CSCI 1411: Fundamentals of Computing
Lab 4
Due Date: September 22, 2023

Name: Brandon Perez

Goals:

- Use of variables of type int, float, and string
- Use of operators +, -, /, *, **, %, //
- Use of Python function math.ceil()
- Understanding integer and float operations

Development Environment: IDLE

Deliverables:

1) This completed document with required screen shots.
2) Python file created for the second part of the lab. Name the file using the following format: `lastnameLab04Part2.py`.

How to take a **screen shot**:

- For a Windows 10: Use Snipping Tool to copy and press CTRL + V to paste screen shot.
- For Mac: Press Shift + Command (⌘) + 4 to copy and press Command (⌘) + V to paste screen shot.

Part I – Skills Practice

A great (and fun) way to practice Python is to use IDLE shell to evaluate simple expression. In this lab we will be studying the following Python operators:

Operator	Name	Semantics	Example	
			Expression	Result
+	Addition	a + b add a and b	5 + 4	9
-	Subtraction	a – b subtract b from a	5 – 4	1
*	Multiplication	a * b multiply a and b	5 * 4	20
/	Division	a / b divide a by b	17 / 4	4.25
**	Exponentiation	a ** b a raised to the of b	5 ** 4	625
%	Modulo	a % b remainder when a is divided by b	17 % 4	1
//	Integer (or Floor) Division	a // b integer quotient when a is divided by b	17 // 4	4

Notes:

- You can also use variables with Python operators to form an expression.
- You can use more than one operator in an expression.
- Modulo (%) and Integer Division (//) operators are generally used with integer data only.

Using Python Operators with Integers:

Evaluate the expressions given in the following table using Python shell. Write the result in the column next to the expression. If executing an expression result in an error the write the error message in the column next to the expression. Write one sentence interpreting the result in the next row.

Expression	Result
19 + 5	24
Interpretation:	Result is is the sum of 19 and 5
19 5 +	SyntaxError: invalid syntax
Interpretation:	Syntax error
18 – 5	13
Interpretation:	Result is the difference of 18 and 5
5 – 18	-13
Interpretation:	Result is the difference of 5 and 18
18 * 5	90
Interpretation:	Result is the product of 18 and 5
18 / 5	3.6
Interpretation:	Result is the quotient of 18 divided by 5
5 / 18	2.777777
Interpretation:	result is the quotient of 5 divided by 18
6 ** 4	1296
Interpretation:	Result is 6 to the power of 4
79 % 17	11
Interpretation:	result is the remainder of 79/17 (modulo)
79 // 17	4
Interpretation:	Result is the floor division quotient of 79 divided by 17
17 + 4 * 5	37
Interpretation:	Result is the sum of 17 and the product of 4 and 5
(17 + 4) * 5	105
Interpretation:	Result is the product of 5 and the sum of 17 and 4
a = 14 b = 5 a + b	19
Interpretation:	Result is the sum of 14 and 5 (a and b respectively)
a = 14 a + c	Error/Unknown
Interpretation:	Syntax error since c is undeclared OR unknown since the value of c is not shown

Using Python Operators with Floats:

Evaluate the expressions given in the following table using Python shell. Write the result in the column next to the expression. If executing an expression result in an error the write the error message in the column next to the expression. Write one sentence interpreting the result in the next row.

Expression	Result
19.5 + 5.6	25.1
Interpretation:	Result is the sum of 19.5 and 5.6 (a float)
18.6 – 5.9	12.7
Interpretation:	Result is the difference of 18.6 and 5.9 (a float)
5.9 – 18.6	-12.7
Interpretation:	Result is the difference of 5.9 and 18.6 (a float)
18.2 * 5.3	96.46
Interpretation:	Result is the product of 18.2 and 5.3 (a float)
18.2 / 5.3	3.43396226415
Interpretation:	Result is the quotient of 18.2 divided by 5.3
5.3 / 18.2	0.2912087912
Interpretation:	Result is the quotient of 5.3/18.2
6.5 ** 4	1785.0625
Interpretation:	Result is 6.5 to the power of 4
17.9 + 4.2 * 5.5	41
Interpretation:	Result is the sum of 17.9 and the product of 4.2 and 5.5
(17.9 + 4.) * 5.5	120.45
Interpretation:	Result is the product of 5.5 and the sum of 17.9 and 4

Using Python Operators with Strings:

Can we use Python operators with strings? Let's try some examples. Evaluate the expressions given in the following table using Python shell. Write the result in the column next to the expression. If executing an expression result in an error the write the error message in the column next to the expression. Write one sentence interpreting the result in the next row.

Expression	Result
'apple' + 'banana'	'applebanana'
Interpretation:	result is the concatenation of apple and banana with no space
'apple' - 'banana'	Syntax Error
Interpretation:	- is not a recognized symbol for concatenating strings
'apple' * 'banana'	Syntax Error
Interpretation:	* is not a recognized symbol for concatenating strings
'apple' / 'banana'	Syntax Error
Interpretation:	/ is not a recognized symbol for concatenating strings
'Hi'*3	Syntax Error
Interpretation:	* is not a recognized symbol for concatenating strings
first_name = 'Joe' last_name = 'Doe' first_name + last_name	'JoeDoe'
Interpretation:	Result is the concatenated strings without a space represented as variables

Using Python math function math.ceil()

math.ceil() converts a floating point number into integer which is bigger than or equal to the given number. You will have to import math library to be able to use math.ceil() function. You can do this by typing import.math() in the IDLE shell or my including this statement as the first line in your Python file. Fill in the following table with your results.

Expression	Result
math.ceil (4.1)	5
math.ceil (4)	4
math.ceil (17)	17
math.ceil (17.1)	18
math.ceil (17.5)	18
math.ceil (17.8)	18

Part II – Float versus Integer data types

In this part of the lab, we will be writing some code that demonstrates the difference between floating point division and integer division.

Problem statement:

- You have a **box** that has a length and width in inches. These lengths and widths are not restricted to be in whole numbers (e.g, the length could be 11.2 inches)
- You want to put trim around the box, but the local HW store only sells trim in 12" segments.
- A 12" segment of trim costs $1.88.

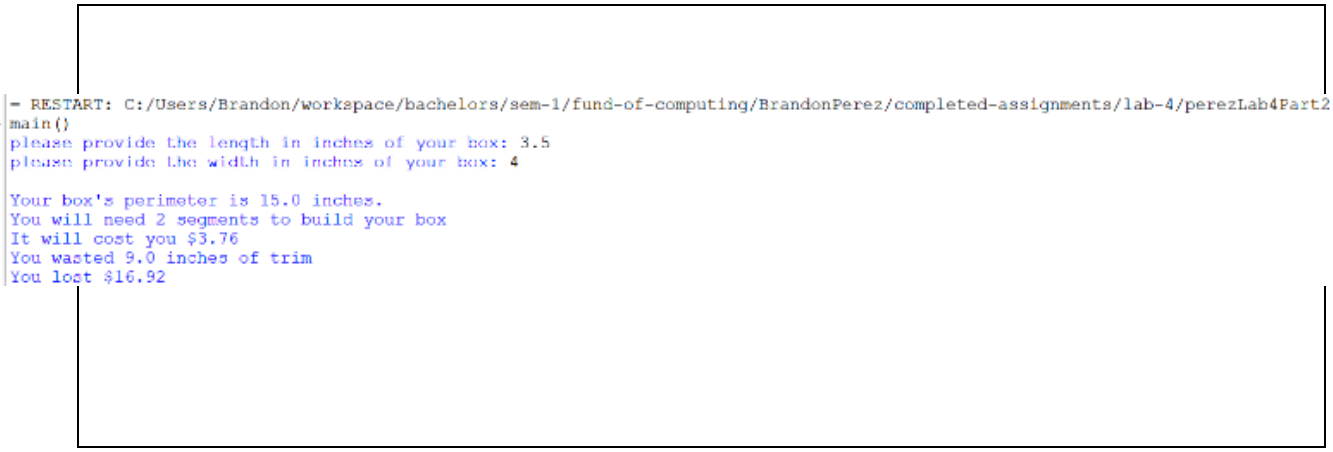
Write a program that does the following things:

1. Ask the user the length, in inches, of the box
2. Ask the user the width, in inches, of the box
3. Calculates the perimeter of the box and prints that out
   perimeter = 2 * length + 2 * width
4. Calculates and the number of segments needed to trim the box (go around the perimeter).
   a. Calculate the amount of trim needed by dividing the perimeter by 12.
   b. Calculate the actual number of segments by applying the math.ceil function to the result of part a.
5. Calculates the cost of the trim (use the actual number of segments calculated in part b above and multiplying it by cost of each segment).
6. Print the following items:
   a. Perimeter of the box
   b. Number of trim segments that you have to buy (calculated in 4b)
   c. Cost of the trim segments (calculated in 5)
   d. Print amount of wasted trim. This is difference between actual number of segments (calculated in 4b) and segments needed (calculated in 4a)
   e. Print amount of $$ you lost. This can be calculated by multiplying the difference between actual number of segments (calculated in 4b) and segments needed (calculated in 4a) by trim cost.

Sample Output:

```
>>>
    = RESTART: /Users/lakhanis/Desktop/Labs/Lab_04_Arithmetic_Operations/Sample Solution
    /lab4Pert2.py
    Enter the length of the box in inches: 99
    Enter the width of the box in inches: 7.2
    Your box has a perimeter of 212.4
    Number of trim segments that you have to buy 18
    Total cost 33.839999999999996
    Amount of trim wasted 0.3000000000000007
    Cost of wasted trim 0.5640000000000013
```

Take a screen shot of your result and post it in the box below:



Every program should have the following comment block at the top. Make sure to fill in your name, class with section number, due date, brief description of your program, and status of your program:

```
#
# Name:
# Class: CSCI 1411-00X
# Due Date:
# Description:
# Status:
```

Rubric for Lab 4:

Criteria	Rating
Part I Using Python Operators with Integers	1 point for each correct result and interpretation. Total points: 12
Part I Using Python Operators with Floats	1 point for each correct result and interpretation. Total points: 9
Part I Using Python Operators with Strings	1 point for each correct result and interpretation. Total points: 6
Part I Using Python math function math.ceil()	0.5 point for each correct result. Total points: 2
Part II (Screen shot 2)	Screen shot included – 5 points No screen shot included – 0 points
Part II (Python Code)	Runs as expected – 25 points Runs as expected but with incorrect results – 5 points Does not run or Python file is not included – 0 points
Total Points	59