

What time does your lab section meet? \_\_\_\_\_

**Part I. Fundamentals of computing (Each question is worth five points.)**

1. Complete the truth table for the logical operation *same*. The logical operation *same* is true only when  $p$  and  $q$  have the same value.

$p$	$q$	$p \text{ nor } q$
false	false	
false	true	
true	false	
true	true	

2. Complete the truth table for the logical operation *(not p) or (not q)*.

$p$	$q$	$(\text{not } p) \text{ or } (\text{not } q)$
false	false	
false	true	
true	false	
true	true	

3. Consider the following syntactically correct code fragment.

```
double a = 1;
double b = 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1
          + 0.1 + 0.1 + 0.1 + 0.1;
if (a == b) {
    System.out.println("Values equal");
}
else {
    System.out.println("Values unequal");
}
```

What does the fragment output? Explain why.

4. Suppose the following definitions are in effect.

```
int i = 10;
int j = 20;
int k = 10;
```

Evaluate the following expressions and give the value. The first one is done for you.

- a) `i < j` true
- b) `i == k && j <= k` \_\_\_\_\_
- c) `j != k && i == k` \_\_\_\_\_
- d) `((i + 10) <= j) || i != k` \_\_\_\_\_
- e) `(j > k) && (i == k) && (j - 10 == k)` \_\_\_\_\_
- f) `(i == j) || (j > k) || (i != k)` \_\_\_\_\_

## Part II. Fundamentals of Java programming (Each question is worth five points.)

5. Consider the following code fragment

```
String s = new String("Wahoo");
String t = new String("Wahoo");
if (s == t) {
    System.out.println("Go Hoos!");
}
else {
    System.out.println("Beat GWU!");
}
```

6. What does the code fragment output?

```
Point p = new Point(5, 5);
Point q = new Point(5, 6);
Point r = q;
r.y = 5;
if (p.equals(q)) {
    System.out.println("same");
}
else {
    System.out.println("different");
}
```

What does this code fragment output?

7. Consider the following code fragment whose output is based on the values of **boolean** variables a, b, c, and d.

```
if (!a && b) {
    if (c && d) {
        System.out.println("Go Hoos!");
    }
    else if (d) {
        System.out.println("Beat Colonials!");
    }
    else if (c) {
        System.out.println("Wahoo Wa!");
    }
} else if (a && !b) {
    if (c && !d) {
        System.out.println("Rah Rah");
    }
    else if (c && d) {
        System.out.println("UVa!");
    }
}
```

- a) Give values for a, b, c, and d that cause the code fragment to display “Go Hoos!”.

a \_\_\_\_\_ b \_\_\_\_\_ c \_\_\_\_\_ d \_\_\_\_\_

- b) Give values for a, b, c, and d that cause the code fragment to display “Beat Colonials!”.

a \_\_\_\_\_ b \_\_\_\_\_ c \_\_\_\_\_ d \_\_\_\_\_

- c) Give values for a, b, c, and d that cause the code fragment to display “Wahoo Wa!”.

a \_\_\_\_\_ b \_\_\_\_\_ c \_\_\_\_\_ d \_\_\_\_\_

- d) Give values for a, b, c and d that cause the code fragment to display “Rah Rah”.

a \_\_\_\_\_ b \_\_\_\_\_ c \_\_\_\_\_ d \_\_\_\_\_

- e) Give values for a, b, c and d that cause the code fragment to display “UVa!”.

a \_\_\_\_\_ b \_\_\_\_\_ c \_\_\_\_\_ d \_\_\_\_\_

Questions 8 – 9 consider the following code segment.

```
if ( i == 4 ) {
    System.out.println("A");
}
else if ( i <= (j - 1) ) {
    System.out.println("B");
}
else {
    if ( (i / j) < 6 ) {
        System.out.println("C");
    }
    else {
        System.out.println ("D");
    }
}
```

8. If  $i$  is 5 and  $j$  is 4, what is the output?

9. If  $i$  is 6 and  $j$  is 7, what is the output?

10. Consider the following code segment.

```
int counter1 = 0;
int counter2 = 0;
int counter3 = 0;
for (int i = 0; i < 2; ++i) {
    ++counter1;
    for (int j = 1; j < 4; ++j) {
        ++counter2;
        for (int k = 0; k < 4; ++k) {
            ++counter3;
        }
    }
}
System.out.println(counter1 + " " + counter2 + " " + counter3);
```

What output does it display?

\_\_\_\_\_

Question 11 involves the following code fragment.

```
BufferedReader stdin = new BufferedReader(
    new InputStreamReader(System.in));

System.out.print("Filename: ");
String filename = stdin.readLine();

BufferedReader file = new BufferedReader(
    new FileReader( filename ) );

int total = 0;
for (String s = file.readLine(); s != null; s = file.readLine()) {
    int k = Integer.parseInt(s);
    if ( k > 0 ) {
        total += k;
    }
}
System.out.println("Total is " + total);
```

11. If the file specified by the user contains the following lines

```
34
-15
10
^Z
```

then what does the fragment output? Recall that ^Z is the end-of-file indicator

Total is \_\_\_\_\_

12. Write a **boolean** expression that evaluates to true only when the value of variable  $x$  is greater than 5 and less than 10.

13. Rewrite the following while loop as an equivalent for loop.

```
i = 0;
while (i > 0) {
    f(i);
    ++i;
}
```

**Part III. Problem solving (Each question is worth 10 points.)**

14. Complete the following program that reads a file that contains one integer per line. Your code segment should the number of positive values, the number of zero values, and the number of negative values in the file. After processing the file of integers, the program displays the number of values in each category. A sample output would be

```
file has 10 positive values, 5 zero values, and 3 negative values
```

Here is the first part of the code

```
import java.io.*;
public class Counter {
    public static void main(String[] args) throws IOException {
        BufferedReader stdin = new BufferedReader(
            new InputStreamReader(System.in));

        System.out.print("Filename: ");
        String filename = stdin.readLine();

        BufferedReader file = new BufferedReader(
            new FileReader( filename ) );

        // Read values and count the number of positive values,
        // zero values, and negative values
```

The remaining questions require you to both use and implement some additional member methods for the class `DataSet` considered in lecture.

15. Implement a new specific constructor that creates a data set with a specified set of values without reading a file. The parameters to the constructor are the size of the data set, the sum of the elements of the data set, and the minimum and maximum values of the dataset. This constructor could be useful for testing. As an example of the use of the constructor, the following code segment

```
Dataset d = new DataSet(10, 55.0, 4.0, 16.0);
Dataset e = new DataSet(5, 10.0, 1.0, 3.0);
System.out.println("d's Minimum: " + d.getMinimum());
System.out.println("d's Maximum: " + d.getMaximum());
System.out.println("d's Mean: " + d.getAverage());
System.out.println("d's Size: " + d.getSize());

System.out.println("\ne's Minimum: " + e.getMinimum());
System.out.println("e's Maximum: " + e.getMaximum());
System.out.println("e's Mean: " + e.getAverage());
System.out.println("e's Size: " + e.getSize());
```

would output

```
d's Minimum: 4.0
d's Maximum: 16.0
d's Mean: 5.5
d's Size: 10

e's Minimum: 1.0
e's Maximum: 3.0
e's Mean: 2.0
e's Size: 5
```

16. Write a **public** string parameterless member method `toString()` that returns a string representing the `DataSet`. As an example of method `toString()`'s use, consider the following code segment.

```
DataSet d = new DataSet();  
d.addValue(4); d.addValue(1); d.addValue(6); d.addValue(2);  
System.out.println("d is " + d);
```

This segment would output

```
d is [n = 4, xSum = 13, min = 1, max = 6]
```