

## Exam 2

CSCI 110 Section 1

Wednesday, October 26, 2016

Name: \_\_\_\_\_

Total: 1000 points

Take a deep breath before starting. When in doubt, make educated guesses. You've got this!

**Answers must be boxed for credit.**

Please leave this exam on your desk when you're finished.



- 1) Please fill in the following table. Write yes/no for first two rows and call methods or get attributes using the instance names provided for the last three rows. Put n/a if something doesn't exist. [50 points]

	<b>ArrayList</b>	<b>array</b>	<b>String</b>
<b>Instance name</b>	list	arr	s
<b>Can increase length?</b>			
<b>Mutable?</b>			
<b>Length</b>			
<b>Get element at index i</b>		arr[i]	
<b>Set element at index i to the value 45</b>			

- 2) What's printed when mystery(5) is called? [50 points]

```
void mystery(int n) {  
    System.out.println("tiger");  
    if (n <= 1) {  
        System.out.println("newt");  
        return;  
    }  
    mystery(n / 2);  
    System.out.println(n);  
}
```

For questions 3 and 4:

```
class Tiger {
    private double hunger;

    public Tiger(double hunger) {
        this.hunger = hunger;
    }

    public double getHunger() {
        return hunger;
    }

    public boolean willEatAntelope() {
        return hunger > 50.0;
    }

    public void eatFood(double food) {
        hunger -= food;
    }
}
```

3) What's printed after the following code is run? [50 points]

```
Tiger t1 = new Tiger(100.0);
System.out.println(t1.willEatAntelope());
t1.eatFood(50.0);
System.out.println(t1.willEatAntelope());
System.out.println(t1.getHunger());
t1.eatFood(50.0);
System.out.println(t1.getHunger());
```

Say the following two functions are defined along with the previous class.

```
void nature(Tiger t) {  
    t.eatFood(20.0);  
}
```

```
Tiger nurture(Tiger t) {  
    t = new Tiger(90.0);  
    return t;  
}
```

4) What's printed after the following code is run? [50 points]

```
Tiger t2 = new Tiger(60.0);  
System.out.println(t2.willEatAntelope());  
nature(t2);  
System.out.println(t2.willEatAntelope());  
t2 = nurture(t2);  
System.out.println(t2.willEatAntelope());
```

- 5) Please number the blanks from 1 through 9 in the order that each expression is evaluated. A blank may have two numbers. [100 points]

```
class SomeClass {  
  
    static void someFn() {  
  
        _____ System.out.println("ok");  
  
    }  
  
    static void otherFn() {  
  
        _____ System.out.println("now");  
  
    }  
  
    public static void main(String[] args) {  
  
        _____ System.out.println("goodbye");  
  
        _____ for (int i = 0; i < 1; i++) {  
            _____ System.out.println(25);  
        }  
  
        _____ someFn();  
  
        _____ System.out.println("done");  
  
    }  
  
}
```

- 6) Please complete the constructor, advance(), rewind(), and getTapeName() methods for the VCR class. [100 points]

```
class VCR {
    private double tapePosition;
    private String tapeName;

    /**
     * Constructs a new VCR with a given tapeName. Initializes
     * tapePosition to 0.
     */

    /**
     * Advances tapePosition by 10.
     */

    /**
     * Rewinds tapePosition to 0.
     */

    /**
     * Returns the name of the tape.
     */

}
```

- 7) Write a function to translate a pixel intensity to a description. Pixel intensities can be between 0 and 255. Intensities between 0 and 85 should be described as "light", between 86 and 170 should be described as "medium", between 171 and 255 should be described as "dark". Intensities outside of this range should result in "INVALID\_INTENSITY". [100 points]
- 8) Say there's a pre-defined function with the call signature "boolean isHappy(int num)" that returns true if a number is happy. Write a function named that returns the sum of the happy numbers between (including) 7 and (excluding) 700. [100 points]

- 9) Write a function named `oppositeDay` that inverts all of the values in an `ArrayList<Boolean>` i.e. turns `true` to `false` and `false` to `true`. This function shouldn't return anything. [100 points]
- 10) Say that monthly listener counts on Spotify are represented as an array of `long`. We want to take this array and find how many people listen to the most and least popular artists. Write a function named `findDrakeAndMeek` that finds both the min and max of an array of `long` and returns the sum of the min and max. [100 points]



11) Write a function that takes an array and returns a new array with two copies of the original, one after the other. For instance, if this function was given [6, 2, 3, 4], it should return [6, 2, 3, 4, 6, 2, 3, 4]. [100 points]

12) Write a function named `bullseye` that takes a 2D array of ints. The outer array always has an odd number of elements. Each inner array always has the same odd number of elements. `bullseye` should set the middle element of the middle array to be 1. [100 points]

Example:

```
int[][] grid = {{0, 0, 0, 0, 0},
                {0, 0, 0, 0, 0},
                {0, 0, 0, 0, 0},
                {0, 0, 0, 0, 0},
                {0, 0, 0, 0, 0}};
```

```
bullseye(grid);
```

```
// grid now contains:
//      {{0, 0, 0, 0, 0},
//      {0, 0, 0, 0, 0},
//      {0, 0, 1, 0, 0},
//      {0, 0, 0, 0, 0},
//      {0, 0, 0, 0, 0}}
```

13) Write a function named `makeWeirdCase` that takes a `String` and returns a `String` with the first letter lowercase, the second letter uppercase, the third letter lowercase, and so on. For example, if the function is given "Cheetah", it should return "cHeEtAh". [extra credit, 100 points]

14) Write a function that determines if a given string can be obtained by one concatenation of some string to itself. For example, "lionlion" can be obtained by appending "lion" to itself. "www" is an example of a `String` for which there's no way to do this. [extra credit, 100 points]