Assignment 5

Name: Lukas Pfalz

ITS 1110

Section: CSC-1110-71762

# **Assignment 5**

***COMPLETE THE HONOR CODE BELOW***

**HONOR CODE:**

I pledge that this program represents my own program code, I have received help from no one and I have given help to no one.

**OR**

I received help from **NAME OR NO ONE** in designing and debugging my program.

I have given help to **NAME OR NO ONE** in designing and debugging my program.

I pledge that this program represents my own program code, I have received help from no one and I have given help to no one.

This assignment is required.

The grading form shows point values for this assignment. Please review it now.

Show screen shots of the python code with comments and your input/output window.

You should use several well-planned sets of data to check out your program. Testing your programs with just the data that is asked for in the assignment does not necessarily mean that the programs will work for all cases.

Please include the following comments in each of your Python programs:

Your Name

Section

Date

Description

Assignment Number

A python template (python\_template.py) has been provided for you to use.

Name this document XXX\_Assignment 5 where XXX are your initials. Include a python file named P01.py, P02.py, etc. for each problem.

## For this part you will be writing three functions:

|  |  |
| --- | --- |
| # function: | horizontal\_line |
| # input: | a width value (integer) |
| # processing: | prints a single horizontal line of the desired size |
| # output: | does not return anything |

|  |  |
| --- | --- |
| # function: | vertical\_line |
| # input: | a shift value and a height value (both integers) |
| # processing: | generates a single vertical line of the desired height. The line is offset from the left side of the screen using the shift value |
| # output: | does not return anything |

|  |  |
| --- | --- |
| # function: | two\_vertical\_lines |
| # input: | a width value and a height value (both integers) |
| # processing: | generates two vertical lines. the first line is along the left side of the screen. the second line is offset using the "width" value supplied |
| # output: | does not return anything |

Here is the sample program that you **will** use to test your functions. The expected output is printed below the sample code:

print ("Horizontal line, width = 5:")

horizontal\_line(5)

print ()

print ("Horizontal line, width = 10:")

horizontal\_line(10)

print ()

print ("Horizontal line, width = 15:")

horizontal\_line(15)

print ()

print ("Vertical Line, shift=0; height=3:")

vertical\_line(0, 3)

print ()

print ("Vertical Line, shift=3; height=3:")

vertical\_line(3, 3)

print ()

print ("Vertical Line, shift=6; height=5:")

vertical\_line(6, 5)

print ()

print ("Two Vertical Lines, height=3; width=3:")

two\_vertical\_lines(3, 3)

print ()

print ("Two Vertical Lines, height=4; width=5:")

two\_vertical\_lines(4, 5)

print ()

print ("Two Vertical Lines, height=5; width=2:")

two\_vertical\_lines(5, 2)

**Expected Output:**

Horizontal line, width = 5:

\*\*\*\*\*

Horizontal line, width = 10:

\*\*\*\*\*\*\*\*\*\*

Horizontal line, width = 15:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Vertical Line, shift=0; height=3:

\*

\*

\*

Vertical Line, shift=3; height=3:

\*

\*

\*

Vertical Line, shift=6; height=5:

\*

\*

\*

\*

\*

Two Vertical Lines, height=3; width=3:

\* \*

\* \*

\* \*

Two Vertical Lines, height=4; width=5:

\* \*

\* \*

\* \*

\* \*

Two Vertical Lines, height=5; width=2:

\*\*

\*\*

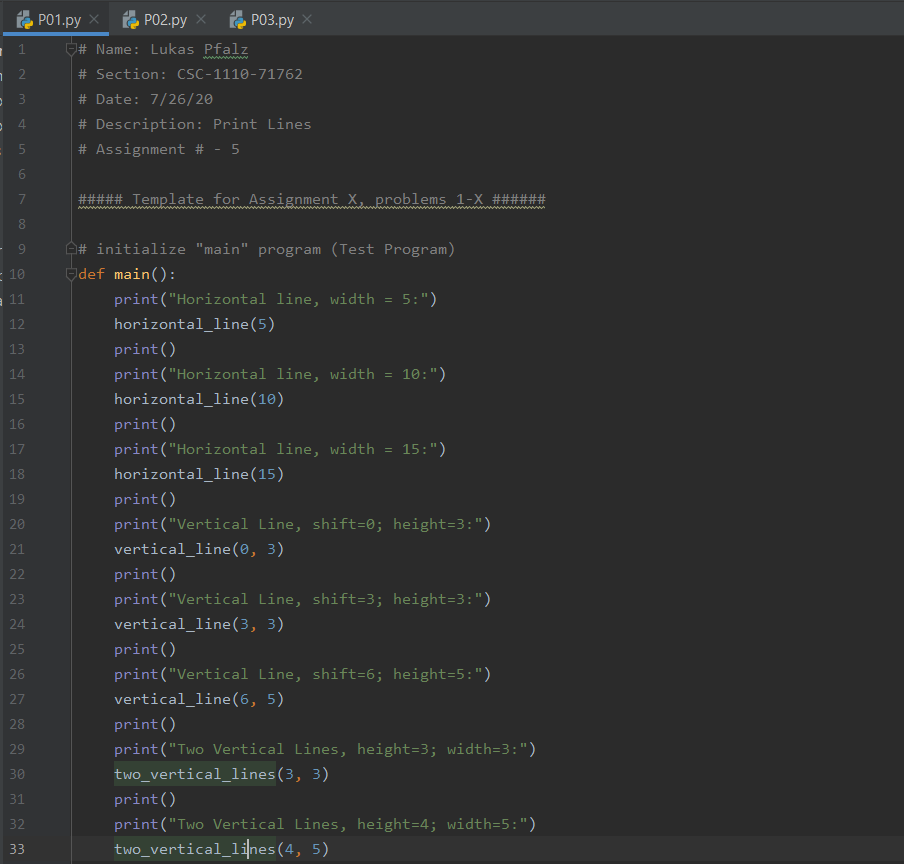
\*\*

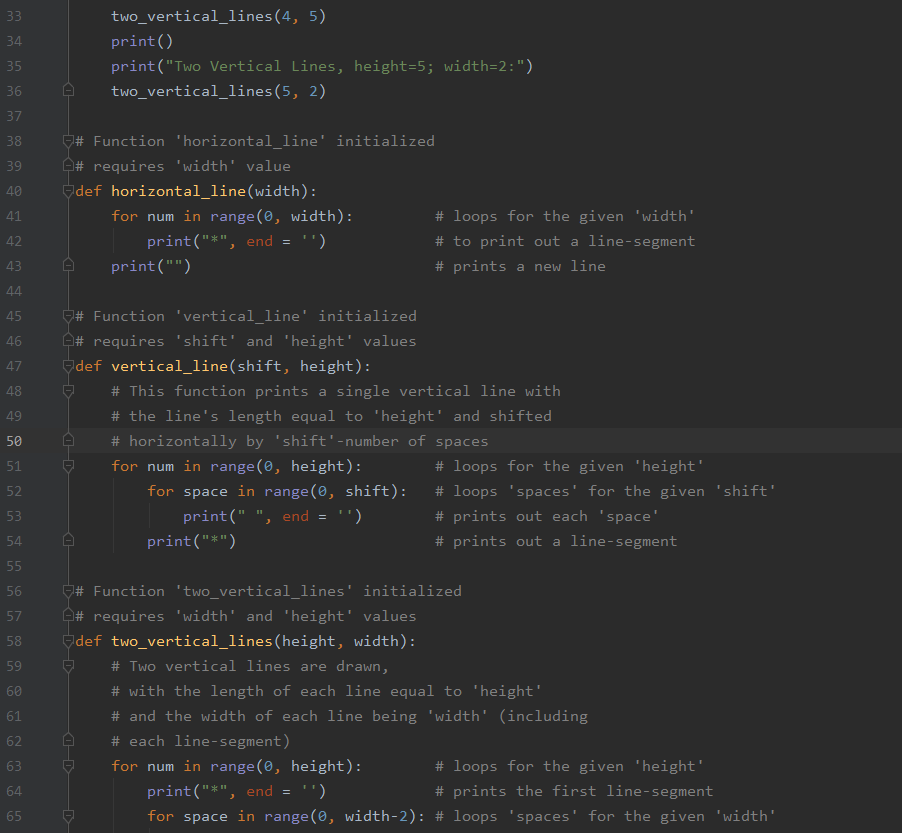
\*\*

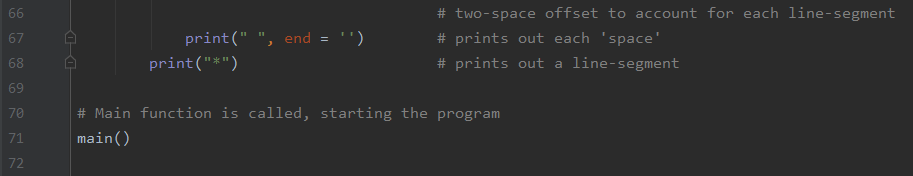
\*\*

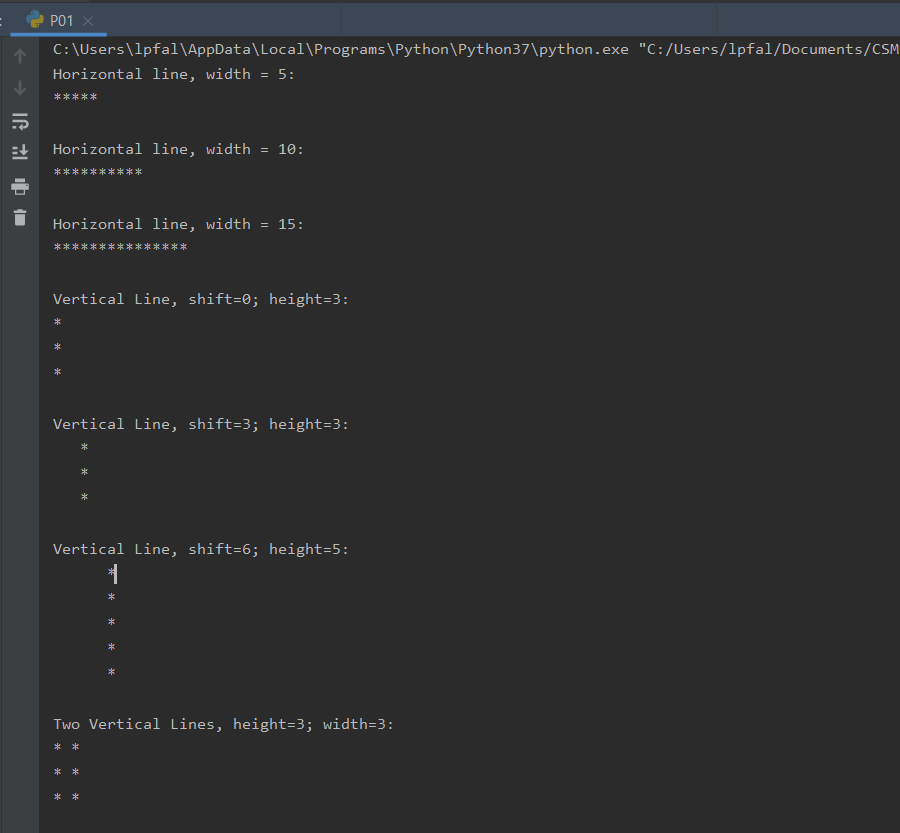
Comment your source code and describe your code to someone who may be viewing it for the first time.

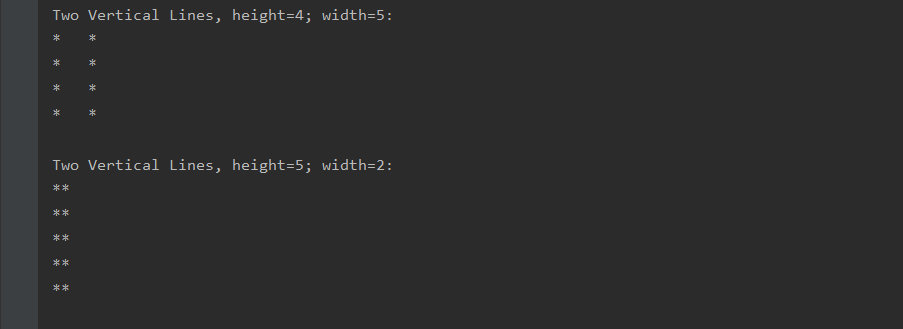
**PLACE SCREEN SHOTS OF THE PYTHON CODE AND ALL I/O BELOW.**











## Write a function called "check\_answer" which will determine if a given addition or subtraction problem was solved correctly.

Here's the information for the function you are to create:

|  |  |
| --- | --- |
| # function: | check\_answer |
| # input: | two numbers (number1 & number2, both integers); an answer (an integer) and an operator (+ or -, expressed as a String) |
| # processing: | determines if the supplied expression is correct. for example, if the operator is "+", number1 = 1, number2 = 2 and answer = 3 then the expression is correct  (1 + 2 = 3) |
| # output: | returns True if the expression is correct, False if it is not correct |

Here's the sample program that you can **will** use to test your function:

answer1 = check\_answer(1, 2, 3, "+")

print (answer1)

answer2 = check\_answer(1, 2, -1, "-")

print (answer2)

answer3 = check\_answer(9, 5, 3, "+")

print (answer3)

answer4 = check\_answer(8, 2, 4, "-")

print (answer4)

And here's the expected output:

True

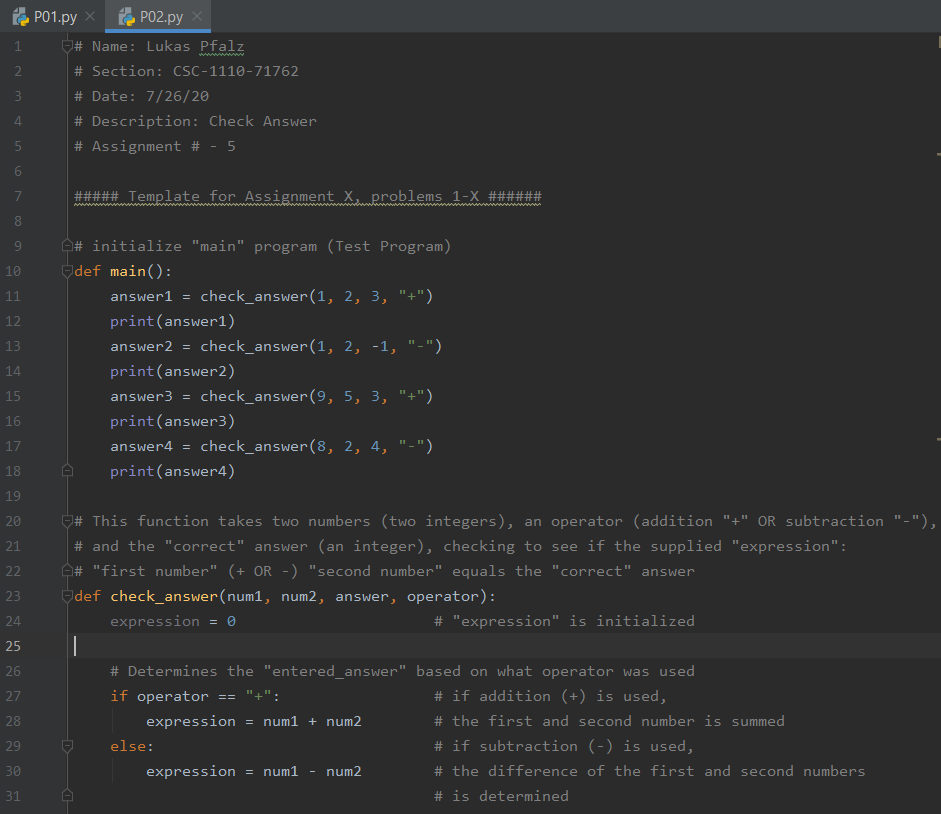
True

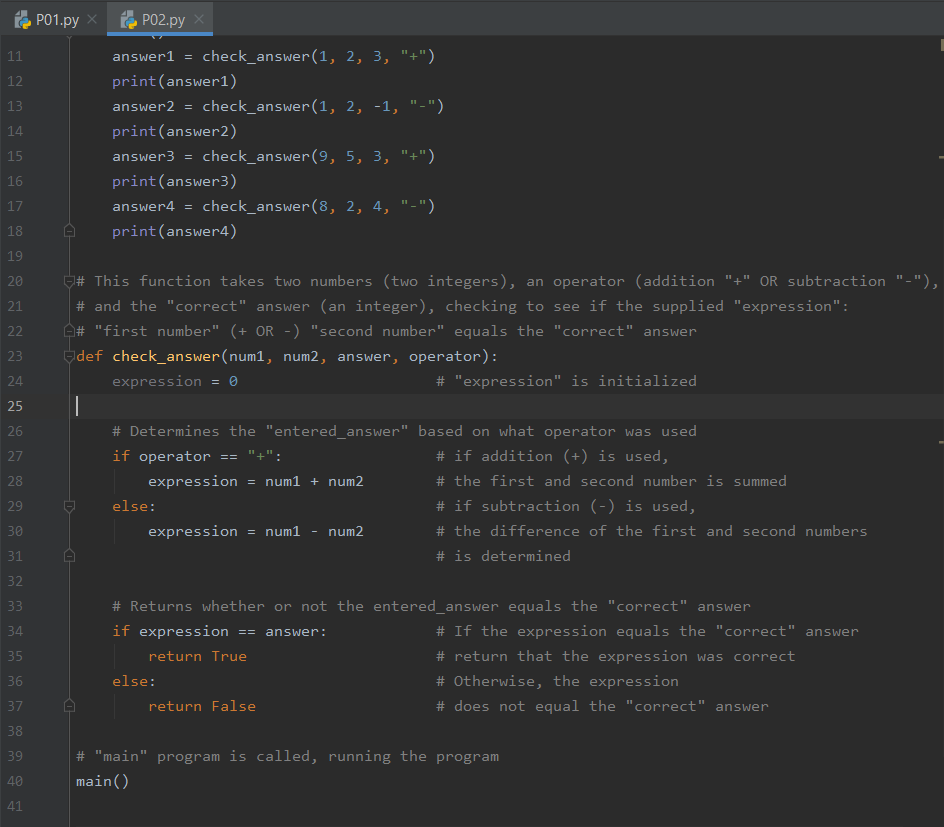
False

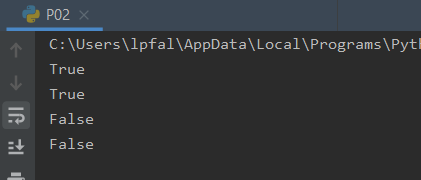
False

Comment your source code and describe your code to someone who may be viewing it for the first time.

**PLACE SCREEN SHOTS OF THE PYTHON CODE AND ALL I/O BELOW.**







## Write a program that lets the user type in the name of a file. Attempt to open the supplied file for read access. If the file exists, you can print out a confirmation message. If the file doesn't exist, you should tell the user that the file cannot be found.

Hint: use a try/except block to do this (don't just use a series of "if" statements - we want this program to be as "generic" as possible).

You will not get credit for this part of the program if you write something like the following to identify valid data files:

filename = input("Enter a filename: ")

if filename == "class1":

# open class1.txt

elif filename == "class2":

# open class2.txt

else:

print ("Sorry, I can't find this filename")

Here's the sample of the program you will use to test this problem:

Enter a class file to grade (i.e. class1 for class1.txt): foobar

File cannot be found.

Enter a class file to grade (i.e. class1 for class1.txt): class1

Successfully opened class1.txt

Comment your source code and describe your code to someone who may be viewing it for the first time.

**PLACE SCREEN SHOTS OF THE PYTHON CODE AND ALL I/O BELOW.**

