1.00 Introduction to Computers and Engineering Problem Solving

Quiz 1 March 7, 2003

Name:	
Email Address:	
TA:	
Section:	

You have 90 minutes to complete this exam. For coding questions, you do not need to include comments, and you should assume that all necessary files have already been imported.

Good luck.

Question	*Points*
Question 1	**/ 15**
Question 2	**/ 15**
Question 3	**/ 20**
Question 4	**/ 50**
Total	**/ 100**

Problem 1. (15 points)

Refer to the following code fragment to answer the TRUE or FALSE statements as appropriate:

```
public class Quiz1 {
     private double quizAvg;
     public static void main(String[] args) {
        int i = 2;
        double d = Math.pow(36, 1/i);
        System.out.println(d);

        float k = i;

        for (i=0; i<5; i++) { //outer for loop
            for (int j=0; j<5; j++) { //inner for loop
                if (j==2) {
                    break;
                }
            }
        }
    }
    public Quiz1(double q) {quizAvg= q:}
}
```

a. The program above outputs the value `6.0`.

TRUE **FALSE**

b. The statement `float k = i;` is a legal statement. (*I.e.*, it will compile and run.)

TRUE FALSE

c. When `j` is equal to 2, the `break` statement forces the `main` method to quit the outer `for` loop and to proceed with the statement after the `break`.

TRUE **FALSE**

d. `String` is a Java primitive data type.

TRUE **FALSE**

e. The `Quiz1` class has one constructor. In general, a class can have more than one constructor.

TRUE FALSE

Problem 2. (15 points)

Consider the following Java code fragment, which compiles and runs without error:

```
public class Mystery
{
    public static void main(String[] args)
    {
        int row = 5, column;
        while (row >=1)
        {
            column = 1;
            while (column <= 3)
            {
                if (row %2 == 1)
                    { System.out.print("<"); }
                else
                    { System.out.print (">"); }
                ++column;
            }
            --row;
            System.out.println();
        }
    }
}
```

In the space below, write the exact output of this program. Note that `System.out.print` outputs a character on the same line as the previous character, whereas `System.out.println` inserts a line break before outputting the next character.

Output:

```
<<<
>>>
<<<
>>>
<<<
```

Problem 3. (20 points)

Complete the following method to find the maximum element in an array of integers (elements of the array can be positive, negative, or zero). If the array contains multiple copies of the largest integer, it does not matter which of these is returned. For instance, if an array contains the elements -1, 5, 0, 1, 18, -9, 18, -2, 18, method `max` must return 18. Note: you need not use recursion. Assume that the array contains at least one element.

```
public static int max(int[] arr)
{
    // Your code here

     int maxValue = arr[0];
     for (int i = 1; i < arr.length; i++)
          if (arr[i] > maxValue)
               maxValue = arr[i];

     return maxValue;

}
```

Problem 4. (50 points)

George Lucas calls you on the phone and offers you 800 billion dollars to help adapt his popular *Star Wars* films into a computer game. Accepting his lucrative deal, you decide to define a class `JediKnight` to represent the films' Jedi Knight characters. Knowing that every Jedi Knight is either on the Light side of the Force or the Dark side of the Force, you begin by writing the following:

```
public class JediKnight {

     /* total number of JediKnights created */
     private static int totalCount = 0;

     /* number of JediKnights on the Light side of force */
     private static int lightCount = 0;

     /* name of the JediKnight */
     private String name;

     /* true if light, false if dark */
     private boolean isOnLightSide;

     /* JediKnight constructor; takes two arguments */
     public JediKnight(String jediName, boolean isLight) {

     // Your code for part (a) here (see next page)
```

a

```
          name = jediName;
          isOnLightSide = isLight;
          totalCount++;
          if (isLight) lightCount++;
```

```
     }
}
```

a. Complete the constructor for `JediKnight.` IMPORTANT: Be sure to initialize or update all the appropriate class and instance variables. Write your answer in the space provided above.

b. Add to the `JediKnight` class a public, static method named **`totalJediCount()`** that returns the total number of `JediKnight` instances that have been created. Please write your method in the space provided below.

b

```
public static int totalJediCount() {
     return totalCount;
}
```

c. Add to the `JediKnight` class a public, static method **`print()`** that prints out the number and percentage of `JediKnights` on the **Dark** side of the Force. Your method must not return a value. Again, please write your method in the space provided below.

c

```
public static void print() {

     int darkCount = totalCount – lightCount;

     System.out.println(
       "Total Jedi knights on dark side: " + darkCount);

     System.out.println(
       "Percentage of Jedi knights on dark side: " +
          (100*darkCount/totalCount) + "%");


}
```

Having completed the `JediKnight` class, you define a new class `JediTest` to test your implementation:

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

     // Your code for part (d) here (see instructions below)
```

d

```
JediKnight a = new JediKnight("Abdallah", true);
JediKnight b = new JediKnight("Elana", true);
JediKnight c = new JediKnight("Fernando", false);
```

```
     // Your code for part (e) here (see instructions below)
```

e

```
JediKnight.print();
```

```
          System.exit(0);
     }
}
```

d. Write code in the `main()` method of `JediTest` above to instantiate three `JediKnights`, two on the Light side of the Force, and one on the Dark side of the Force (you may name them whatever you like). Put your answer in the box labeled *d* above.

e. Write code in the `main` method of `JediTest` above to invoke the static method you wrote in part c (*i.e.*, the `print()` method). Put your answer in the box labeled *e* above.

f. Indicate the expected output of running your `main()` method below.

f

```
Total Jedi knights on dark side: 1
Percentage of Jedi knights on dark side: 33%
```