1.00 Introduction to Computers and Engineering Problem Solving

Quiz 1 March 7, 2003: SOLUTION

Name:	Teaching Assistant
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	/ 14
Question 2	/ 9
Question 3	/ 9
Question 4	/ 33
Question 5	/ 35
Total	/ 100

Problem 1. (14 points)

```
public class Quiz1 {
  private static int quizCount = 0;
  private final int pageCount;

  public Quiz1(int pc) {
     pageCount = pc;
  }

  public void printInfo() {
   // omitted
  }

  public static void printStats() {
   // omitted
  }

  public void partF() {
     int x = 5, y = 10;
     double z = (double)(x/y);
     System.out.println(z);
  }
}
```

Refer to the code above when answering the true or false questions below.

Circle TRUE or FALSE for each of the following statements:

A. The static data member `quizCount` is associated only with a specific instance of the class.

TRUE [✓ **FALSE**]

B. The `final` data member `pageCount` cannot be changed after it is initialized.

[✓ **TRUE**] FALSE

C. The non-static method `printInfo` can access static data members of the class.

[✓ **TRUE**] FALSE

D. Only the class's own methods have access to its private data members `quizCount` and `pageCount`.

[✓ **TRUE**] FALSE

E. You can use the `this` keyword in both static and instance methods, *e.g.*, in `printInfo()` and `printStats()`.

TRUE [✓ **FALSE**]

F. The method `partF()` will output "0.5":

TRUE [✓ **FALSE**]

G. A non-final static variable, such as `quizCount`, can be read and modified by any instance of the same class.

[✓ **TRUE**] FALSE

Problem 2. (9 points)

Consider the three code fragments below.

```
// Fragment X
for (int i=0; i <5; i++)
     System.out.println(i);
```

```
// Fragment Y
int i=0;
while (i<5) {
     System.out.println(i);
     i++;
}
```

```
// Fragment Z
int i = 0;
do {
     i++;
     System.out.println(i);
} while (i<5);
```

Write the output of each code fragment in the space below. (Don't be concerned with whitespace.)

A. Output of fragment X

```
0
1
2
3
4
```

B. Output of fragment Y

```
0
1
2
3
4
```

C. Output of fragment Z

```
1
2
3
4
5
```

Problem 3. (9 points)

Consider the condition of an `if` statement:

```
boolean b;
...
if (b == true) {
  /* do something */
}
```

Circle the letter next to <u>all</u> of the expressions below that are equivalent to the conditional expression "`b==true`" in the statement above. There may be more than one answer.

```
A. b == false

B. b != true

C. !b
```

[✓ **`D. !(b == false)`**]

[✓ **`E. b != false`**]

[✓ **`F. b`**]

Problem 4. (33 points)

Consider the following class, which represents an electric lightbulb

```
     public class Bulb {
       private int wattage;         /* in watts */
       private int voltage;         /* in volts */
       private String color;        /* color of the light,
     e.g., "red" */

       public Bulb(int W, int V, String C) {
          wattage = W;
          voltage = V;
          color = C;
       }
      ... // other methods omitted
     }
```

A. Write the fragment of code below that will create a yellow, 150-watt bulb, at 115 volts. Make sure you properly declare and assign a variable to the object you create.

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

     Bulb b = new Bulb(150, 115, "yellow");


}
```

B. Add a public, no-argument constructor (that is, a constructor that has no parameters) to the `Bulb` class which assigns default values of 1 watt, 5 volts and white light to new bulbs.

```
public  Bulb() {
     this(1, 5, "white");
}

/* or */

public Bulb() {
  wattage = 1;
  voltage = 5;
  color = "white";
}
```

Now suppose we add a boolean member variable `isOn` to the `Bulb` class as follows:

```
     public class Bulb {
      // previous lines omitted
      ...
      // true if the bulb is on, false if the bulb is off
      private boolean isOn = false;

      ...
     }
```

C. Write a public method for the `Bulb` class named "`togglePower`" that works as follows. If the bulb is on, it is turned off. If the bulb is off, it is turned on. The method must return the new state of the bulb (`true` if on, `false` if off).

```
public boolean togglePower() {
     return isOn = !isOn;
}

/* or */
public boolean togglePower

     if (isOn==true) isOn = false;
     else isOn = true;

     return isOn;
}

/* many equivalent variations are also okay. */
```

Problem 5. (35 points)

A. Complete the following method to sum the numbers between 1 and n. For example, if this method were passed the argument 3 it would return 6 (since 1 + 2 + 3 = 6). You can assume that n will always be positive.

```
public static int sum(int n)
{
     int total = 0; /* must be initialized */

     for (int i = 1; i <= n; i++)
          total += i;

     return total;


     // this is also okay:
     // return n*(n+1)/2;

}
```

B. Complete the same method again, this time using an array of `int`s. This method must return the sum of all the elements in `arr`.

For example, if `arr` were the array `{ 3, 7, 2, 4 }`, then the `sum()` method would return 16, which is 3+7+2+4.

```
public static int sum(int[] arr)
{
    // Your code here
     int total = 0;
     for (int i = 0; i < arr.length; i++)
          total += arr[i];

     return total;

}
```