

Study questions set #5

1. True or false:
 - a. Winston Royce's "waterfall" model is an inaccurate model of real-world software development.
 - b. Primitive integer variables on the stack have a default value of zero if not explicitly initialized.
 - c. Strings are mutable, primitive wrapper classes are not.
2. Sketch the stack and heap at the end of the following main method. Be sure to include all local variables, objects, and arrays. Reference variables should contain "@" with an arrow pointing to the associated memory address. Indicate String objects with a box containing the string in quotes. You may assume no memory has been garbage collected.

```
public static void main(String[] args) {  
  
    String hello = new String("hello");  
    String hi = hello;  
    String[] strs = new String[4];  
    strs[0] = hi;  
    strs[1] = new String("hello");  
    strs[2] = new String("world");  
    strs[3] = strs[0]+strs[1]+strs[2];  
  
}
```

3. Will the following code compile? If not, where is the error? If so, what is the output?

```
public final class FoolProof {  
  
    private static final String[] strs = {"abc","def","ghi"};  
  
    public String[] getStrs() {  
        String[] ret = new String[strs.length];  
        for(int i = 0; i < strs.length; i++) ret[i] = strs[i];  
        return ret;  
    }  
  
    public static void main(String[] args) {  
        FoolProof safe = new FoolProof();  
        System.out.println(safe.getStrs()[0]);  
        safe.getStrs()[0] = "gotcha";  
        System.out.println(safe.getStrs()[0]);  
        strs[0] = "gotcha2";  
        System.out.println(safe.getStrs()[0]);  
    }  
}
```

4. Implement the following methods.

```
//Compute the average value of the elements in data
public static double avg(int[] data) {...}
//Return an array that only contains the same elements as the original,
//in the same order, except with duplicates removed. For example:
//[1,2,4,5,1,3,2,5] ---> [1,2,4,5,3]
public static int[] unique(int[] orig) {...}
//Determine the mode of the data.
//This is the value that occurs most frequently.
//In case of a tie, return the right-most such value.
public static int mode(int[] data) {...}
//Return a new array which is a sorted version of the original.
//The elements should be sorted in ascending order (low to high)
public static int[] sort(int[] orig) {...}
//Return a transposed version of the original 2D array.
//The value at (i,j) in the original should be at (j,i) in the
//transpose.
public static int[][] transpose(int[][] orig) {...}
//Return a 2D ragged array containing the specified ranges.
//For example, lows==[1,3] and highs==[4,5] should return
//[ [1,2,3,4], [3,4,5] ]
public static int[][] ranges(int[] lows, int[] highs) {...}
//Return a filtered version of the original 2D array.
//Each non-boundary entry (i,j) in the new array should be replaced
//by the average of the eight surrounding entries.
public static double[][] smooth(double[][] original) {...}
```