

ECE 558 Fall 2018
Homework #1

THIS ASSIGNMENT SHOULD BE SUBMITTED TO D2L BY 10:00 PM ON SAT, 06-OCT-2018. THE ASSIGNMENT WILL BE GRADED AND IS WORTH 100 POINTS. IT IS EASIEST TO GRADE, AND I BELIEVE EASIEST FOR YOU SUBMIT, IF YOU SUBMIT YOUR SOURCE CODE AS TEXT FILES INSTEAD OF TRYING TO CUT/PASTE YOUR CODE INTO A .doc OR .docx FILE. CLEARLY NAME THE FILES SO THAT WE KNOW WHICH PROBLEM THE CODE REFERS TO AND SUBMIT THE PACKAGE AS A SINGLE .ZIP OR .RAR FILE (EX: rkravitz_ece558f18_hw1.zip) TO YOUR D2L HOMEWORK #1 DROPBOX

Question 1 (35 pts): Object Oriented Programming

In this problem we are going to write a class encapsulating the concept of coins, assuming the coins have the following attributes: a number of quarters, a number of dimes, a number of nickels, and a number of pennies. We have made the problem simpler for you to write by providing starter code w/ TODO's for the members of the class. The test client is yours to write.

- a. [5] Write getters (accessors) and setters (mutators) for your instance variables. Your setters should not change the value if the new value is a negative number.
- b. [5] Write a constructor for the class that accepts initial values for the number of quarters, dimes, nickels and pennies. Make use of the setters you wrote in part a.
- c. [5] Write a method that returns the total amount of money in dollar notation with two significant digits after the decimal point (ex: \$4.27).
- d. [5] Write four additional methods that return the value of each of the 4 coin types. For example, a method called `moneyFromQuarters()` that returns 0.75 if there were three quarters.
- e. [15] Write a client class to test all of the methods in your class.

Question 2 (25 pts): Java Basics

a. [15] Correct the following code snippets:

- The following code sequence is intended to print Hello four times; however, it only prints Hello once. Where is the problem in this code sequence?

```
for (int i = 0; i <= 3; i++);  
    System.out.println("Hello");
```

- You coded the following in:

```
int a = 47;  
a += 13;  
System.out.println("The value of a is " + a);
```

You expected the value of a to be 60, but instead a was displayed as 13. Explain what the problem is and write the code to fix it.

- You coded the following:

```
int[][] a = {{3,2,1,0},{5,10,15,20}};  
for (int j = 0, j <= a[1].length; j++) {  
    if (a[1][j] == 20) {  
        System.out.println("Found 20 at column index " + j +  
            + " of second row");  
    }  
}
```

The code compiles properly, but when you run the program you get the following output:

Found 20 at column index 3 of second row

Exception in thread "main"

*java.lang.ArrayIndexOutOfBoundsException: 4 at
Test.main(Test.java:14)*

Explain what the problem is and how to fix it.

- b. [10] Write a method that takes an integer input from the user, then prompts for additional integers and keeps track of the sum of all of the integers. When the user enters a negative number the program should print out the sum of all of the integers. Do not include the negative number in the sum.

Question 3 (15 pts): Subclasses, Inheritance, etc.

a. [5 pts] Consider the following class definition:

```
public class Store {
    public final double SALES_TAX_RATE = 0.06;
    private String mName;

    public Store(String newName) {
        setName(newName);
    }

    public String getName() {
        return mName;
    }

    public void setName(String name) {
        mName = name;
    }

    public String toString() {
        return("name: " + mName);
    }

    public boolean equals(Object o)
    {
        if (!(o instanceof Store))
            return false;
        else
            Store s = (Store) o;
        return (mName.equalsIgnoreCase(s.name));
    }
}
```

Define a class encapsulating a music store, which inherits from Store. Code the constructor and the toString() and equals() methods of this new class. A music store has the following additional attributes:

- The number of titles it offers
- The address of the store

- b. [10 pts] Write an *abstract* superclass encapsulating some food; the food class has two attributes:
- Description of the food
 - Number of calories per serving

The class also has an abstract method taking a number of servings as a parameter and returning the number of calories.

This class has two non-*abstract* subclasses: one encapsulating a liquid food (such as a drink), and the other encapsulating a fruit (such as a pomegranate). A liquid food has an additional attribute: its viscosity. A fruit has an additional attribute: its season. Write these two subclasses to implement the *abstract* method defined in its superclass and provide getters and setters for the additional attributes.

Question 4 (25 pts): Java Programming Project

In cryptograms each character is encoded into another. If the text is long enough, one can, as a strategy, make use the frequency of occurrence of each character. In English text the most frequently occurring character will likely be the code for an *e*, because *e* is the most frequently used letter of the English alphabet.

- a. [10 pts] Design and implement a class that attempts to determine the relative frequency of each letter. Input should come from a text file (the name of the text file can be hardcoded into the program if that's simpler). As the file is read character-by-character the code that reads the file should keep track of the number of times each of the characters appears. Besides reading the file (which could be done in the constructor or as a separate method), the class should also provide these two methods:

- `highestFrequencyCharacter()` – returns the character code for the character that appeared the most times in the text.
- `lowestFrequencyCharacter()` – returns the character code for the character that appeared the least times in the text.

Your class may also want to implement a method that returns the number of times a character (passed as a parameter to the method) occurs (ex: `int getNumOccurrences(char c)`)

- b. [10 pts] Write a client test program (includes `main()`) that tests your class. Make sure your test client checks all of your methods and that you created. You may also want to test more than one text file. We have provided a simple text file that contains all the letters, but you could do better than that.
- c. [5 pts] Submit the source code that you wrote for the project and a transcript (ex: a console log file) showing that your program works.