Lab Exercise 6

Focus

1. Library functions
2. User-defined value returning functions
3. Modules

This lab maps to learning objectives 1 through 4 in Competency Module Six Write a Working Program that Uses Value Returning Functions and Modules that Store Functions.

Part A: Building on an Existing Solution

For this portion of the lab, you will reuse the program you wrote in Lab2 part B (*Lab2.py*). Redesign the solution so that each one of the calculations is done in a value-returning function. (LO 1) as follows

1. Create a value-returning function called *milesToKm* that receives miles as a parameter and returns the calculated kilometers.
2. Create a function called *degreesFToC* that receives the Fahrenheit temperature as a parameter and returns the calculated Celcius temperature.
3. Create a function called *gallonsToLiters* that receives the gallons as a parameter and returns the calculated liters.
4. Create a function called *poundsToKg* that receives the pounds as a parameter and returns the calculated kilograms
5. Create a function called *inchesToCm* that receives the inches as a parameter and returns the calculated centimeters.

Put all these conversion functions in a module called firstname_lastname_Lab6a_module.py.

Have the main program import your module. Your main program should do all the input and output, and call the conversion functions in the module to do the calculations. Put your main program in a function called *main()* and be sure to call it. (LO 4)

Save the program as firstname_lastname_Lab6a.py where you will replace firstname and lastname with your actual first and last name.

Part B: Write Something New

Write a complete and syntactically correct Python program to solve the following problem:

Write a simple guessing game. The computer will pick an integer and ask the user to guess it. The user will keep guessing numbers until they guess the correct number. Hint: What type of loop keeps reading input until a special value is input?

Use the following specifications:

1. Generate a random number between 1 and 100. (LO 1, 2, 3)
2. Ask the player to guess the number. You will keep track of the number of guesses.
3. If the player's guess is larger than the generated number, display the message "Too high!"
4. If the player's guess is samller than the actual number, display "Too low!'

Once the player guesses the number, display a congratulatory message like "You guessed the number in x tries!" where x is the actual number of tries it took the player to guess the number. You can write any message as long as you include the number of tries in the message.

Once the player has guessed the number, ask them if they wish to play again. If they do then generate another random number and start the game over again. (LO 3)

All input to the program will be interactive from the keyboard

Use the IDLE programming environment if you are using Python with IDLE. Please save your file as firstname_lastname_Lab6b.py where you will replace firstname and lastname with your actual first name and last name. Remember to use the extension .py.

Run and test your program for all conditions. Once you are sure it works you will turn in the items listed in the next section.

Turn In

All labs will be graded in Blackboard. Once you are done with the lab turn it in to the Lab 6 link. If you have any questions or problems with submission, contact the instructor.

For this lab you will turn into Blackboard:

1. The Python code files you saved in part A.
2. The Python code file you saved in part B