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

1. Complete the truth table for the logical operation *and*.

p	***q***	***p and q***
false	false	
false	true	
true	false	
true	true	

2. Complete the truth table for the logical operation *exclusive or*. The operation *p exclusive or q* is true when exactly one of p and q is true.

p	***q***	***p exclusive or q***
false	false	
false	true	
true	false	
true	true	

3. Which statement is *false*.
   a) There are `while` loops whose bodies (actions) are executed at least once.
   b) There are `while` loops whose bodies are never executed.
   c) There are `do-while` loops whose bodies are executed at least once.
   d) The test expression of a `for` loop is always executed at least once.
   e) None of the above

4. Are `for` loops truly needed in a programming language that provides a `while` loop? Why?

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

5. The == operator when applied to reference variables tests whether the variables refer to the same memory location.

   a) true
   b) false

6. When evaluating a logical operation, Java always evaluates the right operand before it evaluates the left operand. If the value of the operation can be determined from the right operand, then the left operand is not evaluated.

   a) true
   b) false

7. What does the `String` method `equals`() test?

   a) Whether two string objects have the same representation
   b) Whether a string object has been initialized with an assignment
   c) Whether two string variables have the same value
   d) All of the above
   e) None of the above

Questions 8 – 9 consider the following code segment.

```
if ( i == 2)  {
   System.out.println("A");
}
else if ( i < (j - 1) ) {
   System.out.println("B");
}
else {
   if ( (i / j) < 3 ) {
      System.out.println("C");
   }
   else {
      System.out.println ("D");
   }
}
```

8. If i is 9 and j is 2, what is the output?

9. If i is 5 and j is 6, 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 = 0; j < 4; ++j) {
      ++counter2;
      for (int k = 0; k < 2; ++k) {
         ++counter3;
      }
   }
}
System.out.println(counter1 + " " + counter2 + " " + counter3);
```

What output does it display?

Questions 11 – 12 consider the following code segment.

```
BufferedReader stdin = new BufferedReader(
      new InputStreamReader(System.in));
String s = "null";
while (s != null) {
   s = stdin.readLine();
   System.out.println("looping");
}
```

11. If standard input contains

```
1
```

then "looping" is displayed

a) 0 times

b) 1 times

c) 2 times

d) 3 times

e) None of the above

12. If standard input contains

```
abc
                            <----- Blank line
def
```

then "looping" is displayed

a) 0 times

b) 1 times

c) 2 times

d) 3 times

e) None of the above

13. Write a `boolean` expression that evaluates to true only when the value of variable `x` is at least 2 and is at most 5.

14. Consider the following code segment. Why does it not compile?

```
int n = 5;
for (int i = 0; i < n; ++i) {
   --n;
}
System.out.println(i);
```

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

15. Complete the following code segment that displays the integers in the user-specified interval `m` … `n`. You may assume that the inputs are correctly supplied.

```
BufferedReader stdin = new BufferedReader(
      new InputStreamReader(System.in));
int m = Integer.parseInt(stdin.readLine());
int n = Integer.parseInt(stdin.readLine());
```

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

16. Complete the following code segment that is to read values from standard input and adds them to the DataSet associated with variable d. The process completes when a sentinel value of -1 is read. The sentinel value is not to be added to the DataSet.

```
BufferedReader stdin = new BufferedReader(
      new InputStreamReader(System.in));
DataSet d = new DataSet();
System.out.println("Enter a list of numbers, one per line");
```

17. Write a public parameterless double method spread() that returns the difference between the minimum and maximum values in the data set being represented. It is expected that your implementation makes use of other DataSet methods. As an example of spread()'s use, the following code segment should display 5.

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

Pledge