CIS 611
Spring 2017
Individual Programming Assignment: PA06 – Class Inheritance, abstract classes & interfaces

Due**:** Friday Mar 24, 2017 11:59pm

Total Points: 20

Class Inheritance, abstract classes & interfaces

The purpose of this programming assignment is to:

- Develop subclass from super class through inheritance
- Override instance methods in the subclass
- Define abstract classes and classes that implement abstract methods

Question 1: (20 Points) Electricity Bill Calculation

Write a Java program to calculate monthly Electric bill using the following rate information.

		Rate Schedule	
		Residential customer	**Commercial customer**
Service & Facility charges	**Monthly base charge**	**$6.75**	**$10.75**
Energy charges, per kWh	**Winter months**	**$0.04604**	**$0.03920**
	Summer, Tier1 (First 500 kWh)	**$0.04604**	**$0.06450**
	Summer, Tier2 (All above 500 kWh)	**$0.09000**	

- Summer season is June 1 through September 30
- Winter season is October 1 through May 31

The program must prompt the user to input the following:

a) Type of customer i.e. 0 : Residential or 1 : Commercial
b) Name of the customer – First Name & Last name
c) Energy used (this can be decimal value) in kWh
d) Month (1 for Jan, 2 for Feb … 12 for Dec) for which billing is to be generated.

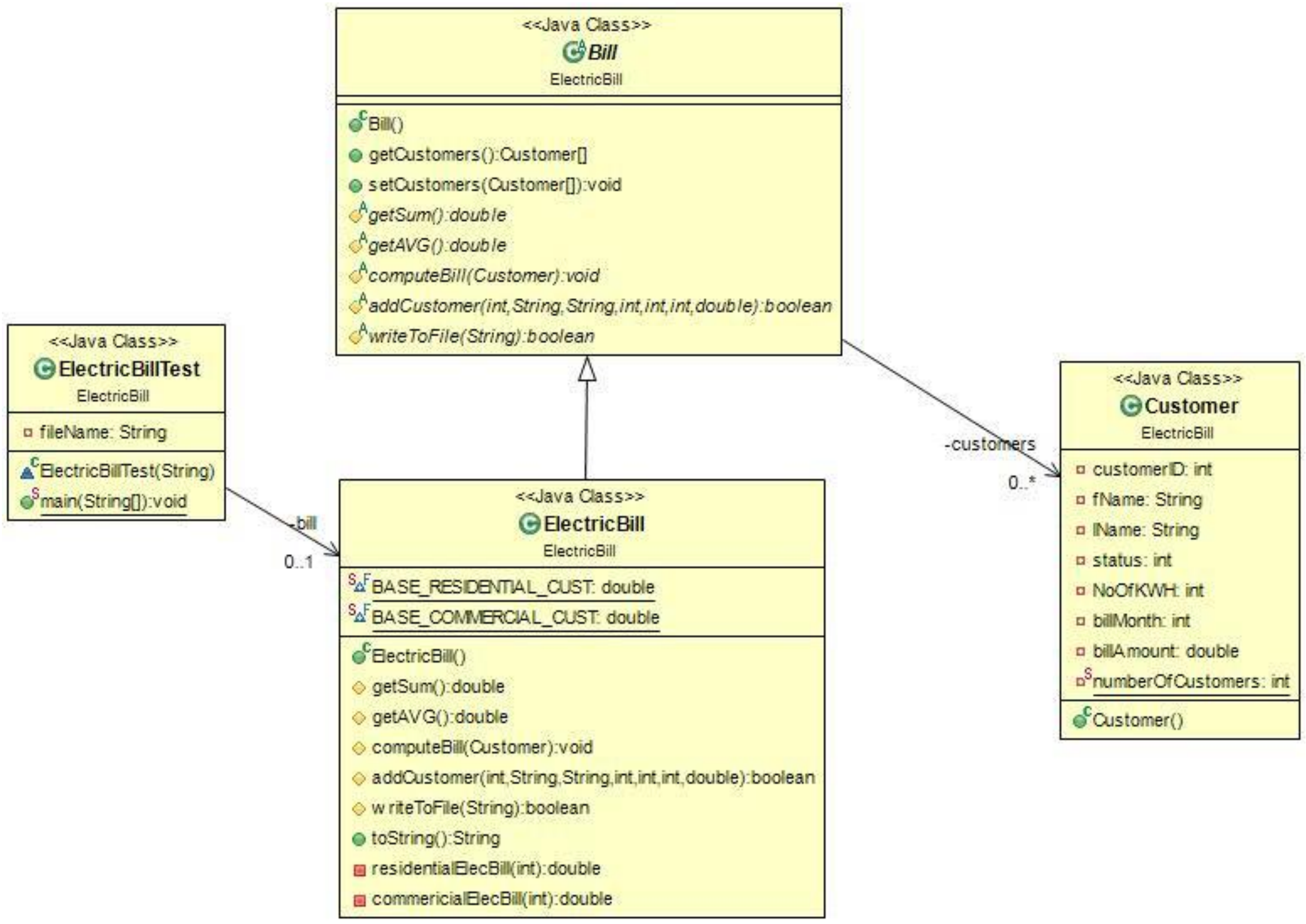
The bill amount is calculated as**: Monthly base charge + Charge based on Energy usage**

The calculated bill amount output must be rounded to 2 decimal places.

Use JOptionPane messages for the program input and output.

This Java program must implement the Java Application UML diagram and the associated source code files with the assignment document.
Complete the implementation of the methods' stubs in the provided Java source files.

You must maintain the same relationship between classes as shown in the UML class diagram and do not change the classes' implementation in a way violates the UML class diagram contract.
As shown in the UML diagram below, this Java application has 2 classes (***ElectricBill***, and ***Customer*** classes) and 1 abstract class ***Bill***.
Take a close look at the UML diagram in order to perceive the relationship between these classes.



ElectricBill class inherits from the ***Bill*** class and implements the abstract methods in ***ElectricBill*** class.

Once the program starts, it must read the user input to create 4 customer objects.
For every customer information, you must create a customer object and add it to the ***customers[]*** array in Bill class.
The customers' array size is 6, hence you must not exceed it in your program and so there must be appropriate check and error display when max array Capacity has been reached.

Then, the program displays the customer information as well as the current bills' total sum and average in the following format:

```
CustomerID    FirstName     LastName      Customer Status    NoOfKWH   Month   Bill Amount
44321584,    James,        Butt,         Residential,          350,    Jul,    22.86
51631223,    Josephine,    Darakjy,      Commercial,           700,    Jul,    55.90
27701760,    Lenna,        Paprocki,     Residential,          600,    Nov,    34.37

Sum of Bill Amount: 156.97     Average Bill Amount: 31.39
```

The program also stores the customer information in the above format (without header line & sum and average) in the ***customers.txt*** text file.

The program calculates the customer bill, only if the program inputs are correctly entered. The program must not crash with invalid or required data inputs. For example, the program must not accept 0 value customerID or customerID less than 8 digits, the number of kWH must be an integer number, a customer status must be 0 or 1, etc.
The JOptionPane class must be used to display invalid inputs' messages.

Program must adhere to the following specifications:

1. If the user does not provide valid input, the program must not abnormally abort, instead it must show a proper error message and allow the user to provide the required input.
2. Above requirement applies for invalid inputs.
3. The program must continue reading the inputs, calculating and displaying the output using **JOptionPane** showMessage box in the above format.
4. The program must have a correct logical order and output the anticipated result
5. The sequence, selection, and iteration structures must constitute correct program logic solutions to the assignment problem
6. The program must terminate gracefully as specified
7. The program must not abnormally abort with invalid/required user inputs
8. The program must terminate gracefully as specified
9. The program must not abnormally abort with invalid/required user inputs
10. You must handle invalid/required inputs in the program using try catch clauses or throws exceptions statements
11. You must follow the correct submissions format as described in this document

Evaluation Criteria:

1. You must use the class template in your program classes
2. The program must not have any compilation or runtime errors
3. All tasks must be completed to receive a complete credit for this assignment
4. The program must perform all the requirements correctly, including the read and output of data
5. The program must have a correct logical order and output the anticipated result
6. The sequence, selection, and iteration structures must constitute correct program logic solutions to the assignment problem
7. The program must terminate gracefully as specified
8. The program must not abnormally abort with invalid/required user inputs
9. You must handle invalid/required inputs in the program using try catch clauses or throws exceptions statements

10. You must follow the correct submissions format as described in this document

<u>Submission</u>

1. Zip all the java source files into one file that must be named following the provided naming format in this course, and then upload the zip file under this assignment answer in Canvas.

   Folder Name: *FFLLPA06.zip (where FF = your First Name, LL = your Last Name)*

   Summary for naming conventions for this assignment:

   Name: Jim Brown
   Zipped Folder Name: JimBrownPA06.zip

CIS611 – Spring 2017
PA06

Name: ____

Question 1

Requirements	Any comment provided by grader	Max Points Allowed	Points Earned
General Code Structure: Proper naming convention used for file, Comments used in the code to explain the purpose of the code, Indentation of the code for better readability, Good choice of variable names, implementation of the expected `classes`.		5	
Input, Output, User Interface: Proper coding implementation of the logic to read the data and display the expected value, proper coding implementation of dialog box/boxes, general aesthetics of user interface. Exception handling of the invalid input data (e.g., no data is entered, empty space is entered, invalid data is entered), Input of expected data and display of the expected data.		5	
General Algorithm and Logic: Proper instantiation of the abstract class, subclass (e.g., construction of Customer Class, abstract Bill Class, ElectricBill Class), override, construction of methods (e.g., getSum, getAVG, computeBill).		10	
		20	

Total ____/20