CS 61BL Summer 2019 — Lab 3

June 27, 2019

Name:	SID:

Section Number: 01 02 03 04 05 06 07 08 09 10 11 12

Write your name and login above. Please complete this worksheet during your lab, and turn it in to your TA by the end of your section. You are encouraged to work with your partners and neighbors collaboratively.

1 Date Converter

1.1 (a) In the left column, list out a variety of different inputs that you could provide into 'DateConverter.java'. They should test different aspects of the expected behavior and edge cases. In the right column, list out the answer you would expect the program to provide if it is working correctly.

Input	**Output**

(b) Did they all work as expected?

(c) Can you think of a way that your tests might not have been comprehensive (meaning that all your tests pass but there is still some type of input which will not work the way you want it to)?

2 `while`-to-`for` Translation

2.1 Translate the following `while` loops to `for` loops. The body of the `for` loops should contain only the call to `println`. Either do these with your partner, or compare notes with them after you complete them.

```java
int k = 0;
while (k < 10) {
    System.out.println(k);
    k = k + 1;
}
```

(b)

```java
int k = 0;
while (k < 10) {
    k = k + 1;
    System.out.println(k);
}
```

3 Nonstandard `for` Loop

3.1 What output is produced by the following program segment? We strongly recommend that you use a table to keep track of the value of `k`.

```
for (int k=1; k<10; k=k+1) {
    // increment is not normally done in both places
    k = k + 1;
    System.out.print(k + " ");
}
System.out.println();
```

4 Practice with Arrays of Objects

4.1 What is printed by the program below?

```
import java.awt.*;

public class Test {
   public static void main (String[] args) {
     Point[] line1, line2;
     line1 = new Point[2];
     line2 = new Point[2];
     line1[0] = new Point();
     line1[1] = new Point();
     line1[0].x = 1;
     line1[0].y = 3;
     line1[1].x = 7;
     line1[1].y = 9;
     line2[0] = line1[1];
     line2[1] = line1[0];
     line1[0].x = 11;
     line1[1] = line1[0];
     System.out.println(line2[0].x + " " + line2[0].y +
                           " " + line2[1].x + " " + line2[1].y);
   }
 }
```

5 Recognizing Purpose

5.1 In the real world, often you will have to deal with code you did not write. Sometimes you will be provided documentation, sometimes you will not. Being able to recognize what purpose code serves is an important skill. Provide a good (read: descriptive) name for each of the following methods. Assume that values contains at least one element.

As you work through each problem, spend some time reflecting on your problem solving process. Some questions you might ask yourself and your partner include:

- How did you go about solving each problem?
- Did you come up with a few representative examples?
- Did you trace through the code?
- How did you and your partner keep track of the different variable?
- Which variables were easy to identify, and what made them easy to identify?

(a)

```
private static int _____________________ (int[] values) {
    int rtn = values[0];
    int k = 1;
    while (k < values.length) {
        if (rtn < values[k]) {
            rtn = values[k];
        }
        k++;
    }
    return rtn;
}
```

(b)

```
private static void _____________________ (int[] values) {
    int k = 0;
    while (k < values.length / 2) {
        int temp = values[k];
        values[k] = values[values.length - 1 - k];
        values[values.length - 1 - k] = temp;
        k = k + 1;
    }
}
```

(c)

```
private static boolean ____________________ (int[] values) {
    int k = 0;
    while (k < values.length - 1) {
        if (values[k] > values[k + 1]) {
            return false;
        }
        k = k + 1;
    }
    return true;
}
```

(d)

```
private static int ____________________ (int[] values, int a) {
    int k = 0;
    int n = 0;
    while (k < values.length) {
        if (values[k] == a) {
            n++;
        }
        k++;
    }
    return n;
}
```