Critical Thinking assignment Module 5-

Part 1-

Solution-

“The range() function allows counting in for loops as well. range() generates a sequence of numbers, starting at zero and ending before a value given inside the parentheses. Ex: for i in range(3) sets i to 0 during the first iteration of the for loop, i to 1 during the second iteration, and finally i to 2 on the third iteration. The value within the parentheses is not included in the generated sequence.

The range() function can take up to three arguments to indicate the starting value of the sequence, the ending value of the sequence minus 1, and the interval between numbers.”

<https://learn.zybooks.com/zybook/CSUGLOBALCSC500MASTER1/chapter/4/section/6>

For this problem, I would think of using 2 for loops first one iterates for number of years and the second, internal for 12 months for each year, I would ensure that I am getting correct output so I added a print statement to check if my for loop is functioning as desired, I did a bunch of trials got a lot of errors which I fixed one by one.

Here is the pseudo code –

BEGIN

Initialize month to 0

Initialize avg\_rain to 0

Initialize Total\_rain to 0

Prompt the user for the number of years and store it in number\_of\_years

FOR i from 0 to number\_of\_years - 1

Display "This is the year # " with the year number (i+1)

FOR j from 0 to 11

Increment month by 1

Prompt the user for the average rainfall in inches for the current month and store it in rainfall

Add rainfall to Total\_rain

Display "Total rain till now = " with the current Total\_rain

END FOR

END FOR

Calculate avg\_rain as Total\_rain divided by (number\_of\_years multiplied by 12)

Display The total number of Months = ( (number\_of\_years multiplied by 12)

Display "The total inches of rainfall “

Display "The average rain per month over the period entered in the beginning.”

END

Source code-

#program for average rainfall using nested loop

month = 0

avg\_rain = 0

Total\_rain = 0

number\_of\_years = int(input('Enter the number of years :'))#taking input for no. of years

for i in range(number\_of\_years):

print('This is the year # ',i+1) #first FOR loop, external one

for j in range(12):

month = month + 1 #second for loop

#print(' This is month:',month)

rainfall = float(input('Enter average rainfall in inches for this month'))

Total\_rain = rainfall + Total\_rain

print('Total rain till now = ', Total\_rain)

#Total\_rain= rainfall + Total\_rain

avg\_rain = Total\_rain / (number\_of\_years \* 12) #calculates average rain in entire time period

print('The total number of Months = \n', number\_of\_years \*12 )

print('The total inches of rainfall is ',Total\_rain, end='')

print(' inches')

print(f'The average rain per month over the period of {number\_of\_years \* 12} months is: {avg\_rain} inches')

A screenshot of a computer program

Description automatically generated

Execution of the code –

All the errors I got-

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Final correct executions—

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Part 2

BookStore-

“Sometimes the programmer has multiple if statements in sequence, which looks similar to a multi-branch if-else statement but has a very different meaning. Each if statement is independent, and thus more than one branch can execute, in contrast to the multi-branch if-else arrangement”

<https://learn.zybooks.com/zybook/CSUGLOBALCSC500MASTER1/chapter/5/section/3>

I would be using multiple distinct if statements. I can also think of using multiple IF and ElseIF, statements as there are multiple conditions to be satisfied in order to get the final answer/solution.

Since we have multiple conditions that are not mutually exclusive and we want to check each condition separately, using distinct if statements allow us to do so. This means that multiple blocks of code can run if their respective conditions are true.

Using separate if statements can also make the code more readable.

<https://learnpython.com/blog/multiple-conditions/>

<https://www.askpython.com/python/examples/multi-conditional-if-statement>

Pseudocode-

BEGIN

Initialize the variable to hold the value of points earned, initialize it with 0

Take another variable books\_purchased and prompt user to input the number of books purchased this month assign it to the books\_ purchased variable.

Check for the following conditions and show the output using print statement.

IF a customer purchases 0 books, Display they earn 0 points.

IF a customer purchases 2 books, Display they earn 5 points.

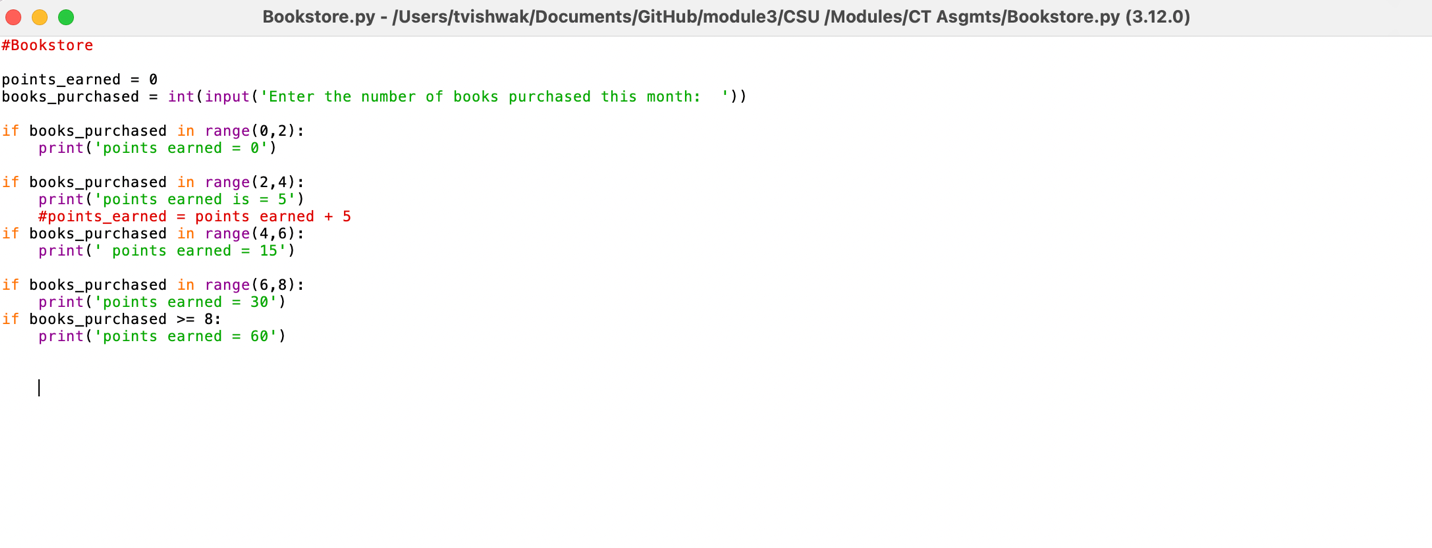
IF a customer purchases 4 books, Display they earn 15 points.

IF a customer purchases 6 books, Display they earn 30 points.

IF a customer purchases 8 or more books, Display they earn 60 points.

END

Source Code-



Code Execution-

A screenshot of a computer

Description automatically generated

Link to Git repo for both the programs-

<https://github.com/triptiV1/module3/commit/fbabb9651fc2616ce9ebe954fd00000bc7eaada3#diff-9e24fdf7cf3f240fadac5940284456b9b18226cd14224483eb4088ec788d5a36>

<https://github.com/triptiV1/module3/commit/fbabb9651fc2616ce9ebe954fd00000bc7eaada3#diff-d19ea1f6548c5ac5a1c53b3bfa8b9c6bd00dc3875e4782e778b7235e1b5046f5>