CECS 220
Summer 2019
Assignment 4
July 2nd, 2019

Due Date: Sunday, July 14, 2019, 23:59
Total Points: 20

1. **(12 points)** Solve the following problem.

   (a) Define an abstract class, called *UtilityCustomer*, that has one Integer instance variable, an account number, and an abstract method, called *calculateBill*, that returns the bill amount as double. The *UtilityCustomer* class implements the *Comparable* interface, and includes a constructor that is passed a parameter for initializing the instance variable. It includes also accessor and mutator methods for the instance variable, and a *toString* method that returns a string for displaying the account number. The *UtilityCustomer* class has two non-abstract subclasses that inherit from the *UtilityCustomer*: *GasCustomer* and *ElectricCustomer*.

   (b) *GasCustomer* class has two additional fields, the *cubicMetersUsed* instance variable, and a constant for the price of gas per cubic meter of $2.75. Include appropriate constructor for initializing the class instance variable, accessor and mutator methods for class's instance variable, and implement the *calculateBill* method. Write a *toString* method, that calls the *toString* method of the *UtilityCustomer* a formatted string to display the gas customer's account number, the gas consumption, and the amount charged.

   (c) *ElectricCustomer* class has three additional fields, *kWattHourUsed* instance variable, along with two constants, one for the price of electricity per kilowatt hour, and the other for a flat power delivery fee of $30 that is added to every bill. Include appropriate constructor for initializing the class instance variable, and accessor and mutator methods for class's instance variable. Write a *toString* method, that calls the *toString* method of the *UtilityCustomer*, and implement the *calculateBill* method.

   (d) Write a client class, called *CollectionOfUcustomers*, that maintains a list of *UtilityCustomer* objects created by a user. The client class prompts a user for the type of a *UtilityCustomer* and its corresponding parameters to create. The program allows the user to specify up to 10 *UtilityCustomer* objects to be stored in an array. The program should display the information of *UtilityCustomer* objects in descending order based on their account numbers.

2. **(8 points)** Write a JavaFX application that allows a user to enter the food charge for a meal at a restaurant using a text field. The application provides the user with four options for tip percentages to select from: 0%, 15%, 18%, and 20%. You can use controls for determining the tip percentage such as radio buttons, or choice box. The application provides a button that is used to calculate and display the amount of tip on the meal charge, an 8% sales tax amount, and the total of all three amounts using Text objects. You can choose any layout to organize the control elements. **(Note: Radio buttons are introduced in section 5.10, and choice boxes are introduced in section 8.9.)**

Submission Requirements:
Your presentation in your report reflects great deal about you, your understanding of the assignment and on how much this course means to you. I try very hard to look at the substance of the report but I will be lying if I said that presentation does not influence my judgment. So, I expect your reports to be well organized and conform to the following rules:

All reports must be submitted in PDF format. Each assignment should contain the following:

1. Title page with your name, assignment number and the day you are actually submitting this report (Not the assignment due date).

2. A brief description of your solution of each problem of the assignment separately, you can also explain your solution using a pseudocode. *Number your descriptions according to the problem numbers.*

3. A comprehensive set of snapshots showing the inputs submitted, outputs obtained in the case of a successful output or a failure, including required output formatting, prompts, and messages.

4. Java source files that contain your solutions. It must be a "*.java" file. Source programs should contain meaningful comments and variable names.

5. Please zip both the PDF document with the source code and submit one zipped file. Please name your zipped file as "HWx_firstname_lastname.zzz". Where, "firstname" and "lastname" refer to your first and last names, "x" refers to the homework number (e.g., 1, 2, etc), "zzz" refers to the file name extension for the software used for archiving.

6. Submissions after the due date are accepted with a penalty of 25% per day (weekend days are counted as one-day delay).

Grading Table Summary

Problem	Item		Points
Problem 1	Report (Description of solution, discussion, & analysis)		1
	Output snapshots (input/output, formatting)		1
	Java Source Code Implementation	*UtilityCustomer* Class	2
		GasCustomer Class	2
		ElectricCustomer Class	2
		CollectionOfUcustomers Class	4
Problem 2	Report (Description of solution, discussion, & analysis)		1
	Output snapshots (input/output, formatting)		1
	Java Source Code Implementation	start Method	4
		Handler Methods	2
	Total		20