

Assignment on Functions:

For each of the items, you are to submit the source code (printout and softcopy). For the printout, attach each source code with the accomplished test script (for items A-C) and answer to the questions (for item D). The attachments can be re-typed using the given format or they can be handwritten on the printout of these pages. For the softcopy, bring a copy of each of the source codes in USB. Ideally, you also have another back-up copy (maybe sent to YOUR OWN email account).

A. Currency Converter v1

Michelle wants to know how much her money (in Philippine peso) can get her these days if she lived in another country. Currently, the US Dollar, the Euro, the Japanese Yen, the British Pound and the Swiss Franc are the five major world currencies in use. Your task is to create a program (currency.c) that converts Philippine peso to those 5 major world currencies. Follow the screen format below. Remember, the peso amount WILL COME FROM THE USER. DO NOT HARDCODE THIS VALUE. Assume that the input is a non-negative value.

Note: Use this website for exchange rates: <http://www.exchange-rates.org/MajorRates.aspx>

Sample Screen Output:

```
*****CURRENCY CONVERT*****
Please input an amount in Peso:
1.00
Converting... Done!
In US Dollars: 0.02
In Euro: 0.01
In Japanese Yen: 1.93
In British Pound: 0.01
In Swiss Franc: 0.02
Thank You!
*****
```

Print the test script with the following categories (based on description) of test cases per function

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
main()	1	Input is base value used in the computation	Peso: 1.00	USD: Euro: JPY: GBP: CHF:	USD: Euro: JPY: GBP: CHF:	
	2	Input is a whole number	Peso: 547	USD: Euro: JPY: GBP: CHF:	USD: Euro: JPY: GBP: CHF:	
	3	Input is a real number	Peso: 6789.65	USD: Euro: JPY: GBP: CHF:	USD: Euro: JPY: GBP: CHF:	
	4	Input is less than 1	Peso: 0.30	USD:	USD:	

		peso		Euro: JPY: GBP: CHF:	Euro: JPY: GBP: CHF:	
--	--	------	--	-------------------------------	-------------------------------	--

Bonus achievements:

Symbol Mastery (1 point)

Use currency symbols in your final output, just before displaying the value (i.e. for US Dollars display \$, for Yen display ¥, for Pound display £ and for Franc display **f**. Unfortunately, the Euro sign cannot be displayed, so just use €).

Printf Mastery (1 point)

Use no more than two printf statements in your final answer.

B. Currency Converter v2

Create another program (currency2.c) that will perform similarly to version 1. However, you are to write the following functions:

- getInput() – gets the peso input from the user
- toUSD() – generates the USD equivalent of the peso value. Note, this function should not ask for user input (i.e., no scanf statement in this function).
- toEuro() – generates the Euro equivalent of the peso value. Note, this function should not ask for user input.
- toJPY() – generates the Japanese Yen equivalent of the peso value. Note, this function should not ask for user input.
- toGBP() – generates the British pound equivalent of the peso value. Note, this function should not ask for user input.
- toCHF() – generates the Swiss Franc equivalent of the peso value. Note, this function should not ask for user input.
- displayResult() – displays the resulting converted values.
- main() – calls the functions to get input, compute for the conversion, and to display the result.

Complete the test script below with the following categories of test cases per function:

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
toUSD()	1	Input is base value used in the computation	Peso: 1.00			
	2	Input is a whole number	Peso:			
	3	Input is a real number	Peso:			
	4	Input is less than 1 peso	Peso:			
toEuro()	1	Input is base value used in the computation	Peso: 1.00			
	2	Input is a whole number	Peso:			
	3	Input is a real number	Peso:			
	4	Input is less than 1 peso	Peso:			
toJPY()	1	Input is base value	Peso: 1.00			

		used in the computation				
	2	Input is a whole number	Peso:			
	3	Input is a real number	Peso:			
	4	Input is less than 1 peso	Peso:			
toGBP()	1	Input is base value used in the computation	Peso: 1.00			
	2	Input is a whole number	Peso:			
	3	Input is a real number	Peso:			
	4	Input is less than 1 peso	Peso:			
toCHF()	1	Input is base value used in the computation	Peso: 1.00			
	2	Input is a whole number	Peso:			
	3	Input is a real number	Peso:			
	4	Input is less than 1 peso	Peso:			

C. Currency Converter v3

Create another program (currency3.c) that will perform similarly to version 1. However, you are to write the following functions:

- getInput() – gets the peso input from the user
- convert() – generates the equivalent of the peso value, given the conversion rate. Note, this function should not ask for user input (i.e., no scanf statement in this function). Use the following function prototype, where fPeso is the peso value and fRate is the conversion rate:
float convert(float fPeso, float fRate);
- displayResult() – calls the function to convert the peso value to different currencies and display these on screen.
- main() – calls the functions to get input and to display the result.

Complete the test script by indicating **at least** four categories of test cases for function convert():

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
convert()	1		fPeso:			
			fRate:			
	2		fPeso:			
			fRate:			
	3		fPeso:			

			fRate:			
	4		fPeso: fRate:			

D. Currency Converter v4

Create another program (currency4.c) that will perform similarly to version 1. However, you are to write the following functions:

- a.) getInput() – gets the peso input from the user
 - b.) convert() – generates the equivalent of the peso value to all the other currencies. Note, this function should not have any call to scanf() or printf(). Use the following function prototype, where fPeso is the peso value and the rest are addresses referring to the converted currency values.

```
void convert(float fPeso, float *pUSD, float *pEuro, float *pJPY, float *pGBP, float *pCHF);
```
 - c.) displayResult() – displays the resulting conversion of the peso value to different currencies on screen.
 - d.) main() – calls the functions to get input, convert, and to display the result.
1. After writing four different versions of the program to convert the peso amount to five different currencies, this shows that there are different ways to solve the same problem. In your opinion, which of the four versions is the best solution and why?
 2. Why do you write / create user-defined functions (that is, why don't you just put all your code in the main program)?
 3. What is the purpose of parameters (e.g., fPeso)?
 4. What is the purpose of using addresses as parameters (e.g., pUSD)?

E. Just for Fun Ascii Art

Now that you know how (and why there is a need) to write functions, create a program that will show an Ascii art of your nickname (or just use your initials). Let's limit this exercise to creating showing at least 3 and at most 5 Ascii art, each of which is a 5 x 5 Ascii art of a letter. The display is vertical, per character. For example, if I were to write a program, the output would show the following (where each unique letter should be implemented as a function, i.e., 1 function to display N, another to display A, etc.):

```
N    N
NN   N
N N  N
N    NN
N    N
```

```
   AAA
A    A
AAAAA
A    A
A    A
```

```
TTTTT
  T
  T
  T
  T
```

```
   SSS
S
SSSS
      S
SSSS
```