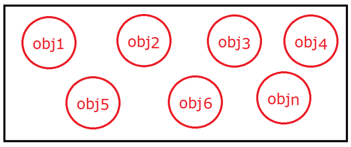
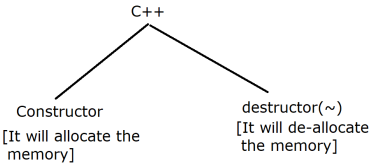
What is Garbage Collector in java

Whenever we create an object in java then with the help of new keyword memory is allocated for the Object (Non static members) in the HEAP area.



HEAP AREA

In older languages like C++, Memory allocation and memory de-allocation is the responsibility of programmer



C++

Constructor
[It will allocate the memory]

destructor(~)
[It will de-allocate the memory]

* If C++ programmer forgot to delete the memory then there is a chance of getting OutOfMemoryError

---

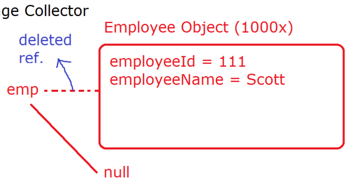
How many ways we can make an object eligible for Garbage Collector :

There are 3 ways to make an object eligible for Garbage Collector

1) Assigning null literal to the reference variable :

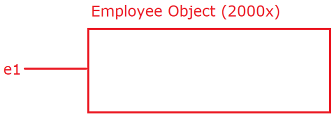
```
Employee emp = new Employee(111, "Scott");
emp = null;
```

Note : emp variable was pointing to Employee Object created at 1000x memory location but now emp variable is pointing to null, as a result now the obejct is un-reference, hence eligible for Garbage Collector.



2) Creating an Object inside a method :

```
public void createObject()
{
     Employee e1 = new Employee();
}
```

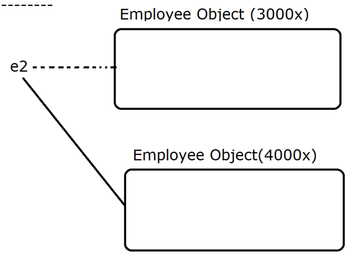


Here e1 is a local reference variable so created in the Stack Memory, We know that it is created in the Stack Frame so once method execution is over this e1 variable will be deleted from the Stack Frame so object is un-referenced, eligible for Garbage Collector.

3) Assigning a new Object to the existing reference variable :

```
Employee e2 = new Employee();
e2 = new Employee();
```

e2 reference variable was pointing to 3000x but not e2 reference variable is pointing to 4000x so the 3000x object is un-reference hence eligible for Garbage Collector



HEAP and STACK Digram :

```
class Student
{
    int roll = 101;
}

public class Test
{
  public static void main(String [] args)
  {
      int localVar = 500; ✓
      Student s1 = new Student(); ✓
      s1 = new Student();
  }
}
```

HEAP Memory
1000x : StudentObject, roll : 101
2000x : StudentObject, roll : 101

STACK Memory
main_stack
----
localVar : 500
s1 : ~~1000x~~ 2000x

In java, Whenver we execute a program then JVM gets some memory from the Operating System, The memory is divided into two section 1) HEAP memory 2) Stack Memory