TipCalc

Professor Caleb Fowler

Problem.

Create a program which will calculate various bill totals based on different tip percentages.

- Calculate bill total based on 10%, 15%, and 20% tip amounts.
- Create 3 functions which return a `double` . The first function returns a total based on a 10% tip. The second function, a 15% tip and the third function, a 20% tip.
- Call these three functions `calc_10, calc_15 and calc_20` , respectively.
- Create a void function, `print_tips` which prints a message and the total bill.

This is the basic homework problem for you to solve. This is the minimum set of specifications in order to obtain a 'C' or higher grade (This is before any late penalty).

Constraints.

- Use only 1 `using namespace std;` statement in this program.
- Keep the Input, Transformation, and Output sections of your code separate.

These are additional requirements on how you are to do the assignment. These are also required, but they apply to more then one specification task - and are included here to stand out.

Bonus Features.

- Add a novel programmer defined feature of your choosing (counts as 2 bonuses!).
- Add another tax rate (counts as 1 bonus).
- Wrap the input code in it's own function, too.

This is how you can go after higher grades. These are additional features you can add to the program. Adding 1 bonus is required to get you to a Highly Competent solution, 4 are necessary for a Sophisticated solution.

Sample Run and Test Data Set.

```
g++ -std=c++11 -g -Wall tipcalc.cpp -o tipcalc
./tipcalc
Enter the amount of the bill: 100
The bill + tip amount is $110
The bill + tip amount is $115
The bill + tip amount is $120
```

First sample run and test data set.

Due Date and Turn In.

This assignment is due on Saturday by 11:59 PM on the week it appears under Hw Due in your syllabus. Remember, online classes run from the Sunday shown on the Class Schedule to the following Saturday.

Turn homework in by uploading to the appropriate D2L Dropbox folder. You do not need to put your name in the **filename**; Homework1, 2 whatever will be just fine. D2L appends student information to the files when I download them, so I will see all this information

automatically. I will review your work using the rubric at the end of the assignment.

Do NOT SAVE your code as a .cpp file! Save it as a .txt file instead. Don't zip or otherwise compress your files. I will be able to read them once you get them on D2L. I have a script which converts the files to .cpp and automatically executes them. this script also runs other tests with them as well.

Using the Work of Others.

This is an individual assignment, you may use the Internet and your text to research it, but I expect you to work alone. Copying code from someone else and turning it in as your own is plagiarism. However, you **may** discuss code and the assignment. I have opened discussion groups in D2L to do this. I will monitor this, but not interfere. D2L will check your code against a database of other assignments. It tells me how similar your code is to someone else's. I consider isomorphic homework to be plagiarism. Do your own work.

Rubric for Evaluating this Assignment.

Grading Rubric					
	Sophisticated	Highly Competent	Competent	Not Yet Competent	Unacceptable
Solution Fit with Client Needs	As Highly Competent, but also successfully performs 3 bonus features (for a total of 4).	As Competent but also successfully performs 1 bonus feature also	Successfully accomplishes all specifications and constraints with the test data set.	Accomplishes some specifications and/or constraints with test data set. May have logic errors.	Does not meet any specifications or constraints. May not compile.
User Friendliness	~ Code has program greeting to introduce itself. ~ Program identified input expected from user.	~ Code has program greeting to introduce itself. ~ Program identified input expected from user.	~ Code has program greeting to introduce itself. ~ Program identified input expected from user.	Program requires omniscient users to divine expected input(s).	Input prompts are just a blinking cursor.
Comments and Documentation	~ Proper program header. ~ Function's properly commented. ~ Comments identify blocks of logically different code, and/or, modifications to formula's are noted. ~ Good use of whitespace.	~ Proper program header. ~ Function's properly commented.	~ Proper program header.	1 Line comment header and/or comments don't match code.	Missing program header, and/or, missing or incoherent comments.