CS 121 – Lab #9 – Function practice

The purpose of today's lab is to create simple functions, and write function calls for them. You will see how parameters are used by a function, and how return values are used by the main program. On your USB drive or account, create a new directory called lab09. You will create 2 programs today. Open Anaconda Spyder as usual.

Part 1: Financial functions

This program will be called finance.py. Let's write functions that perform three useful financial computations. Each function will implement an important formula from the world of personal finance: compound interest, future value of an annuity, and a loan payment. Here are the respective formulas.

The formula for compound interest is:

$$FV = PV\ (1 + i)^n$$

where PV is how much money you start with, n is the number of years, and i is the interest rate expressed as a decimal.

The fomula for the future value of an annuity is:

$$FV = amount\ \frac{(1 + i)^n - 1}{i}$$

where amount is how much money is invested per year, n is the number of years, and i is the interest rate expressed as a decimal.

And the formula for calculating a monthly loan payment is:

$$PMT = PV\ \frac{i}{1 - (1 + i)^{-n}}$$

where PV is the amount of money being borrowed, n is the number of monthly payments, and i is the *monthly* interest rate expressed as a decimal.

Inside finance.py, create three functions. Suitable names for these functions are: compound, annuity, and loan. All three functions will take three parameters: an amount of money, a number of years, and an interest rate. You should assume that when the interest rate is passed as a parameter, it will be expressed as a percentage. For example, 5 per cent will be represented as 5. Your functions will need to convert this to a decimal.

Underneath your three functions, you can write the *main program*. It should call each function and print the result in a complete sentence such as "The future value is $ 92.50" or "The loan payment is $ 18.68". The function calls should determine the following.

- If I have $1,000 earning 7% interest compounded for 10 years, how much money will the account grow to?
- The IRS allows people to invest $19,000 per year into their 401(k) plan. If I invest $19,000 each year earning 3%, how much money will I have after 30 years?

- To finance the purchase of a house, I need to borrow $100,000 for 15 years at 5%. What will my monthly loan payment be?

Part 2: Calling a function from inside a loop

Begin by creating a second source file.

1. Write a function called fun1 that takes one real-number parameter. Let’s say the parameter is called x. The function should return the value $3x^2 – 2x + 1$.

2. In your main program, let’s evaluate fun1 for all integer values from 1 to 10 inclusive. In pseudocode, our plan is to do this:
   ```
   for i = 1 to 10,
      print both i and fun1(i)
   ```

3. Let’s add another function to the top of the program. Write a function called fun2. It also takes a single real-number parameter, which we can call x. The function should return the value x^{-x}. Remember that you can use ** to perform exponentiation in Python.

4. The fun2 function is studied in calculus. For example, we may want to know where this function reaches its maximum value. It is somewhere between x = 0 and x = 1. Let’s print a data table for fun2 that evaluates for each hundredth from 0.01 to 1.00. In other words, we should call fun2 on 0.01, 0.02, 0.03, etc. all the way up to 1.00. The pseudocode looks like this:
   ```
   for i = 0.01 to 1.00, incrementing by .01,
      print both i and fun2(i)
   ```

   If you use a for-loop, remember that the range() function only accepts integers.

5. Modify the code for the loop that you just wrote. We want to keep track of the (nearly) exact location of the maximum value of fun2. It would go something like this:

   ```
   max_i = 0.01
   max_value = fun2(0.01)
   for i = 0.01 to 1.00, incrementing by .01,
      print both i and fun2(i)
      if fun2(i) > max_value,
         update both max_i and max_value
   ```

   At the end of the program, tell the user at what value of i fun2 achieved its maximum, and what that maximum value was.

When you are ready, please show your results to the instructor or lab aide. You’re done!