Garbage collector in JAVA

Garbage Collection in Java is a process by which the programs perform memory management automatically.

The Garbage Collector (GC) finds the unused objects and deletes them to reclaim the memory.

In Java, dynamic memory allocation of objects is achieved using the new operator that uses some memory and the memory remains allocated until there are references for the use of the object.

When there are no references to an object, it is assumed to be no longer needed, and the memory occupied by the object can be reclaimed. There is no explicit need to destroy an object as Java handles the de-allocation automatically.

The technique that accomplishes this, is known as Garbage Collection. Programs that do not de-allocate memory can eventually crash when there is no memory left in the system to allocate. These programs are said to have memory leaks.

Garbage collection in Java happens automatically during the lifetime of the program, eliminating the need to de-allocate memory and thereby avoiding memory leaks.

In C language, it is the programmer’s responsibility to de-allocate memory allocated dynamically using free() function. This is where Java memory management leads.

# Advantages of Garbage Collection:

1. It makes java memory efficient because garbage collector removes the unreferenced objects from heap memory.
2. It is automatically done by the garbage collector (a part of JVM) so we don't need to make extra efforts.

finalize() method

The finalize() method is invoked each time before the object is garbage collected. This method can be used to perform clean-up processing.

gc() method

The gc() method is used to invoke the garbage collector to perform clean-up processing. The gc() is found in System and Runtime classes.