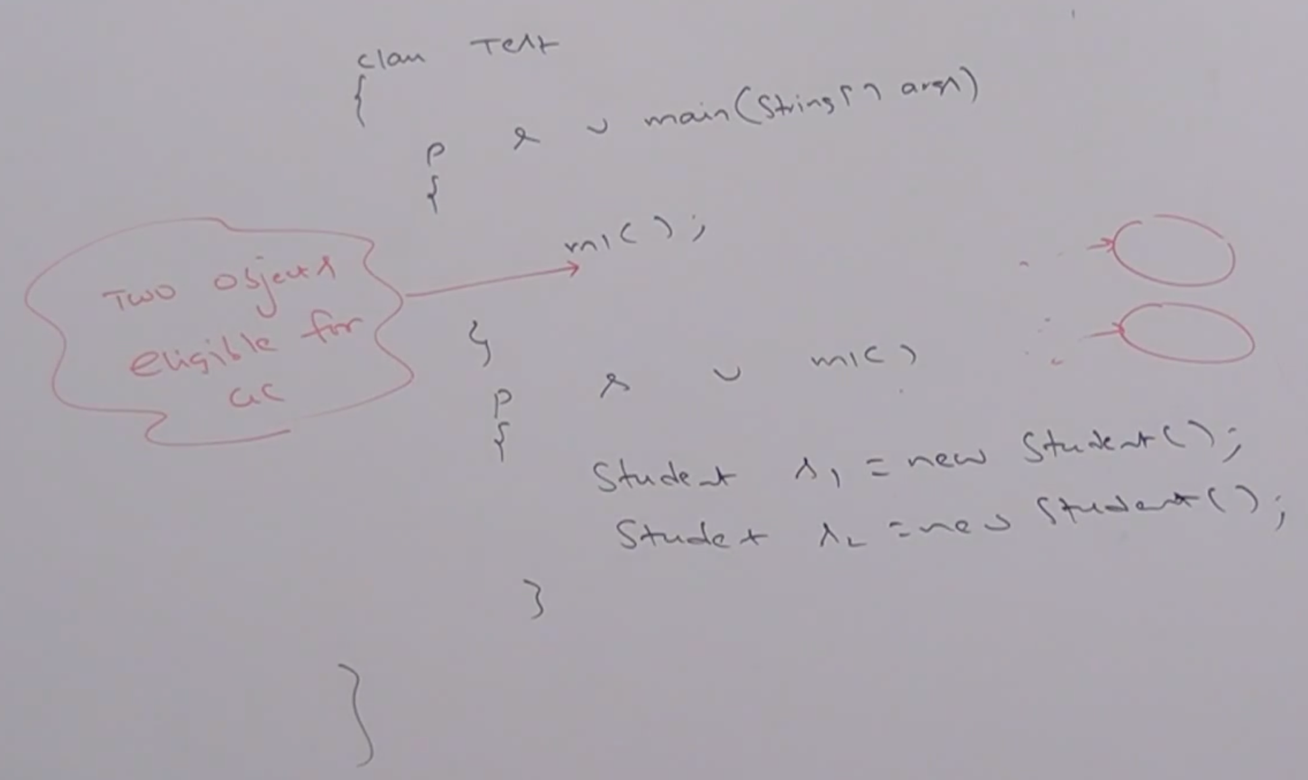
If an object no longer required, then reassign its reference variable to some other object then old object by default eligible for garbage collection.



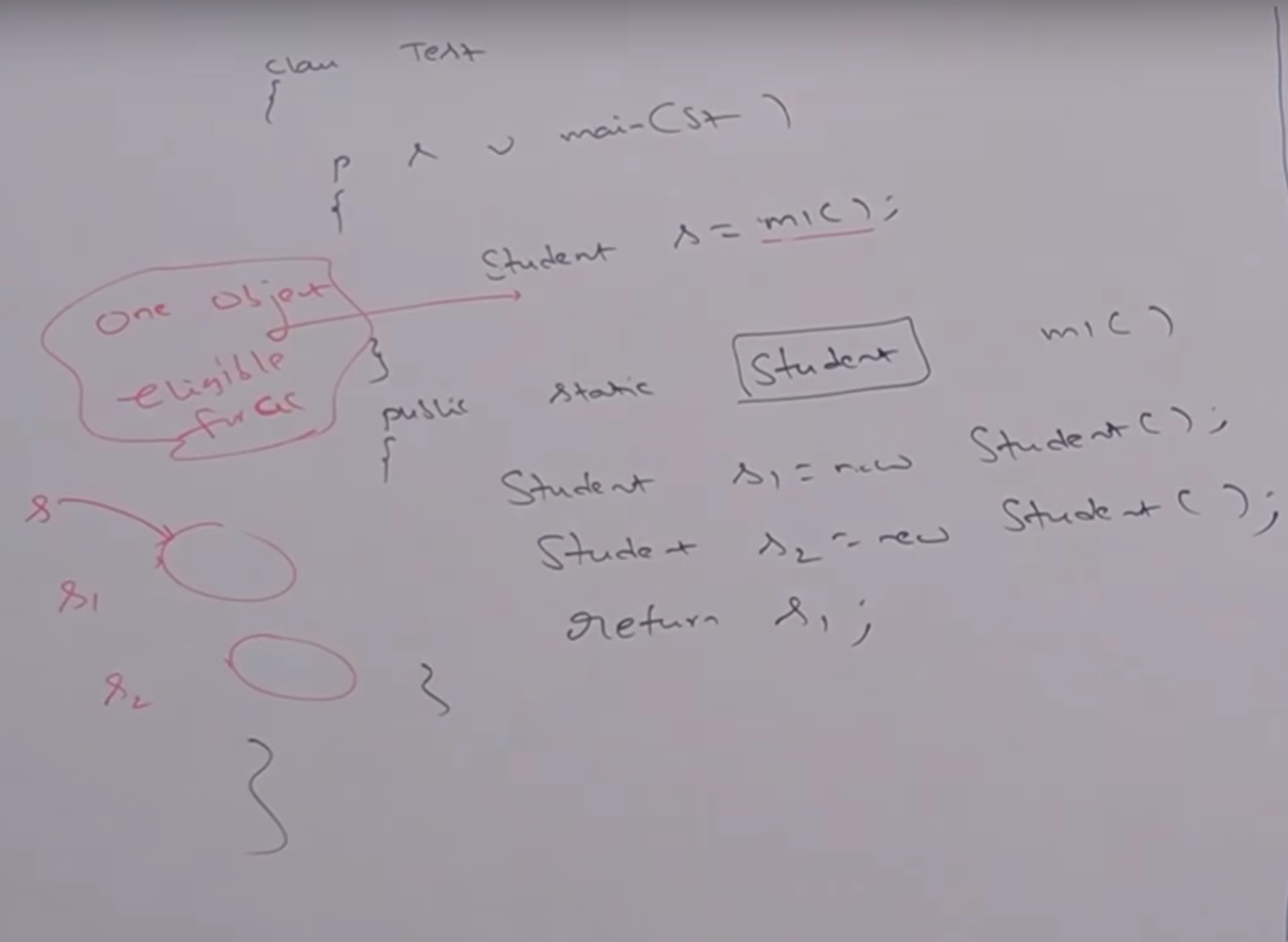
* **Way\_03: Objects created inside a method:**

The objects which are created inside a method are by default eligible for GC once method completes.

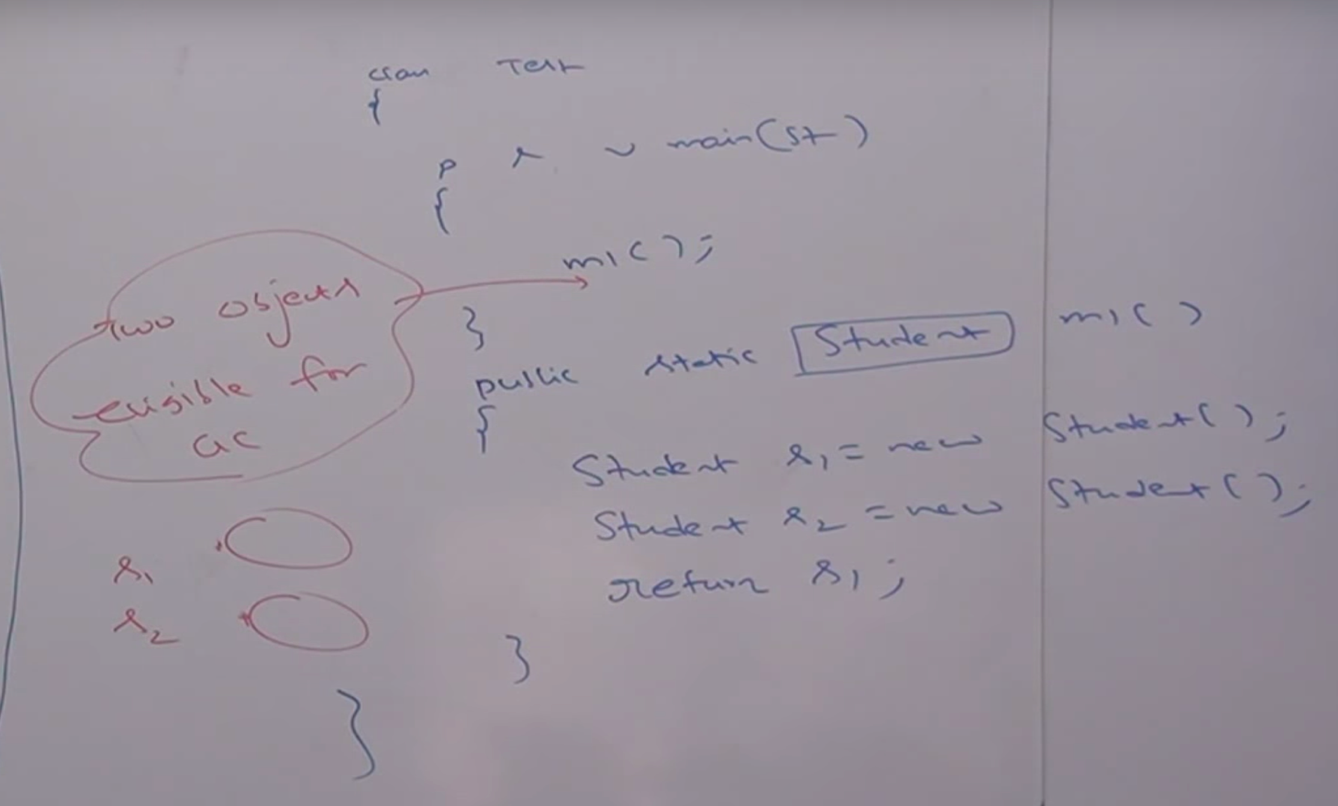
Example\_01:

****

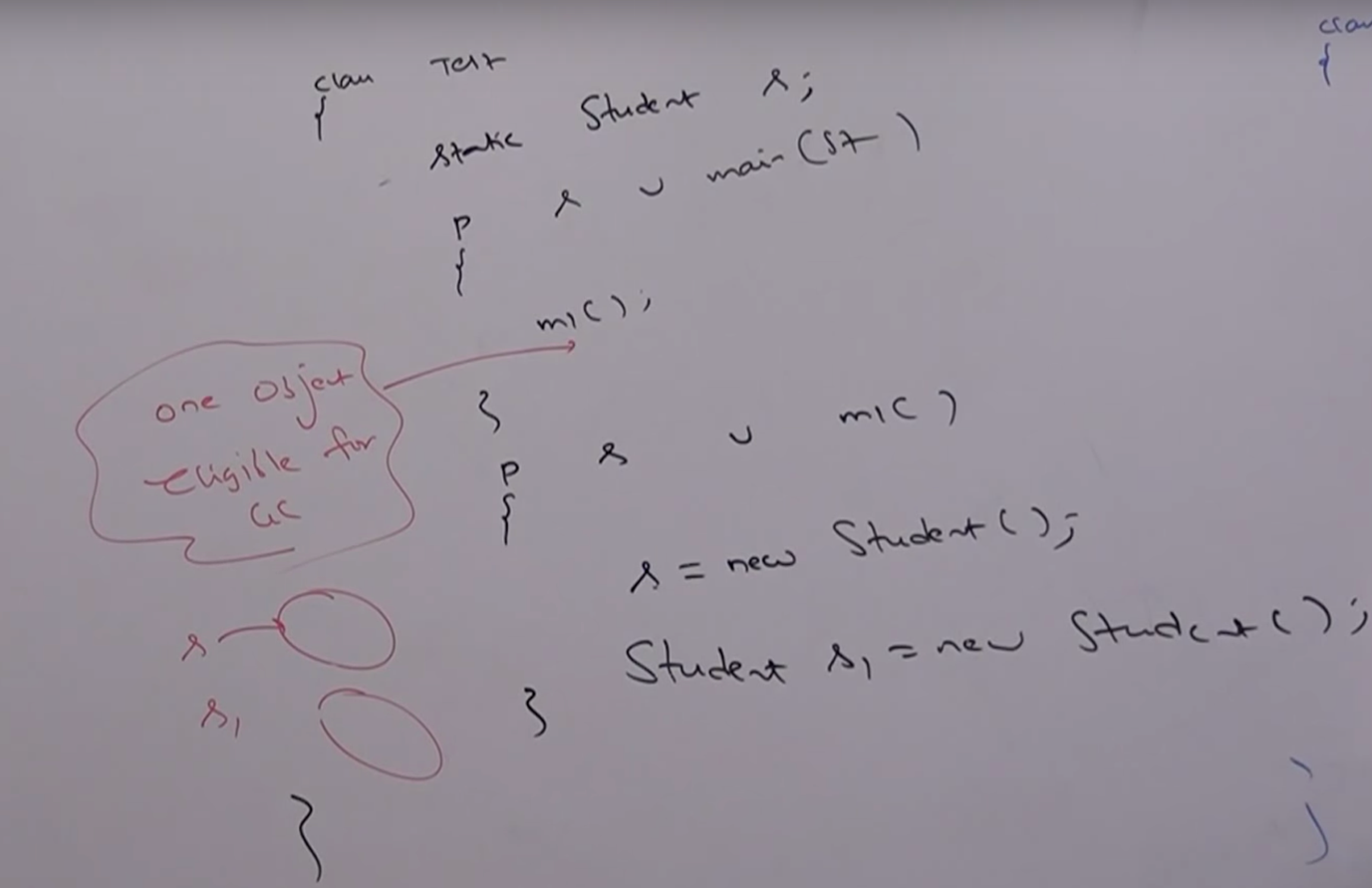
Example\_02:



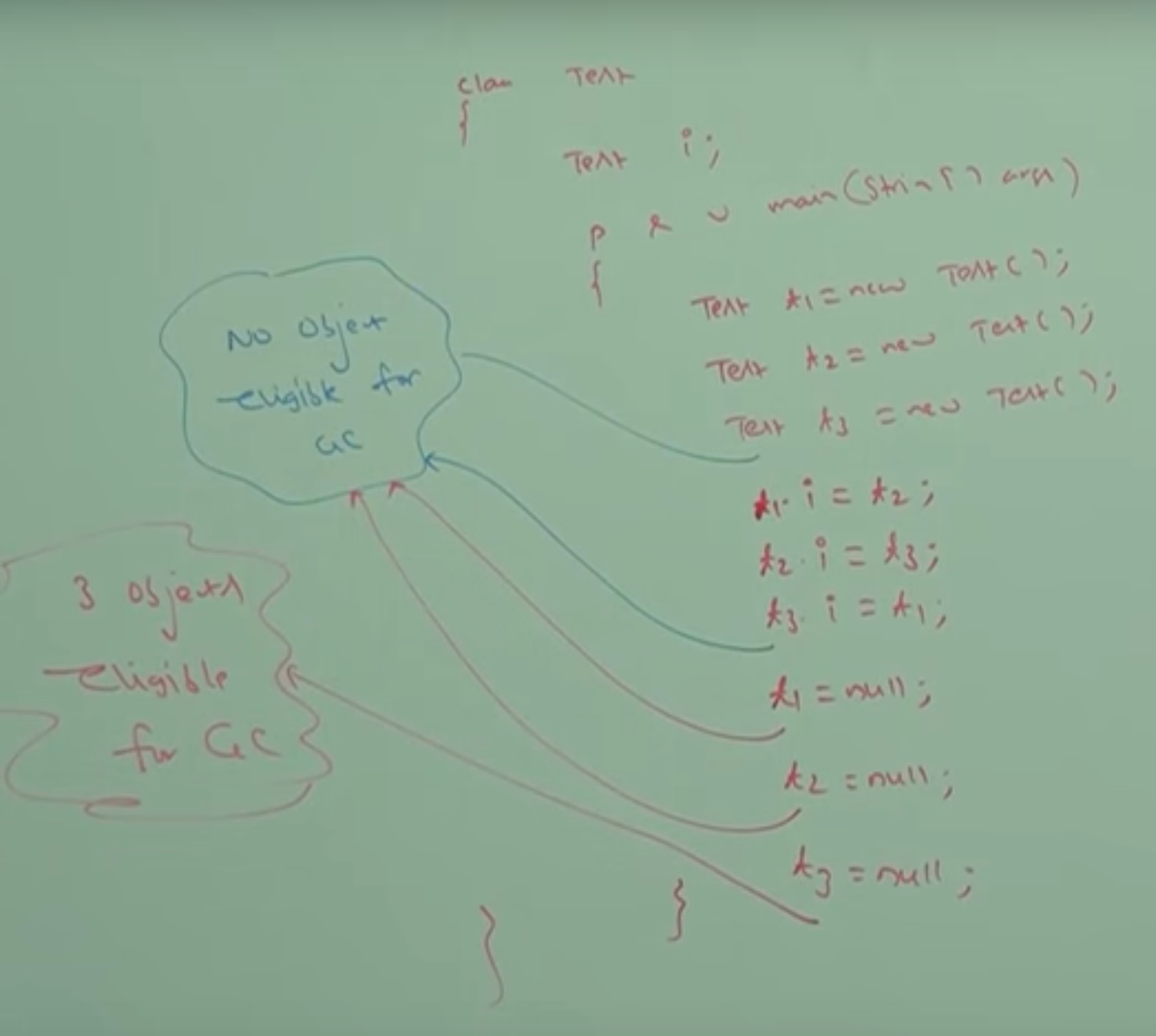
Example\_03:

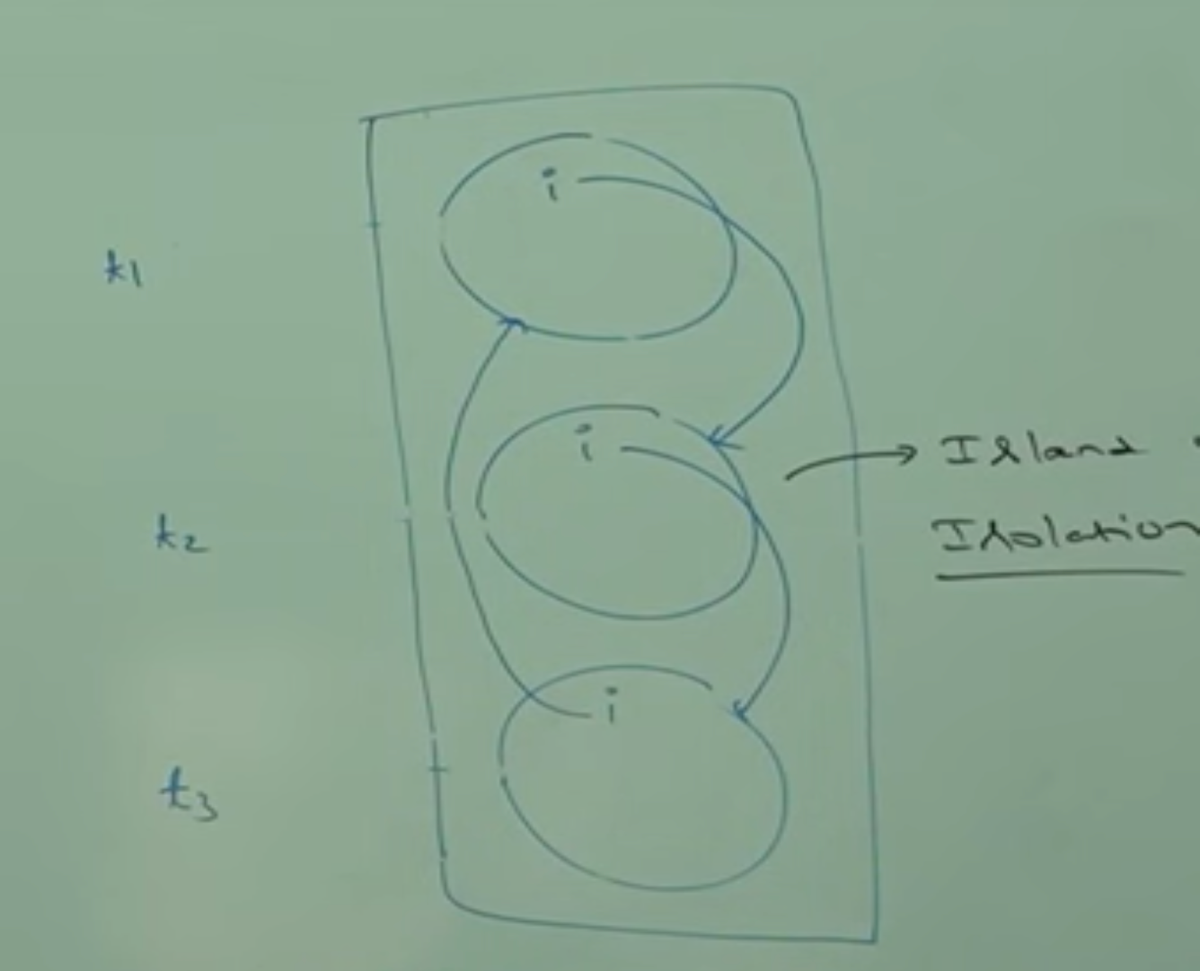


Example\_04:



* **Island of isolation:**





Note:

1. If an object doesn’t contain any reference variable then it is eligible for garbage collection always.
2. Even though object having references sometimes it is eligible for garbage collection (if all references are internal references). Example: Island of isolation.