Garbage Collection

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 not longer in use. Garbage collector destroys these objects.

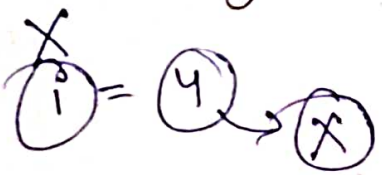
Garbage collector is always running in backgrd.

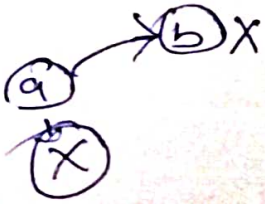
Main objective of Garbage Collector is to free heap memory by destroying unreachable objects.

Unreachable objects

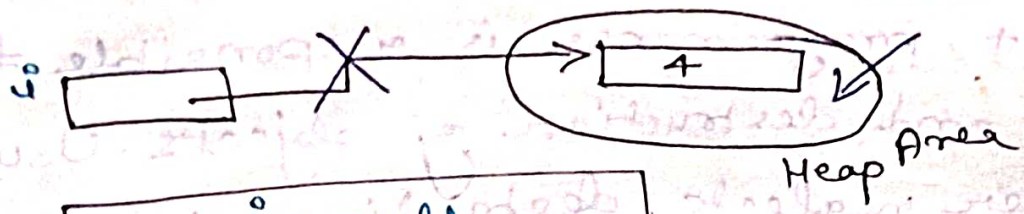
An object is said to be unreachable if it doesn't contain any reference to it. Also note that objects which are part of island of isolation are also unreachable.

```
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.
```





```
Integer i = new Integer(4);
```




```
i = null
```

i → null

Island of Isolation

Object 1 references Object

Eligibility for Garbage Collection

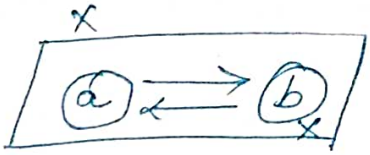
An object is said to be eligible for GC (garbage collection) if it is unreachable. In above image, after i = null; integer object 4 in heap area is eligible for garbage collection.

Island of isolation

Object 1 reference Object 2 and Object 2 reference Object 1. Neither object 1 nor object 2 referenced by any other object. That's an island of isolation.

Basically, an island of isolation is a group of objects that reference each other but they are not referenced by any active object in the application.

Strictly speaking, even a single unreferenced object is an island of isolation too



```
public class Test
{
  Test i;
  public static void main (String [] args)
  {
    Test t1 = new Test();
    Test t2 = new Test();

    // object of t1 gets a copy of t2
    t1.i = t2;
    // object of t2 gets a copy of t2
    t2.i = t1;
    // Till now no object eligible
    // for garbage collection.
    t1 = null;
    // now two objects are eligible for
    // garbage collection.
    t2 = null;
    // calling garbage collection.
    System.gc();
```

TYPES OF GARBAGE COLLECTION

Serial Garbage Collector.
Parallel Garbage Collector.
CMS Garbage Collector.
G1 Garbage Collector