Garbage Collection

CSCI 201
Principles of Software Development

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu
Outline

- Garbage Collection
Garbage Collection

- Java doesn’t require much memory management
  - C++ has `malloc`, `free`, `delete`

- Since Java does not have pointers (or these key words), another mechanism was needed to clean up memory after it was used
  - When a memory location is no longer referenced, the Garbage Collector will come around and delete that reference in memory

- Garbage collection looks at memory and identifies which locations are still referenced by variables and which are not
  - A memory location is a candidate for garbage collection when it no longer is referenced by any part of your program
Garbage Collection Example

- When are the memory locations pointed to by each of the following variables candidates for garbage collection?

```java
public class Garbage {
    private ArrayList num;
    private static String val;
    public Garbage(ArrayList num) {
        this.num = num;
        val = “hello”;
    }
    public static void meth() {
        Garbage g = new Garbage(new ArrayList());
        g.num = new ArrayList();
        Garbage.val = “hi”;
    }
    public static void main(String [] args) {
        Integer number = new Integer(3);
        for (int i=0; i < number; i++) {
            System.out.print(i);
        }
        meth();
        System.out.println(number);
    }
}
```
Garbage Collection Steps

- **Step 1 – Marking**
  - The garbage collector determines which pieces of memory are in use and which are not

```
<table>
<thead>
<tr>
<th>Before Marking</th>
<th>After Marking</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>
```

- **Live object**
- **Unreferenced Objects**
- **Memory space**

**Garbage Collection Steps**

- **Step 2 – Normal Deletion**
  - Garbage collector removes unreferenced objects leaving referenced objects and pointers to free space

*Image Source: http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html*
Garbage Collection Steps

- Step 2a – Deletion with Compacting
  - Garbage collector deletes unreferenced objects and compacts the remaining referenced objects

Invoking Garbage Collection

- Programmers are not able to explicitly invoke the garbage collector
- We can make a suggestion to the JVM to run the garbage collector with
  
  ```java
  System.gc();
  ```
  
  This does not mean that the garbage collector will be run immediately though

- More garbage collection information can be found at
  http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html