Contents

# Garbage collector

* In C/C++, programmer is responsible for both creation and destruction of objects. Usually programmer neglects destruction of useless objects. Due to this negligence, at certain point, for creation of new objects, sufficient memory may not be available and entire program will terminate abnormally causing OutOfMemoryErrors.
* But in Java, the programmer need not to care for all those objects which are no longer in use. Garbage collector destroys these objects.
* Garbage collector is best example of [Daemon thread](https://www.geeksforgeeks.org/daemon-thread-java/) as it is always running in background.
* Main objective of Garbage Collector is to free heap memory by destroying unreachable objects.

In Java, garbage collection is the process of managing memory, automatically. It finds the unused objects (that are no longer used by the program) and delete or remove them to free up the memory. The garbage collection mechanism uses several GC algorithms. The most popular algorithm that is used is Mark and Sweep

When a program executes in [Java](https://www.javatpoint.com/java-tutorial), it uses memory in different ways. The heap is a part of memory where objects live. It's the only part of memory that involved in the garbage collection process. It is also known as garbage collectible heap. All the garbage collection makes sure that the heap has as much free space as possible. The function of the garbage collector is to find and delete the objects that cannot be reached.

## When an Object becomes eligible for Garbage Collection?

An object become eligible if it is not used by any program or thread or any static references or its references is null. If two objects having reference (cyclic reference) of each other and does not have any live reference then both objects collected by the garbage collector. There are some other cases when an object become eligible for garbage collection:

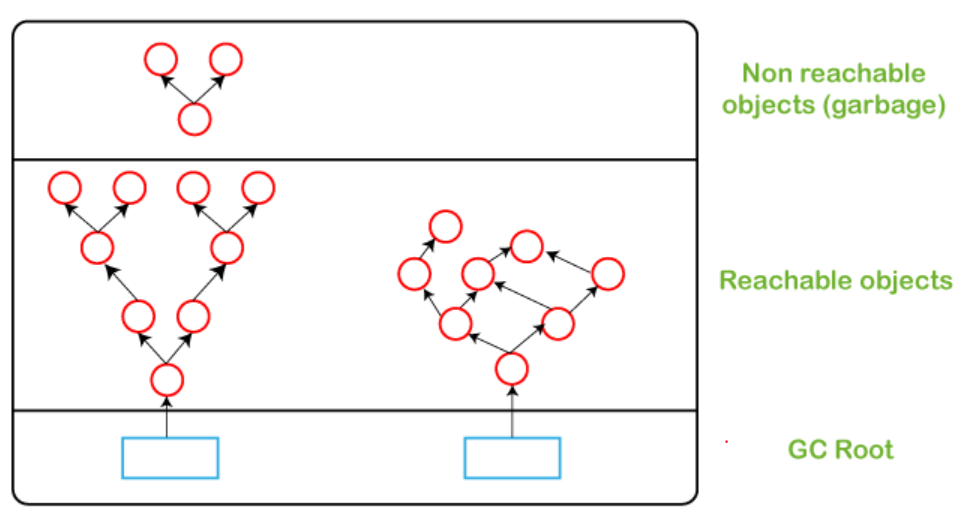
* If the reference of that object is explicitly set to null.
* The object also becomes eligible if it is created inside a block and the reference goes out of the scope once control exit from the block.

JVM controls the garbage collector. JVM decides when to perform the garbage collection. You can also request to the JVM to run the garbage collector. But there is no guarantee under any conditions that the JVM will comply. JVM runs the garbage collector if it senses that memory is running low. When [Java program](https://www.javatpoint.com/java-programs) request for the garbage collector, the JVM usually grants the request in short order. It does not make sure that the requests accept.

Every Java program has more than one thread. Each thread has its execution stack. There is a thread to run in Java program that is a main() method. Now we can say that an object is eligible for garbage collection when no live thread can access it. The garbage collector considers that object as eligible for deletion. If a program has a reference variable that refers to an object, that reference variable available to live thread, this object is called reachable.

## Mark and Sweep Algorithm

Mark Phase: Objects that are accessible from the threads, native handles, and other GC root sources are marked as live. Every object tree has more than one root object. GC root is always reachable. So, any object that has a garbage collection root at its root. It identifies and marks all objects that are in use, and the remaining can be considered garbage.



Sweep Phase: In this phase, the heap is traversed to find the gap between the live objects. These gaps are recorded in the free list and are available for new object allocation.

#### **Important terms :**

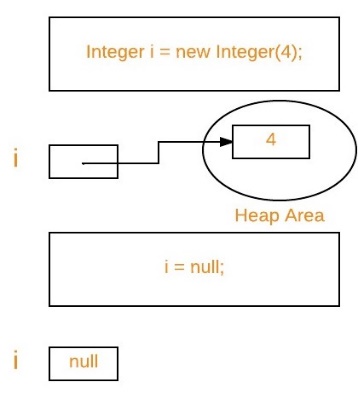
Unreachable objects : An object is said to be unreachable iff it doesn’t contain any reference to it. Also note that objects which are part of [island of isolation](https://www.geeksforgeeks.org/island-of-isolation-in-java/) are also unreachable.

Integer i = new Integer(4);

// the new Integer object is reachable via the reference in 'i'

i = null;

// the Integer object is no longer reachable.



Eligibility for garbage collection : An object is said to be eligible for GC(garbage collection) iff it is unreachable. In above image, after i = null; integer object 4 in heap area is eligible for garbage collection.

## Ways to make an object eligible for GC

* Even though the programmer is not responsible to destroy useless objects but it is highly recommended to make an object unreachable(thus eligible for GC) if it is no longer required.
* There are generally four different ways to make an object eligible for garbage collection.
  + 1. Nullifying the reference variable
    2. Re-assigning the reference variable
    3. Object created inside method
    4. [Island of Isolation](https://www.geeksforgeeks.org/island-of-isolation-in-java/)
* Once we made object eligible for garbage collection, it may not destroy immediately by the garbage collector. Whenever JVM runs the Garbage Collector program, then only the object will be destroyed. But when JVM runs Garbage Collector, we can not expect.
* We can also request JVM to run Garbage Collector. There are two ways to do it :
  + 1. Using System.gc() method : System class contain static method gc() for requesting JVM to run Garbage Collector.
    2. Using Runtime.getRuntime().gc() method : [Runtime class](https://www.geeksforgeeks.org/java-lang-runtime-class-in-java/) allows the application to interface with the JVM in which the application is running. Hence by using its gc() method, we can request JVM to run Garbage Collector.

Note :

* 1. There is no guarantee that any one of above two methods will definitely run Garbage Collector.
  2. The call System.gc() is effectively equivalent to the call : Runtime.getRuntime().gc()

## Finalization

* Just before destroying an object, Garbage Collector calls finalize() method on the object to perform cleanup activities. Once finalize() method completes, Garbage Collector destroys that object.
* finalize() method is present in [Object class](https://www.geeksforgeeks.org/object-class-in-java/) with following prototype.
* protected void finalize() throws Throwable

Based on our requirement, we can override finalize() method for perform our cleanup activities like closing connection from database.

Note :

* 1. The finalize() method called by Garbage Collector not [JVM](https://www.geeksforgeeks.org/jvm-works-jvm-architecture/). Although Garbage Collector is one of the module of JVM.
  2. [Object class](https://www.geeksforgeeks.org/object-class-in-java/) finalize() method has empty implementation, thus it is recommended to override finalize() method to dispose of system resources or to perform other cleanup.
  3. The finalize() method is never invoked more than once for any given object.
  4. If an uncaught exception is thrown by the finalize() method, the exception is ignored and finalization of that object terminates.