CSCI 1470 **Fall 2024** **Mr. Gustavo Dietrich**

Homework 1: arithmetic expressions, formatting, rounding, etc. **Due Date:** 9/25/24

Suppose you have amounts in Mexican pesos and Euros that you would like to convert to U.S. Dollars to find out how much money they are worth in total. Write the algorithm (steps in pseudocode) and the corresponding program to convert Mexican Pesos and Euros to U.S. Dollars and display their equivalent and the total on the screen.

The program must prompt the user to enter the amount in Mexican Pesos and Euros (**double** precision real numbers) to be converted and then display the result in the format shown below. The amounts in U.S. Dollars (**double** precision real numbers) must be **rounded off** to the **cents** (second decimal digit).

Assume that the amounts in Mexican Pesos and Euros will be entered with no more than two decimal digits (cents).

For the conversion assume that:

```
1 U.S. Dollar = 17.47 Mexican Pesos
1 U.S. Dollar =  0.94 Euros
```

Solution requirements:

1) Declare named constants **DOL2MEX** (17.47) and **DOL2EURO** (0.94) for working with these values in your program. Declare variables **pesos** and **euros** to store the amounts in Mexican pesos and Euros entered by the user. Be sure of choosing the appropriate data types for them.

After rounding the values in **U.S. Dollars** to the cents you must add them to get the total and then **convert** each of them to **two separate whole numbers**, one corresponding to the whole part (bills) and the other corresponding to the decimal digits (coins). The numbers MUST be **converted** to whole numbers, not just displayed as whole numbers. Since you work with real numbers you should round them to get the correct amount in cents. Use the following identifiers for the variables that will hold the separated amounts:

- **wholedollarsp**: stores the whole dollars of the amount obtained from converting the pesos
- **centdollarsp**: stores the cents of the amount obtained from converting the pesos
- **wholedollarse** stores the whole dollars of the amount obtained from converting the euros
- **centdollarse**: stores the cents of the amount obtained from converting the euros
- **wholetotal**: stores the whole dollars of the total amount
- **centtotal**: stores the cents of the total amount

Note: you can declare other variables if necessary.

2) You **can** use ONLY the material learned and used in the first 6 lab assignments.

3) Your program must pass the 4 tests shown at the end of this handout.

4) Use the **round()** function to round your values.

5) Format the output to display **real** numbers in fixed format with 2 decimal digits. Pay attention to the format of the output (especially the alignment of columns).

The program must compile without errors or warnings.

Open **hw1.cpp** in your IDE and implement the above algorithm (already provided in the source code as comments).

Enter your algorithm (as comments) and implement it in C++.

Note:

- Do NOT remove or modify the statements that I use to test certain things in your program.
- Pay attention to the input and the output formats. Your solution must behave exactly like mine.
- Carefully analyze the sample run shown below and use it as a reference to ensure you do the right things.
- Use the values for pesos and euros specified in the four tests to check if you get the right results (compare with my solution). If you get an error message on the output, read the comment on the line specified in the message to find out what is wrong.

Sample run of the program

Review the examples discussed in class, the lab assignments done so far, and the textbook to get an idea of what you need to do. The **algorithm** must be written in **pseudocode** and should look like my lab handouts. **Include your algorithm in the source code as comments.**
Do not hesitate to use the corresponding topic in Discussions to post your questions/doubts about this assignment. I will reply as soon as I can.

IMPORTANT:

You must submit ONLY ONE solution per team.
Your program must be well commented, use meaningful identifiers, use named constants, and use indentation as shown in the textbook.
Your program must have the following comments at the top:

```
//*************************************************************************
// Team #            CSCI 1470              Fall 2024          Homework # 1
// First and Last Name
// First and Last Name
// Using your own words, write here a description of what the program does.
//
//*************************************************************************
```

Include the names of both teammates only if both participated in the solution, otherwise just enter your name. Include also your team number next to the #.
When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.

Paste the link to your final solution along with your source code in the textbox opened when you click on Create Submission before you click on Submit.

Grading criteria

You start with 100 points and then lose points as you don't do something that is required.

-30: Used material not corresponding to what was used in the first 6 lab assignments.
-10: Too few/no comment (no description of what the program does).
-10: Didn't use named constants.
-10: Data type is not correct.
-10: Mixed data types in expressions.
-10: Incorrect output format.
-20: Didn't round off.
-10: Incorrect rounding.
-10: Did not convert the amounts to whole values.
-10: Incorrect expression used to calculate the cents of U.S. Dollars
-10: Did not pass test. (**each**)
-30: Missing/incorrect algorithm.
-15: Poor quality algorithm.
-20: Incorrect/missing source code
 20: Incorrect/missing link to your solution
-50: Incomplete program.
-50: Program does not compile.
-100: No team contribution.
-100: **No team submission** or no submission at all.
-100: The code submitted is not your creation (you got it from a web site or another person)
-10: Late submission.

Important: more points may be lost for other reasons not specified here.

Sample runs of the program:

Note: I am showing the input and the output generated (you can see the values that were input from it).
Use the input values used in each of these sample runs to test your program to ensure it works correctly.

Test 1

Test 2

```
Mexican Peso and Euro to U.S. Dollar converter

Please enter the amount in Pesos: 188.25
Please enter the amount in Euros: 130.00


Mexican Peso and Euro to U.S. Dollar converter


    188.25 pesos:        10 US dollars with 78 cents

    130.00 euros:       138 US dollars with 30 cents

           Total:       149 US dollars with  8 cents


Testing your solution
```

Test 3

```
Mexican Peso and Euro to U.S. Dollar converter

Please enter the amount in Pesos: 200.00
Please enter the amount in Euros: 200.00


Mexican Peso and Euro to U.S. Dollar converter


    200.00 pesos:        11 US dollars with 45 cents

    200.00 euros:       212 US dollars with 77 cents

           Total:       224 US dollars with 22 cents


Testing your solution
```

Test 4

```
Mexican Peso and Euro to U.S. Dollar converter

Please enter the amount in Pesos: 856.70
Please enter the amount in Euros: 93.07


Mexican Peso and Euro to U.S. Dollar converter


    856.70 pesos:        49 US dollars with  4 cents

     93.07 euros:        99 US dollars with  1 cents

           Total:       148 US dollars with  5 cents


Testing your solution
```