Name: ______________________________ Andrew Id: ____________________

15-121 Sample Assessment 1

Up to 50 minutes. No calculators, no notes, no books, no computers. Show your work!

1. **Short Answer**

   (a) (2 points) In just a few words, what's the difference between a class and an object?

   (b) (2 points) Write two lines of code to generate a random number between 5 and 15 (inclusive, meaning both 5 and 15 could be chosen) and store it in an integer named `myRand`.

   (c) (2 points) In just a few words, what's the difference between `public` and `private` when applied to instance variables?

2. (4 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box under the code.

```
public class IncrementorExercise {
        public static void main(String[] args) {
                int a = 2;
                int b = 3;

                System.out.println(--b - --b + a++ - ++b + a++ - b-- - ++b);
                System.out.println(a);
                System.out.println(b);
        }
}
```

3. (6 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box under the code.

```
public class Quiz3CT {
        private String[] hey = { "fox", "cat", "dog" };
        private int jude;

        public Quiz3CT(int a, int b) {
                int t = 0;
                for (int i = a; i < b + 1; i++) {
                        t += i / 3;
                }
                this.jude = t;
                System.out.println("W: " + t);
        }

        public void tweak() {
                this.jude = this.jude % hey.length;
        }

        public String toString() {
                System.out.println("D: " + this.jude);
                return this.hey[jude];
        }

        public static void main(String[] args) {
                Quiz3CT a = new Quiz3CT(5, 10);
                a.tweak();
                System.out.println(a);
        }

}
```

4. (12 points) **Free Response:** Write the function `sequentialSum` which, given a number range and a target sum, determines if there is a *consecutive* sequence of three numbers that sum to the target. If there is, it prints the three numbers and returns `true`. If there is not, it prints nothing and returns `false`.

Don't forget that the numbers need to be consecutive.

Consider the following three examples:

	Example 1	**Example 2**	**Example 3**
Method Call	`sequentialSum(1, 10, 9)`	`sequentialSum(1, 10, 24)`	`sequentialSum(1, 10, 27)`
Output	2 3 4	7 8 9	
Return Value	`true`	`true`	`false`

```java
/**
 * Try to find a consecutive sequence of 3 numbers in range [min,max) that sum
 * to target. If there is such a consecutive sequence, print all three numbers
 * and return true. If there isn't such a consecutive sequence, return false.
 *
 * @param min    The minimum value in the range (inclusive)
 * @param max    The maximum value in the range (exclusive)
 * @param target The value to sum to
 * @return true or false as specified above
 */
public static boolean sequentialSum(int min, int max, int target) {
```

5. **Free Response**

Consider the following skeleton code:

```
public class StringList {
    private String[] strArr;
    private int numItems;

    public StringList() {
        this.strArr = new String[10];
        this.numItems = 0;
    }

    /**
     * Adds `item` to the end of the list. If the list has space, then it simply
     * adds the item. If the list is full, then it first resizes it, making it twice
     * as large as before, then adds the new item. Ensures numItems is updated
     * appropriately.
     *
     * @param item The item to add to the list
     */
    public void append(String item) {
        // You will write this code
    }

    /**
     * Rotates all the elements in the array one position to the right.
     *
     * Example: if the array before the call to rotateRight is ["a", "b", "c"] it
     * is ["c", "a", "b"] after the call
     */
    public void rotateRight() {
        // You will write this code
    }

    /**
     * Rotates all the elements in the array n positions to the right.
     *
     * Example: if the array before the call to rotateRight(2) is ["a", "b", "c"]
     * it is ["b", "c", "a"] after the call
     */
    public void rotateRight(int n) {
        // You will write this code
    }
}
```

The question continues on the next page.

(a) (5 points) Write the `append` method specified above.

(b) (5 points) Write the `rotateRight()` method specified above.

(c) (2 points) Write the `rotateRight(int n)` method specified above. You may assume that you have a working implementation of `rotateRight()`, even if yours does not work.