Lab 4
Functions

Z. Hutchinson
zachary.s.hutchinson@maine.edu

October 4, 2022

Goals

The goal of this lab is to practice using Python 3. Specifically, you will practice:

- Functions

Instructions

All work is due at the **end of your lab** and must be submitted to Brightspace in the proper place. Unless otherwise instructed, submissions must be python files (e.g. files that end with *.py*). Any other format, even if it is plain text, will **not** be graded. Messy or otherwise unreadable code will lose points. Lab submissions can be all in the same file, but please label with comments to which task code belongs. IMPORTANT: Any code that is commented out will not be graded. **RUN YOUR CODE TO MAKE SURE IT WORKS!!!**

Task 1 - Functions

Write and test the following functions. In this section you will practice functions that have and do not have parameters and functions that do and do not return information.

A - No Parameters, No Return

Write a function *Hello* which simply prints 'Hello world'.

B - One Parameter, No Return

Write a function *HelloTo* which has one parameter, *name*. The function should print: 'Hello, [name]!', where [name] is the value passed in via the parameter.

C - Four Parameters, One Return

Write a function called *Distance* which calculates the distance between two Cartesian coordinates. The function has four parameters: *x1*, *y1*, *x2*, and *y2*. The function should return the distance between points $(x1, y1)$ and $(x2, y2)$. The formula to calculate distance is:

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \tag{1}$$

Task 2 - Functions Using Functions

The functions in this section work together to solve a larger task.

A - Get Direction

The function *GetDirection* should ask the user for one of four possible inputs. The function has no parameters. It should ask the user to enter either, *Q*, *N*, *E*, *S* or *W*. If the user does not enter one of these options, the user is asked until they do. Once they enter one of the valid possibilities, the function returns the user's input.

B - Parse Input

The function *ParseInput* has one parameter, *direction*. The function expects *direction* to contain a single character. Valid values are *N*, *E*, *S* or *W*. Depending on the value in *direction* the function should print: 'The player moves [direction].' For example: if the value in direction is 'N', it should print *The player moves north.*

C - Game Loop

The function *GameLoop* has no parameters and does not return anything. The function contains a sentinel loop. This function uses the previous two functions. Within the loop, it asks for user input using *GetDirection*. If the value returned from *GetDirection* is 'Q' the game loop quits. If it is something else, the value is passed into *ParseInput*. There is no way to win the game.

D - Extra Challenge

Rework *GameLoop* so that it is possible to win the game. To win the game, the player must make four specific moves. For example: S, E, W, S. When the player makes a correct move, they are told (in addition to the normal ParseInput message): 'You are on the right track'. If they make a wrong move, they are told: 'Bad move. Game reset.' The loop will need to keep track of the winning moves, whether the player is on the right track and how far along the track the player is. If the player makes several correct moves and then an incorrect move, their progress is reset. The game continues until they win or quit. This game is vaguely similar to Simon Says.

Task 3 - More Fun With Functions

Convert the following to Python code. Make sure you test your functions by calling them with different values.

A

Write a function called *AbsVal* that takes one parameter, *num*. This function should return the absolute value of the number in *num*. Do not use the *abs()* built-in function. Make sure you test it for positive and negative numbers as well as zero.

B

Write a function called *MinsToDaysHrsMins*. The function takes one parameter, *minutes*, which contains a specific number of minutes. The function should convert and print the number of minutes to days, hours and minutes. For example, if I pass 1927 minutes into the function, it will print: 1 day, 8 hours and 7 minutes.

C

Write a function called *ConvertTemp*. This function takes two parameters: *temp* and *scale*. If *scale* contains the string 'C', the temperature in *temp* should be converted to Celsius (it is assumed to be Fahrenheit). If *scale* contains an 'F', you convert it to Fahrenheit. In both cases, the function should return the converted temperature.

$$C = \frac{F - 32}{1.8} \tag{2}$$

$$F = 1.8C + 32 \tag{3}$$

Task 4 - Reading Code

Determine what each section prints simply by reading the code. Once you have an answer, copy the code into your interpreter and test your answer.

A

```
def FunctionA(numA):
    return numA // 2

def FunctionB(numB):
    if numB % 2 == 1:
        return FunctionA(numB - 1)
    else:
        return numB

def FunctionC(numC):
    if numC > 10:
        return FunctionB(numC // 3)
    else:
        return FunctionA(numC + 10)

def main():
    num = 27
    print(FunctionC(num))
main()
```