CSCI 1933 Lab 6
Midterm 1 Review

Rules

This lab is to be completed without use of a computer. **You must use only a pen and paper to complete these steps.**

This lab is to be done individually. If you complete the lab before the end of the session then you may be checked off by a TA. Due to the midterm this week, you may be checked off for completing the entire lab or staying for the entire lab time. No portion of this lab can be checked off during office hours, but feel free to come in with questions!

1 Class Design

In this step you must write the methods to complete a `Bus` class. Assume that the Passenger class has already been written.

- Each bus has a capacity. A default constructor should initialize the capacity to 40, and another constructor should allow the programmer to specify any positive capacity.
- An `addPassenger(Passenger p)` method should add the given `Passenger` object to the bus. Passengers can't be added if the bus is full. *Hint: what new member variable is needed?*
- A `numberOfPassengers()` method should return the current number of `Passengers` on board the bus.

2 Time Complexity

What does the following method accomplish, and what is the time complexity, in Big-O notation, of the method.

```
public boolean mysteryMethod(int[] array, double x) {
        for (int i = 0; i < array.length; i++) {
                for (int j = 0; j < array.length; j++) {
                        if (i != j && array[i] + array[j] == x) return true;
                }
         }
         return false;
}
```

What is the time complexity, in Big-O notation of the following methods.

```
public void mystery2(double x) {
        for (int i = 0; i < 20; i++) {
                for (int j = 0; j < 5; j++) {
                        System.out.println(x*j + i);
                }
         }
}

public void mystery3(int x) {
        for (int i = 0; i < x*x; i++) {
                System.out.println(i);
         }
}

public void mystery4(int x) {
        for (int i = 0; i < x+x; i++) {
                System.out.println(i);
         }
}

public int mystery5(int[] array, int x) {
        return array[x];
}
```

3 Debugging

The following snippets of code all have at least one bug. Write down fixed code, and explain to a TA what the bugs are.

```
// Constructs a "Whatever" object
private int data;
public Whatever(double data) {
        data = data;
}
public static void setData(int newData) {
        data = newData;
}
```

```
// Computes the sum of the numbers the user enters
Scanner s = new Scanner(System.in);
int sum = 0;
while (s.hasNext()) {
        String data = s.next();
        if (data == "stop") {
                break;
        }
        sum += data;
}
System.out.println("The sum is:" + sum);
```

```
//Find the error(s) that causes this code to crash.
Object a = new Object();
Object b = null;
boolean b1 = a.equals(b);
boolean b2 = a.equals(null);
boolean b2 = b.equals(a) || b == null;
boolean b4 = b == null || b.equals(a);
```

4 Code Comprehension

Give the output (`System.out`) of the following `main` method.

```
public class Whatever {
    private int data;

    public Whatever(int newData) {
        data = newData;
    }

    public void setData(int newData) {
        data = newData;
    }

    public int getData() {
        return data;
    }
```

```
    public static void doWhatever(Whatever w, int i, int d) {
        System.out.println("doWhatever(1): w: "+w.getData()+", i: "+i+"; d: "
            +d);
        w.setData(i);
        d = i;
        i = d;
        System.out.println("doWhatever(2): w: "+w.getData()+", i: "+i+"; d: "
            +d);
    }

    public static void main(String[] args) {
        Whatever w = new Whatever(1);
        int i = 2;
        double d = 3;
        System.out.println("main(1): w: "+w.getData()+", i: "+i+"; d: "+d);
        doWhatever(w, i, (int)d);
        System.out.println("main(2): w: "+w.getData()+", i: "+i+"; d: "+d);
        w = new Whatever(i);
        d = i / 4;
        System.out.println("main(3): w: "+w.getData()+", i: "+i+"; d: "+d);
    }
}
```

5 Recursion

Write a **recursive** method `public static int powerK(int k, int n)` that takes in integers k and n and returns k^n.

Gopher It

The following problem is optional extra practice. An additional 2 sample midterms as well as a topics sheet are posted on canvas. Good luck on the exam!

Greatest Common Divisor

Write a method `public static int greatestCommonDivisor(int a, int b)` to return the greatest common divisor of two integers. The greatest common divisor of two positive integers a and b is defined as the largest positive number that divides both a and b. For example, the greatest common divisor of 18 and 12 is 6.