



# Garbage Collection

Kishan

# Understanding Java Garbage Collection

- Garbage collection is a mechanism of re-Acquiring the heap space by destroying the objects which are eligible for "Garbage Collection".
- Garbage collector always running in the background of a java application for removing useless objects, So that the chance of failing java program is very rare because of memory problems.
- All Java objects reside in an area called the heap.
- The heap is created when the JVM starts up and may increase or decrease in size while the application runs.
- When the heap becomes full, garbage (Unused Objects) is collected by Garbage Collector.

- During the garbage collection objects that are no longer used are cleared, thus making space for new objects.
- The algorithm used by Garbage collector is "**Mark & Sweep**".
- Garbage Collection is not a process of collecting and discards dead objects, It is more like marking the "live" objects (all objects that are reachable from Java threads, native methods and other root sources) and everything else designated as garbage.
- In java objects are deleted by Garbage Collector implicitly.

## When an Object is available for Garbage Collection ?

- Below mentioned are the 3 possible ways where a java object eligible for garbage collection

By Re-assigning the reference variable

By Nullifying the reference variable

All Objects created inside method

### **By Re-assigning the reference variable:-**

- If we are reassigning an object reference to another object then automatically the first object is available for Garbage collection.

### **By Nullifying the reference variable:-**

- If we assign null value to the object reference, then that particular object is eligible for Garbage collection.

### **All Objects created inside method:**

- All objects created inside any method are by default eligible for Garbage Collection, provided after completion of the method implementation.

# How to call Garbage Collector manually?

- We can call the Garbage collector manually in '2' ways

1. By using System Class (System.gc()---> Its a static method)
2. By Using Runtime Class

```
Runtime r=Runtime.getRuntime();  
  
r.gc();
```

# IO Streams

Kishan

- In java a stream represents a sequence of objects (byte, characters, etc.) which we can access them in a sequential order.
- In java, I/O streams represents either an Input source or an output destination.
- There are mainly '4' types of streams

Name	Description
Byte Streams	Read and write stream of data in byte format
Character Streams	Read and write stream of data in Character format
Data Streams	Handles I/O streams of primitive data types
Object Streams	Handles object streams (Serialization)