

# Memory leaks in Java

Difficulty Level : Basic Last Updated : 07 May, 2021

In C, programmers totally control allocation and deallocation of dynamically created objects. And if a programmer does not destroy objects, memory leak happens in C,

Java does automatic Garbage collection. However there can be situations where garbage collector does not collect objects because there are references to them. There might be situations where an application creates lots of objects and does not use them. Just because every objects has valid references, garbage collector in Java can't destroys the objects. Such types of useless objects are called as Memory leaks. If allocated memory goes beyond limit, program will be terminated by rising OutOfMemoryError. Hence if an object is no longer required, it is highly recommended to make that object eligible for garbage collector. Otherwise We should use some tools that do memory management to identifies useless objects or memory leaks like:

- HP OVO
  - HP J METER
  - JProbe
  - IBM Tivoli
- 

```
// Java Program to illustrate memory leaks
import java.util.Vector;
public class MemoryLeaksDemo
{
    public static void main(String[] args)
    {
        Vector v = new Vector(214444);
        Vector v1 = new Vector(214744444);
        Vector v2 = new Vector(214444);
        System.out.println("Memory Leaks");
    }
}
```

Output:

Exception in thread "main" java.lang.OutOfMemoryError: Java heap space exce