CPRG 200 Part 1: Lab Assignment 1

Due: 8 a.m. on Day 4

You may discuss your approach to solving this problem with other students, or ask for advice if stuck, but ultimately you should develop all the code on your own. You may use any code that you have previously developed or that was posted in this course.

Problem Description

Create a C# application that calculates a customer bill for the city power company. Calculation will be done for one customer at a time.

The rates vary depending on the customer type: residential, commercial, or industrial:

- Residential: $6.00 plus $0.052 for each kWh used ($6.00 is charged even when kWh usage is zero)
- Commercial: Flat rate of $60.00 for the first 1000 kWh used (or part of, even when zero), and $0.045 for each additional kWh
- Industrial: Rate depends on whether usage occurred during peak hours or off-peak hours:
    - Peak hours: Flat rate of $76.00 for the first 1000 kWh used (or part of, even when zero), and $0.065 for each additional kWh
    - Off-peak hours: Flat rate of $40.00 for the first 1000 kWh used (or part of, even when zero), and $0.028 for each additional kWh

Step 1

Create a Windows Form that allows user to input data of a residential customer and calculate the charge amount. There should be a prompt label and text box where user can enter kWh used by the customer. Also there should be a label or disabled text box where the calculated charge is displayed. No need to input customer name. Calculated bill amount for the customer should be displayed in currency format. Your application should have a minimum of three buttons:

- Calculate – triggers performing calculations
- Clear – erases input and output data in preparation for another calculation, and resets the controls to reflect the default residential customer type (as when the application starts)
- Exit – terminates the application

Step 2

Extend your application to make it possible to calculate charge for any type of customer. Add a control or controls to the form to allow selecting customer type.

Note that industrial customer requires second input, for the off-peak hours kWh. Also, the first input for an industrial customer has a different meaning: it is peak hours kWh rather that overall kWh usage. Make sure that when user selects customer type, only relevant controls are visible on the form, and that the prompt labels explain the input meaning accurately. You can accomplish it by changing Visible property of the controls and adjusting Text property of the labels.

Residential customer should be assumed as default as the program starts. That is the interface should have a look appropriate for the residential customer and the residential customer should be selected in the selection control. Focus should be on the input text box.

Step3

Validate user inputs to be non-negative whole numbers. Organize validation code in a separate method or methods that are called when needed to avoid code duplication.

Also, code that calculates charge based on kWh used for each customer type should be placed in separate methods. The methods receive as parameters kWh data necessary for the calculation. Each method returns calculated charge amount.

General requirements

You will present for marking only the resulting application after all three steps have been completed.

The look of the form is up to you (feel free to personalize it), but professional look and clarity/easiness to use will be a factor in marking.

Required comments:

- Top block comments that explains purpose of the application, date when created (at least month and year), and the author's name
- Each event method (button click) has a comment that explains its purpose
- Each variable has a comment that explains its meaning
- Each group of statements that perform a single task has a comment that explains the task

Zip the entire folder with your application and submit to the appropriate Dropbox. Make sure that the name of the folder includes your name and *CPRG200_Lab1*, e.g. *BobSmith_CPRG200_Lab2*.

The assignment must be submitted before the due date. Unless an important reason can be documented, late penalty deductions of 20% per day will be applied.

Marking Scheme

Marking Component	**Out of**
Form has all necessary controls, looks professional, and is easy to use	3
Only controls relevant for the current customer type are displayed	2
Selection of customer's type is processed correctly: relevant controls are displayed, and focus is on the input box	3
Input data is validated as required and code is placed in separate method(s)	5
Bill amount is calculated correctly for each customer type and calculation is done in separate methods as required	10
Bill amount is displayed in currency format	1
Button Exit terminates the application	1
Button Clear resets all controls to the starting stage	2
Code is clear, uses good naming practices, and has comments as required	3
Total:	**30**