**CSC 148 Lab 4: Functions and Conditionals**

**September 23h, 2021**

**Total: 15 + 5(bonus)**

**Due: Before class, September 30th, 2021.**

**Covered topics: Variables, Assignments, Functions (with return), Conditionals and Simple looping**

Put your solution of the exercises in **separate .py files**, and name the files using the following convention **QuestionNumber\_YourName.py**.

Put all the .py files in a folder, zip it, and upload the zipped folder to Canvas.

You will exercise function definition, calling return and conditionals (if, else, elif and nested if statement) in this lab.

For more tutorials on if, elif, else, please refer to your textbook.

<http://www.greenteapress.com/thinkpython/html/thinkpython006.html>

and here:

<http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/ifstatements.html>

1. (5pts) You are writing a function to help provide a price estimate for melons. Our salespeople use the following rules to determine the price of melons:

* Normally, melons cost $5
* The base cost of Casabas and Ogens is $1/each more, though, as they're harder to grow.
* Imported melons cost 1.5 times as much as they otherwise would, because of shipping cost.
* Square melons cost 2 times as much as they otherwise would, because of the much more intense effort to grow them.
* In addition, melons are only available in summer months June, July, and August.

Right now, we've just had our salespeople look at the data file and figure out how much a melon will cost by applying all of these rules to the melon.

Write a function called *get\_price(type, imported, quantity, month)* and compute the price of a melon based on its type, quantity, month of purchase, and whether it is imported. You can use *input* to ask the user about what type of melon they want to buy and how many and output the price. If the time of the year doesn’t fit in the melon growing season, tell the customer kindly that they are not available.

**Hint: you might need to nest your if/else statement because an expensive type of melon can also be imported.**

2. (5pts) Fermat’s Last Theorem says that there are no positive ***integers***a, b, and c such that

|  |
| --- |
| an + bn = cn |

for any values of *n* greater than 2.

1. Write a function named *check\_fermat* that takes four parameters—*a, b, c* and *n*—and that checks to see if Fermat’s theorem holds. If n is greater than 2 and it turns out to be true that

|  |
| --- |
| an + bn = cn |

the program should print, “Holy smokes, Fermat was wrong!”. Otherwise the program should print, “No, that doesn’t work.”

1. Write a function that prompts the user to input values for a, b, c and n, converts them to integers, and uses *check\_fermat* to check whether they violate Fermat’s.

3. (5pts) Finish writing a module called height.py. In the module, define three functions based on the specifications (docstring):

“””This module shows off three functions for converting height between

height\_in\_inches and feet\_and\_inches. It also shows how to use variables to represent constants: values given a name in order to remember them

better."""

#constant

INCHES\_PER\_FOOT = 12

def to\_inches(feet, inches):

"""Parameter feet: foot component of height

Parament inches: inch component of height

Returns: feet-and-inches height converted to only-inches height

"""

def get\_feet(height\_in\_inches):

"""Parameter height\_in\_inches: one's height in inches

Returns: foot component of one's height"""

def get\_inches\_left(height\_in\_inches):

"""Parameter height\_in\_inches: one's height in inches

Returns: inches component of one's height"""

A person’s height is 5 feet and 7 inches, but it is also 67 inches.

Import the above module and calculate out the following:

1. What is the person’s height in total inches for a height of 5 feet and 7 inches?
2. What is the feet component of 67 inches after converting it to x feet and y inches?
3. What is the inches component for 67 inches after converting it to x feet and y inches?

height= input(“what is your height in inches?)”  
# write code to output the above 3 questions

Hint: You can write another .py to import the module and do the calculations.

4. (5pts bonus) Turtle graphics. Please do not attempt this if you don’t finish the previous 3 questions.

Import turtle

>>> import turtle

>>> bob = turtle.Turtle()

To move the turtle forward

>>> bob.fd(100)

To make a left turn

>>> bob.lt(100)

You can read more about turtle graphics in Chapter 4 in your textbook.

<https://greenteapress.com/thinkpython2/thinkpython2.pdf>

Write a function called square that takes a parameter named **t**, which is a turtle. It should use the turtle to draw a square. Add another parameter, named **length**, of the square. Test your program to draw squares of different sizes.