CMSC 131 Spring 2005 Quiz 4 Worksheet

The fourth Quiz of the course will be on Wednesday February 23 in lab session. The following list provides more information about the quiz:

- You will have 15 minutes to complete the quiz.
- It will be a written quiz (not using any computer).
- It will be closed-book, closed-notes, and no calculator is allowed.
- Answers must be neat and legible. We recommend that you use pencil and eraser.
- The quiz will be based on the exercises you will find below.

The following exercises cover the Quiz#4 material. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with TAs and instructors during office hours.

**Problem 1 (General Questions)**

a. What is the difference between System.out.println and System.out.print?
b. What is the heap?
c. What is the difference between an object and a class?
d. Is the heap an infinite memory area? Explain.
e. What is garbage collection?
f. Are the following assignments valid?

```java
int age = 17;
double temp = 3.4;
String m = “John” + age + “/” + temp;
```

g. What is a reference variable?
h. What is the difference between a reference and a primitive variable?
i. How many reference variables can be associated with an object?
j. In the following code fragment, how many objects do we have at every step?

```java
String r;
String s;     // Step 1
r = s = “Happy”;     // Step2
r = new String(s);   // Step 3
r = null;            // Step 4
r = new String(“John”);  // Step 5
```

k. When do we use the String equals method?
l. What would happen when you execute the following code fragment?

```java
String k = null;
System.out.println(k);
System.out.println(k.length());
```
Problem 2

A class named Utilities has a method named minimum that takes three integers as parameters and returns the minimum value. Implement the method and the class, and provide a test driver that illustrates how to use the method to compute the minimum value of three values provided by the user. Read values from the user by using JOptionPane. Note: You may not use JOptionPane in the minimum method.

Problem 3

A memory map is a representation of the values of variables, and the objects we have in a program. For example, given the following code segment:

```java
public static void main(String[] args) {
    int x;
    x = 2;
    String p;
    p = new String(“Test”);
    String t = null;
    // STOP HERE
}
```

the memory map we have until the point right before the method finishes (indicated by the comment // STOP HERE) is:

As you can see we represent references (addresses) with arrows, objects with rectangles, and the null value with a line that has a perpendicular line at its endpoint. The stack is where the method’s local variables reside; the heap is any area to the right of the stack.
Using the above memory map approach draw a memory map for each of the following:

a. Problem 1.j above after finishing Step 5.
b. For the following method until we reached the point marked // STOP HERE

```java
public class Test {
    public static void main(String[] args) {
        String p = new String("House");
        String m;
        double cost = 3.4;

        m = new String(p);
        m = null;
        m = p;
        p = new String("Compiler");
        // STOP HERE
    }
}
```

**Problem 4**

The next two classes will be used to answer the questions that follow.

```java
public class MethodsExample {
    public void println(String name) {
        if (validName(name)) {
            System.out.println("Welcome: "+name);
        } else {
            System.out.println("Who are you?");
        }
    }

    private boolean validName(String name) {
        if (name.equals("John") || name.equals("Mary"))
            return true;
        else
            return false;
    }

    public void printAll(String name, int age) {
        println(name);
        System.out.println("Age: "+age);
    }
}

public class Driver {
    public static void main(String[] args) {
        // CODE SEGMENT
    }
}
```
a. In the MethodsExample class the printName method calls the validName method. Can all the methods of the class call each other? Does it make a difference whether the method is public or private? Does it make a difference the methods’ declaration order?

b. Indicate whether each of the following code segments are valid or invalid. Each code segment represents the body of the main method in the Driver class (where the //CODE SEGMENT comment appears).

i. MethodsExample e = new MethodsExample();
   e.printAll("John", 20);

ii. MethodsExample e = new MethodsExample();
    e.validName("John");

iii. MethodsExample e;
    e.printName("John");

iv. new MethodsExample().printAll("Mary", 16);

v. Revisit i. through iv. but assume all methods of the class MethodsExample are public.

vi. Revisit i. through iv. but assume all methods of the class MethodsExample are private.