

## Money Math

Professor Caleb Fowler

January 30, 2016

### Problem.

Enter two dollar amounts (A and B) on the command line and then complete the following tasks:

- Compute  $A + B$ ,  $A - B$ ,  $A \times B$ ,  $A / B$
- Compute simple interest for  $A + B$ . You will need to prompt for an interest rate and the number of years. This will tell you how much money you will earn if you invest  $A + B$  for term years at interest rate  $i$ . Display just the amount of interest you made.
- Compute federal tax of  $A \times B$ . Use a flat rate of 32%.
- Compute CA State tax of  $A \times B$ , too. Use a flat rate of 7%.
- Convert the totals from item 1 into euros and display.

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 CONSTANTS for the conversion rates, instead of literals.
- Enter whole numbers (integers) for dollars (ie \$100.00 is entered as 10000). Print in dollars and cents (ie 100.00). Don't worry if the math causes too many decimals.
- Keep the Input, Transformation, and Output sections of your code separate.
- In Item 1, under problem, use 1 line to declare four results variables and another line of code to initialize them to 0.
- Create a void function called ProgramGreeting(), which displays a welcome message.

These are additional requirements on how you are to do the assignment. These are also required, but they apply to more than 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!).
- Convert to a currency other than Euros. Note: This only counts as 1 bonus because this isn't novel.
- Add taxes for other states (also counts as 1 bonus).
- Print the actual numbers you entered in, say  $100 + 50 =$ , rather than  $A + B =$

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.*

```
Hackintosh:Money Math clf$ python3.5 moneymath.py
Perform a number of exercises using money.
I'm going to ask you for dollars, enter them without the decimal point(ie 1000 for
ten dollars).
Enter the first dollar amount: 100
Enter the second dollar amount: 100
Enter an interest rate: 12
Enter the number of years to calculate the interest: 2
```

## Displaying Math Operations

```
$ 1.0 + $ 1.0 = $ 2.0
$ 1.0 - $ 1.0 = $ 0.0
$ 1.0 x $ 1.0 = $ 1.0
$ 1.0 / $ 1.0 = $ 1.0
```

## Displaying Simple Interest on A + B

An investment with the parameters specified above will generate, \$ 0.48

## Print taxes on A x B

```
Federal taxes: 0.32
CA taxes: 0.07
```

## Dollar to Euro conversion

```
Conversion of A + B: € 1.84
Conversion of A - B: € 0.0
Conversion of A x B: € 0.92
Conversion of A / B: € 0.92
```

First sample run and test data set.

```
Hackintosh:Money Math clf$ █
```

```
Hackintosh:Money Math clf$ python3.5 moneymath.py
Perform a number of exercises using money.
I'm going to ask you for dollars, enter them without the decimal point(ie 1000 for
ten dollars).
Enter the first dollar amount: 100000
Enter the second dollar amount: 15000
Enter an interest rate: 10
Enter the number of years to calculate the interest: 10
```

## Displaying Math Operations

```
$ 1000.0 + $ 150.0 = $ 1150.0
$ 1000.0 - $ 150.0 = $ 850.0
$ 1000.0 x $ 150.0 = $ 150000.0
$ 1000.0 / $ 150.0 = $ 6.666666666666667
```

## Displaying Simple Interest on A + B

An investment with the parameters specified above will generate, \$ 1150.0

## Print taxes on A x B

```
Federal taxes: 48000.0
CA taxes: 10500.0000000000002
```

## Dollar to Euro conversion

```
Conversion of A + B: € 1058.0
Conversion of A - B: € 782.0
Conversion of A x B: € 138000.0
Conversion of A / B: € 6.133333333333334
```

Second sample run and test data set.

```
Hackintosh:Money Math clf$ █
```

```

Perform a number of exercises using money.
I'm going to ask you for dollars, enter them without the decimal point(ie 1000 for
ten dollars).
Enter the first dollar amount: 1
Enter the second dollar amount: 1
Enter an interest rate: 1
Enter the number of years to calculate the interest: 1

Displaying Math Operations
$ 0.01 + $ 0.01 = $ 0.02
$ 0.01 - $ 0.01 = $ 0.0
$ 0.01 x $ 0.01 = $ 0.0001
$ 0.01 / $ 0.01 = $ 1.0

Displaying Simple Interest on A + B
An investment with the parameters specified above will generate, $ 0.00019999999999
99988

Print taxes on A x B
Federal taxes: 3.2000000000000005e-05
CA taxes: 7.000000000000001e-06

Dollar to Euro conversion
Conversion of A + B: € 0.0184
Conversion of A - B: € 0.0
Conversion of A x B: € 9.200000000000001e-05
Conversion of A / B: € 0.92
Hackintosh:Money Math.c1f$ █

```

Third 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.

UNLOADING, 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.

TURN IT IN by uploading to the D2L Dropbox folder for the appropriate assignment. 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.

### *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.

### *Discussion.*

Simple interest is used for calculating how much interest you earn if you make a single contribution at the start of the term. Contrast this with Compound Interest, where you make regular, periodic contributions throughout the life of the investment. The simple interest formula is

$$A = P(1 + rt)$$

Where, A is the total of principal and interest, P is the principal or initial investment, r is the interest rate (as .15), and t is the term (often years).

CONVERTING CURRENCIES USES the following formula:

$$\text{Euros} = \text{Dollars}(\text{CurrentExchangeRateofEuro}) / \text{CurrentExchangeRateofDollar}$$

Hint: The key to understanding this formula is to realize in the VAST majority of situations, you will be converting a foreign currency into 1 US Dollar.

*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.