COURSEWARE

Garbage Collection

Contents

Overview

In Java, **garbage collection** is the process by which Java programs perform automatic memory management.

When Java programs run on the *Java Virtual Machine (JVM)*, objects are created on the *heap*, which is a portion of memory dedicated to the program. Eventually, some objects will no longer be needed; the garbage collector finds these unused objects and deletes them to free up memory.

Java differs from other languages (C, C++, etc.) in that its memory management is fully automatic.

It is therefore useful for us to learn how garbage collection works, so that we can utilise it effectively.

Tutorial

Garbage Collection Eligibility

Even though, in Java, you are not responsible for destroying useless objects, it is highly recommended to make an object unreachable if it is no longer required.

There are generally four different ways to make an object eligible for garbage collection:

- *Nullifying the reference variable*
- *Re-assigning the reference variable*
- *Object created inside method*
- *Creating an island of isolation* - where two objects reference each other, but no other code references either object

Unreachable Code

An object is said to be eligible for garbage collection if it is *unreachable*.

An object is unreachable if code that is currently running no longer stores any reference that object.

```
public static void main(String[] args) {

    Integer i = new Integer(4);

    i = null;
}
```

In the above example, initialising the Integer `i` reserves a memory block in the 'Heap', reassigning the variable `i` to `null` does not remove the value 4 from the heap.

The value in the heap becomes unreachable (The variable `i` no longer points to it) and thus is marked for garbage collection.

Exercises

There are no exercises for this module.