# **CIS 61 :: Lab 01 - Expressions and Names**

#### **Student Name: Collin Hargreaves**

| **Instructions:**  1. *Make a copy of the assignment template. Go to File => Make a copy (or download as a Word file.)* 2. Attach Snipping Photos for each question. 3. *Place your name in the Title of each Assignment*    1. *For Example: CIS 61 - Lab 01 - Expression and Names - Irfan O***.** 4. **Submission:** When done, go to **File -> Download as -> Microsoft Word** and then upload the file to Canvas. |
| --- |

*The lines in the triple-quotes """ are called a* ***docstring****, which is a description of what the function is supposed to do. When writing code in 61A, you should always read the docstring!*

*The lines that begin with >>> are called* ***doctests****. Recall that when using the Python interpreter, you write Python expressions next to >>> and the output is printed below that line. Doctests explain what the function does by showing actual Python code: "if we input this Python code, what should the expected output be?"*

The below command will help you test the python file

**> python3 -m doctest -v filename.py**

**Lab 1 - Expressions and Functions**

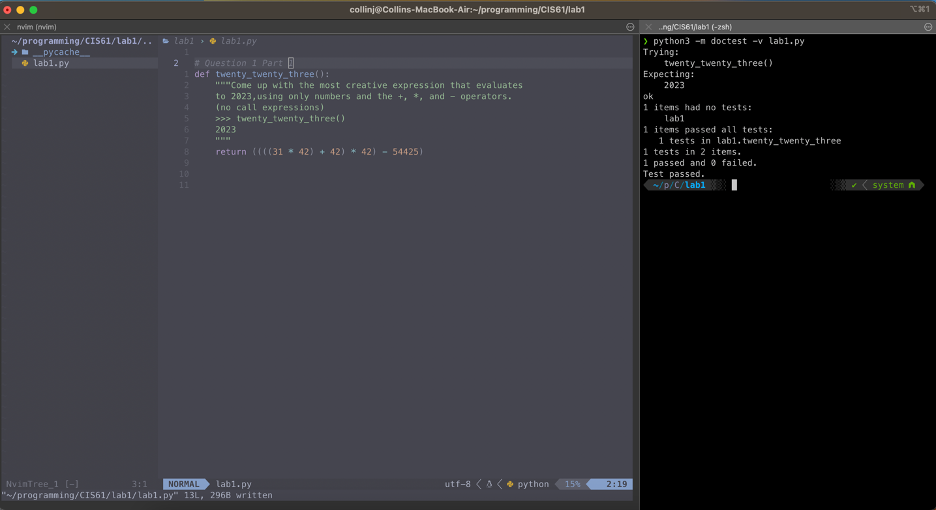
**Instructions**: Use Sublime text editor to write your code and use Python shell to execute the below programs. Attach Snipping photos of **your source code** and **executions of the code in Python shell**.

**Question 1**: **Twenty-Twenty-Three**

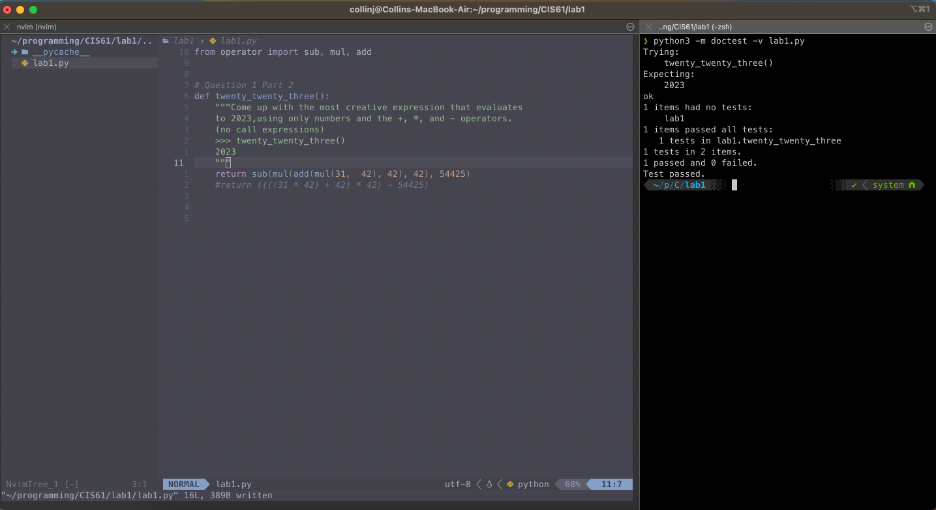
**Part 1:** Come up with the most creative expression that evaluates to 2023, using only numbers and the [+, \*, - ] operators.   
You should replace the underscores in return \_\_\_\_\_\_ with the expression that evaluates to 2023.

**Do not print the result.**

| **def** **twenty\_twenty\_three**():  """Come up with the most creative expression that evaluates   to 2023,using only numbers and the +, \*, and - operators.  (no call expressions)  >>> twenty\_twenty\_three()  2023  """  \*\*\* YOUR CODE HERE \*\*\*  **return** \_\_\_\_\_\_\_ |
| --- |

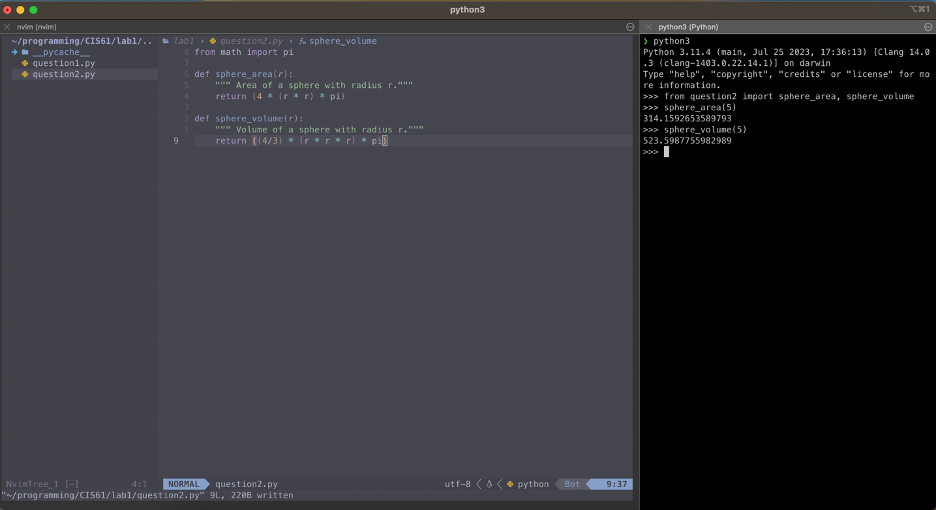
**Sample Submission[DELETE THIS]:  
**

**Part 2:** Try to rewrite the same expression, this time entirely with call expressions   
 (using function calls: add, mul, sub, etc… You can use **from operators import add, mul, sub**)

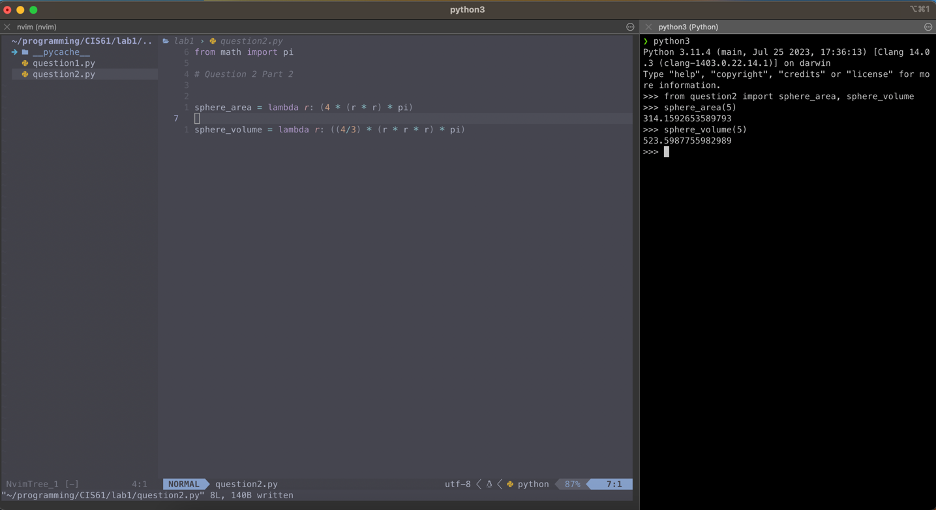


**Question 2**: **Area**  
**Part 1:** Write definitions for these functions:   
sphereArea(radius) returns the surface area of a sphere having the given radius.   
sphere Volume (radius). Returns the volume of a sphere having the given radius.

| **from math import pi**  **def** **sphere\_area**(r):  """ Area of a sphere with radius r."""  \*\*\* YOUR CODE HERE \*\*\*  **def** **sphere\_volume**(r):  """ Volume of a sphere with radius r."""  \*\*\* YOUR CODE HERE \*\*\* |
| --- |



**Part 2:** Rewrite the above functions with lambda expressions and assign them to respective names. You can just use a Python shell and take the screenshot of the code.



**Question 3: Rain or Shine**

**Part 1:** Alfonso will only wear a jacket outside if it is below 60 degrees or it is raining.

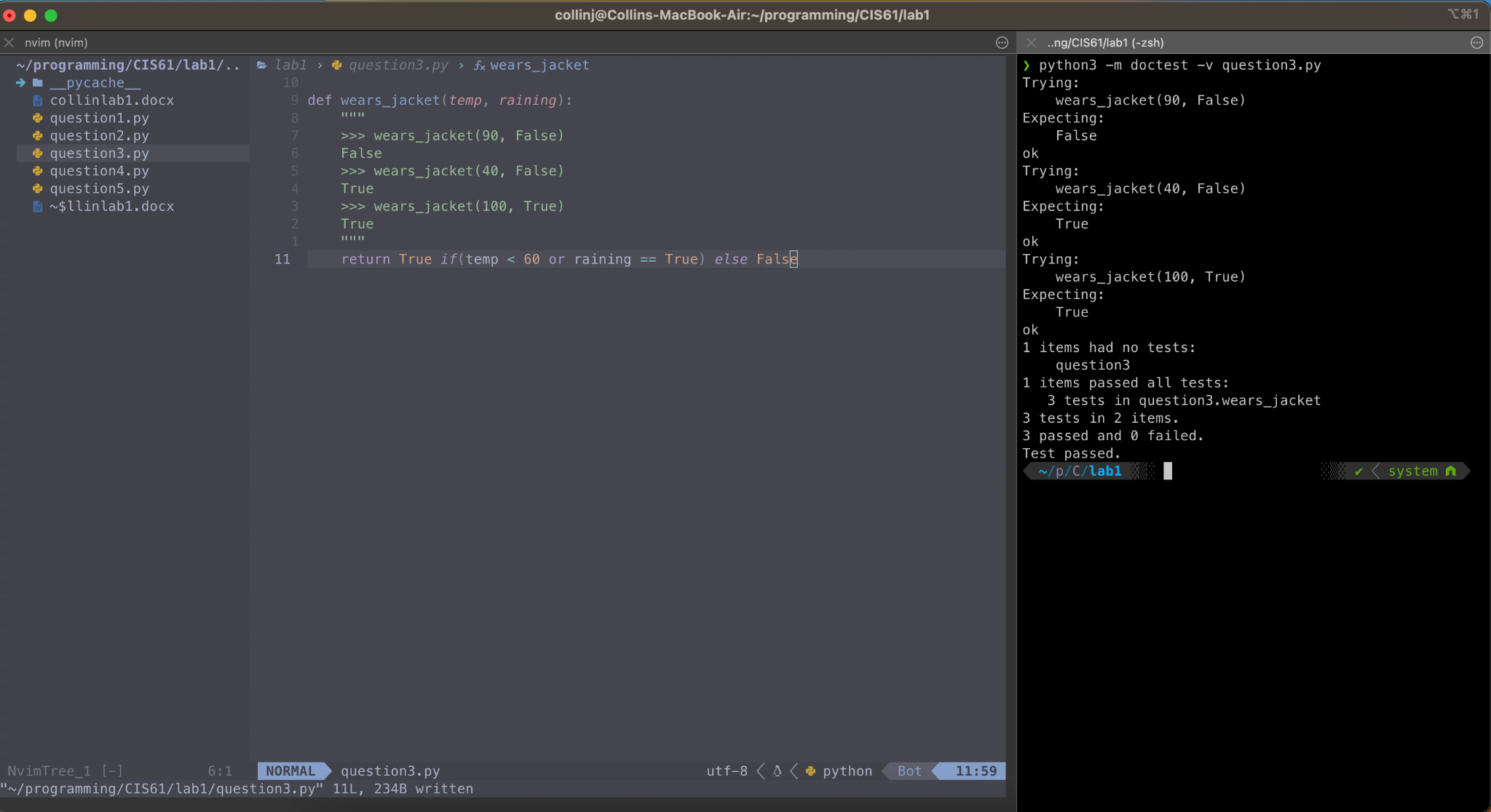
Write a function that takes in the current temperature and a boolean value telling

if it is raining and it should return True if Alfonso will wear a jacket and False otherwise.

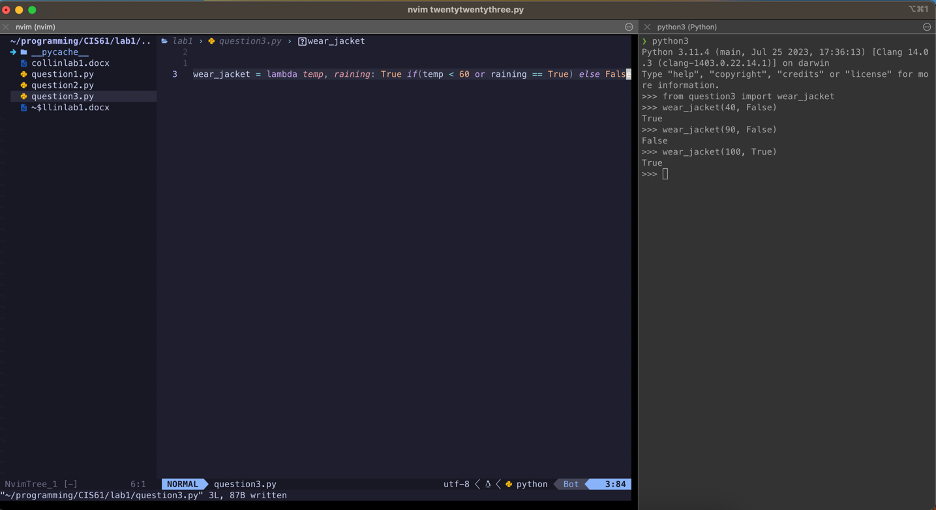
**Try solving this problem with a single line of code.**

| **def wears\_jacket(**temp, raining**):**  """  >>> wears\_jacket(90, False)  False  >>> wears\_jacket(40, False)  True  >>> wears\_jacket(100, True)  True  """  \*\*\* YOUR CODE HERE \*\*\* |
| --- |

*Note that it should either return* ***True*** *or* ***False*** *based on a single condition, whose truthiness value will also be either True or False.*

**

**Part 2:** Rewrite the above function with a lambda expression. You can just use a Python shell and take a screenshot of the code.



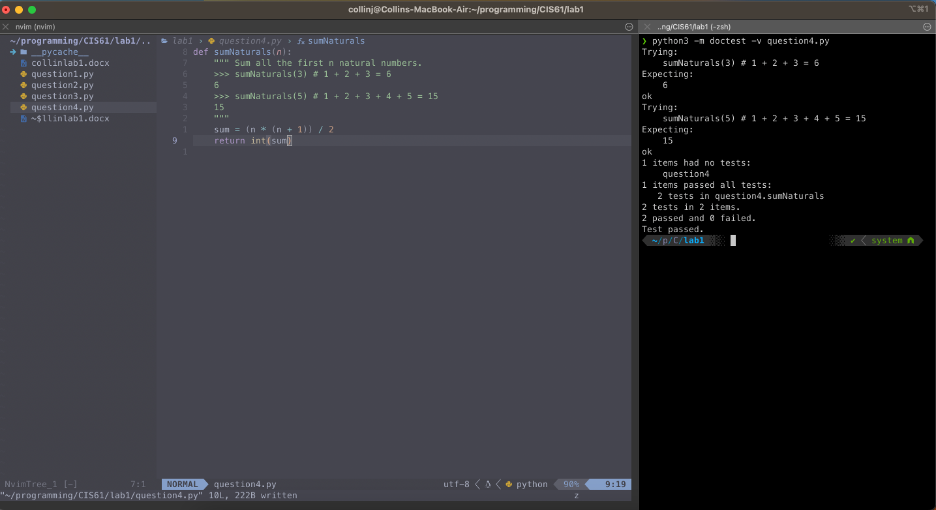
**Question 4: Sum of the first N natural numbers**:

**Part 1:** Write a function sumNaturals (n) that returns the sum of the first n natural numbers. You can use this formula 1 + 2 + ... + n = n(n+1) / 2.

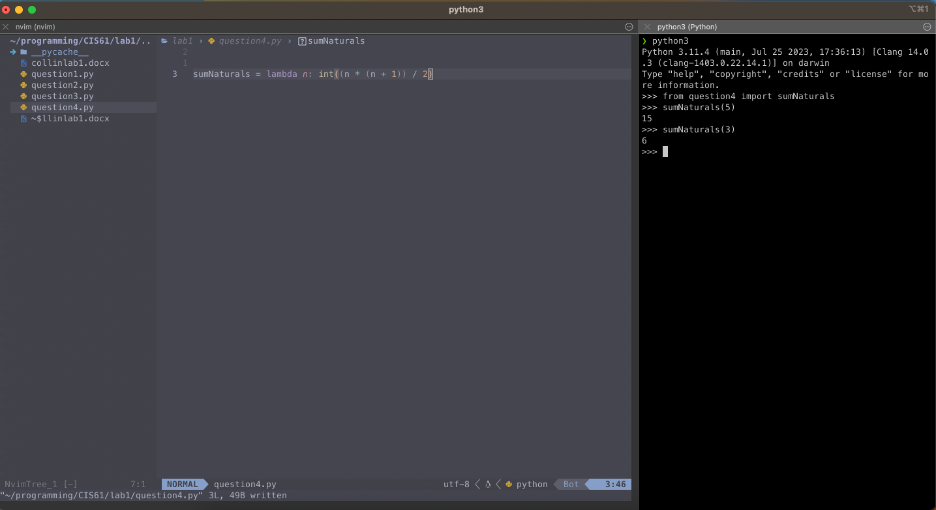
Make sure that the function returns an integer.

**Do not use a for loop or a while loop**.

| **def** **sumNaturals**(n):  """ Sum all the first n natural numbers.  >>> sumNaturals(3) # 1 + 2 + 3 = 6  6  >>> sumNaturals(5) # 1 + 2 + 3 + 4 + 5 = 15  15  """  \*\*\* YOUR CODE HERE \*\*\* |
| --- |



**Part 2:** Define a lambda expression that takes **n** and returns the sum of the first **n** natural numbers, using the above formula. You can just use a Python shell and take a screenshot of the code.

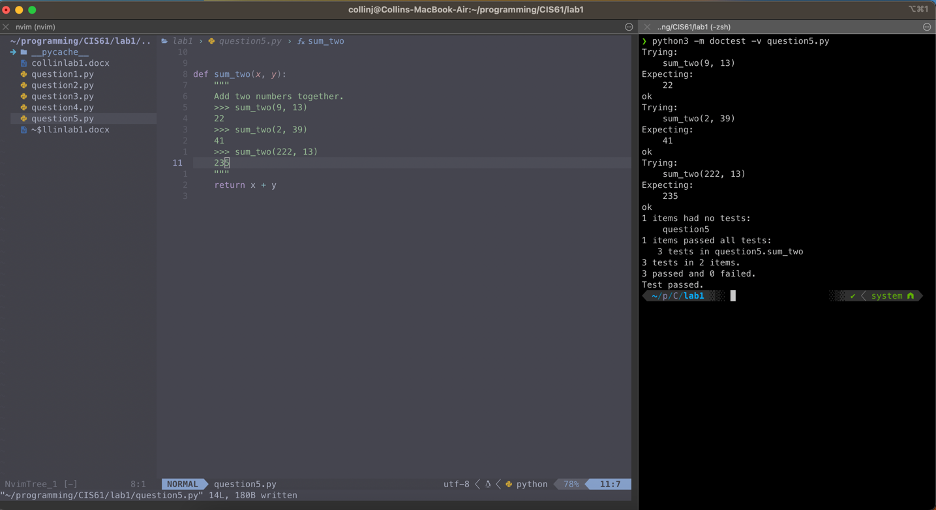


**Question 5: You Define a Function**

Part 1: Write a function that takes in one or two input parameters and returns an output. The function should return the output of a **one-line expression**. Write at least three test cases for your function in the docstring. Use the command line to test your function against the test cases. Take a screenshot of your code and the result of your test. Also, write the function in the below box as well.

Make sure your function has just one line of code

| Def sum\_two(x, y):  Return x + y  Sum\_two = lambda x, y: x + y |
| --- |



Part 2: Write the same function as a lambda function.

