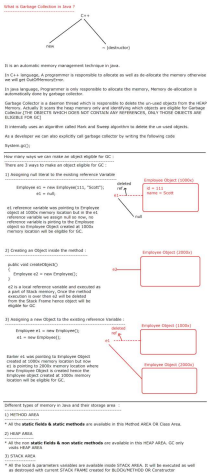
What is Garbage Collection in Java ?
C++
new
~ (destructor)
It is an automatic memory management technique in java .
In C++ language, A programmer is responsible to allocate as well as de-allocate the memory otherwise we will get OutOfMemoryError.
In java language, Programmer is only responsible to allocate the memory, Memory de-allocation is automatically done by garbage collector.
Garbage Collector is a daemon thread which is responsible to delete the un-used objects from the HEAP AREA. It is responsible to free up the memory only and identifying which objects are eligible for Garbage Collector [THE OBJECTS WHICH DOES NOT CONTAIN ANY REFERENCE, ONLY THOSE OBJECTS ARE ELIGIBLE FOR GC]
It internally uses an algorithm called Mark and Sweep algorithm to delete the un-used objects.
As a developer we can also explicitly call garbage collector by writing the following code
System.gc();
How many ways we can make an object eligible for GC :
There are 3 ways to make an object eligible for GC :
1) Assigning null literal to the existing reference variable
Employee e1 = new Employee(111, "Scott");
e1 = null;
deleted
ref
e1
Employee Object (1000x)
ID = 111
name = Scott
null
e1 reference variable was pointing to Employee Object created at 1000x memory location but now reference variable we assign null so now, no reference variable is pointing to the Employee Object created at 1000x hence the 1000x memory location will be eligible for GC.
2) Creating an Object inside the method :
public void createObject()
{
Employee e2 = new Employee();
}
e2
Employee Object (2005x)
e2 is a local reference variable and executed as a part of Stack memory, Once the method execution is completed the stack frame is removed from the Stack Frame hence object will be eligible for GC
3) Assigning a new Object to the existing reference Variable :
Employee e1 = new Employee();
e1 = new Employee();
deleted
ref
e1
Employee Object (1300x)
Employee Object (2300x)
Earlier e1 was pointing to Employee Object created at 1300x memory location but now reference variable e1 is pointing to the new new Employee Object is created hence the Employee object created at 1300x memory location will be eligible for GC.
Different types of memory in Java and their storage area :
1) METHOD AREA
* All the static fields & static methods are available in this Method AREA OR Class Area.
2) HEAP AREA
* All the non static fields & non static methods are available in this HEAP AREA. GC only visits HEAP AREA
3) STACK AREA
* All the local & parameters variables are available inside STACK AREA. It will be executed as well as destroyed with current STACK FRAME created for BLOCK/METHOD OR Constructor