Project 2

Description

Write a program that calculates a customer's monthly bill for a mobile data service.

Our mobile data service provider has three different packages for its customers:

> Package A: For $19.99 per month 2500 megabytes of data are provided. Additional megabytes are $0.05 per megabyte. However, if more than 75% of the data usage by the customer occurs on the weekend, the additional megabytes are only $0.03 per megabyte.
>
> Package B: For $29.99 per month, 5000 megabytes of data are provided. Additional megabytes are $0.01 per megabyte. There is no discount for weekend usage.
>
> Package C: For $39.99 per month, unlimited megabytes of data are provided.

You program should prompt which package the customer has purchased and how many megabytes are used. In addition, if the customer purchased Package A and has used more than the allowed 2500 megabytes, the program should additionally prompt if the customer used more than 75% of the data on the weekend. However, if the customer did not purchase Package A or did purchase Package A but used 2500 or less megabytes, do not ask the question about weekend usage.

When asking for the package, the program should expect a single upper or lower case A, B, or C. However, the program should tolerate invalid input from the user. In particular, if the user of the program just hits <enter> without providing an A, B, or C, the program should print a message to the user and terminate. Or, if the user enters a character other than A, B, or C, the program should print a message to the user and terminate. However, if the user enters additional text following the A, B, or C, the program should ignore that additional text and continue processing. For example, if the user enters "ASTUFF" as the purchased package, your program should interpret that input to mean the customer purchased package A and ignore the remaining text.

When asking for the number of megabytes used, the program should expect an integer. The program must ensure the entered value is non-negative. If the user enters additional text following the integer or enters a floating point number, the program should ignore that additional text or fractional value and continue processing.

When asking about weekend usage, the program should expect an upper or lower case Y or N. It should print a message if the user didn't enter either Y or N and should ignore any additional characters following the Y or N.

With all this input, the program should calculate the monthly charge for the user and display that value, with the $ currency symbol and always with two decimal places, to the screen.

Additionally, if the customer purchased Package A, the program should calculate the costs for Package B and Package C and display the savings if the customer purchased Package B and/or Package C if and only if Package B or Package C would save the customer money. If both Package B and Package C would save the customer money, display the savings for both packages. Or, if the customer purchased Package B, the program should calculate the cost for Package C and display the savings if the customer purchased Package C, if and only if, Package C would save the customer money.

The first line of the source code should be a comment including the following information:

```
// CS 1336.502 Project 2 <YOUR NETID> <YOURNAME>
```

Sample Run

This capture shows an example of running the program. Note the instructor has chosen to output additional information to aid the user and confirm what was input. That is acceptable.

```
Enter A, B, or C to indicate the plan purchased by the user:
   A - $19.99 per month with 2500 megabytes.
       $0.05 per additional megabyte, but only
       $0.03 per additional megabyte if 50% or
       more of the data was used on weekends
   B - $29.99 per month with 5000 megabytes.
       $0.01 per additional megabyte
   C - $39.99 per month for unlimited megabytes.
A
You selected Package: A
Please enter the amount of data used by the customer
Should be a whole, non-negative number): 9000
You entered: 9000 megabytes
Did the customer use 75% or more of the data
 on the weekend? (Y/N)n
Customer will receive a weekend discount: false
The customer's cost for the month is $344.99
Customer would have saved $275.00 on Package B
Customer would have saved $305.00 on Package C
```

Submitting the assignment

Submit only the source code (e.g., all cpp files) and header files you write for your program for the program. You must submit these to eLearning in response to this assignment. If there are multiple files, you may combine them into an archive file (e.g., zip, tar, 7z).

As part of this assignment, you may be required to demonstrate the following to the grader:

- If required by the grader, downloading the source file from eLearning.
- If required by the grader, building the program in Eclipse with MinGW. You must either bring your laptop with Eclipse and MinGW installed on Windows or coordinate with the grader to use a UTD machine configured with Eclipse and MinGW. The grader will check your project settings to ensure they are set as required in this class (e.g., -pedantic, -Wall, -Wextra, -Wconversion).
- If required by the grader, demonstrate the execution of the program to the grader, using input of the grader's choosing.
- If required by the grader, explain the source code to the grader and answer questions from the grader about the code constructs used.

Grading

The program will be graded as follows:

10% - The program was submitted as a single file called main.cpp and other instructions followed.

10% - The program builds without error or warnings in Eclipse with MinGW configured as specified.

20% - When executed with correct input, the submitted program provides the required output.

20% - When executed with invalid input (e.g., extra characters, not providing an expected value), the submitted program prints a message and exits without crashing.

20% - The program code is well structured, readable, and implemented the required functionality in a direct and understandable manner

20% - The program is documented, describing the overall purpose of the program, describing how the functionality is implemented and describing what is happening in each logical section of the program. The identifiers are meaningful and consistent. The coding style is clear and consistent. If the grader requires a demonstration, answer the questions posed by the grader and explain the decisions made while writing the program.