



School of Science, Computing and Artificial Intelligence
The University of the West Indies, Five Islands

COMP1205 - Lab 6:

Functions

1. What does the following code print? Explain.

```
number = 50
def func(number):
    print(number)
    number = 2
    print(number)
func(number)
print(number)
```

2. write a program that prompts the user for the radius and height of a 3-dimensional cone and then calculates and prints the surface area and volume of the cone. The calculation of the surface area and the volume will be done in functions, as will the gathering of the inputs.

Part I:

Your program for this part will function as follows:

- (a) Print out a message indicating what the program does.
- (b) Prompt the user for the radius (a non-negative float) in feet.
- (c) Prompt the user for the height (a non-negative float) in feet.

- (d) Print the radius and height, but rounded to 2 decimal digits.
- (e) Print the surface area and volume, rounded to 2 decimal digits.

This version of the program will not check that the input is correct; if the user types anything other than a non-negative float, it can crash.

Your program must define and invoke two functions:

```
cone_surface_area(r,h):returns the surface area of a cone of radius r and height h
cone_volume(r,h) : returns the volume of a cone of radius r and height h.
```

The formulas for finding these values are as follows:

```
surface area = pi * r * (r + sqrt(r^2 + h^2))
volume = pi * r^2 * h / 3
```

Import the math module and use math.pi and math.sqrt() in these calculations.

Show the Code to Instructor

Part II:

Next, you will replace the input commands that you used to get the radius and the height in the previous program with a function that encapsulates the input processing to check for correct input. You will define two functions for this purpose.

`is_nneg_float(s)`: checks if string `s` denotes a non-negative floating point value (not in scientific notation). This function should return `True` if `s` contains (at most) one decimal point and one or more digits (and nothing else); and `False` otherwise. According to this definition an integer should return `True`. (See below for some examples illustrating calls to this function.)

`get_nneg_float(p)`: repeatedly prompts the user using `p` until the user inputs a non-negative floating point value, returning the non-negative float that was input (as a float, not a string). This function should invoke the previous function to check the input string before converting it to a float. To clarify how these functions are to behave, the result of some example calls follow:

```
>>> is_nneg_float("2.15")
True
```

```

>>> is_nneg_float("3.")
True
>>> is_nneg_float(".5")
True
>>> is_nneg_float("123")
True
>>> is_nneg_float("-12")
False
>>> is_nneg_float("1e10")
False
>>>

>>> get_nneg_float("Enter a non-negative float (NN.NN): ")
Enter a non-negative float (NN.NN): -3
Must be a float value; please try again.
Enter a non-negative float (NN.NN): 3e10
Must be a float value; please try again.
Enter a non-negative float (NN.NN): 2.1
2.1
>>> get_nneg_float("Float, please (NN.NN): ")
Float, please (NN.NN): 105
105.0
>>> get_nneg_float("And another in same form (NN.NN): ")
And another in same form (NN.NN): x10
Must be a float value; please try again.
And another in same form (NN.NN): .19876
0.19876

```

Replace the input commands in your first program with appropriate calls to the second of these functions.

3. Write a function that takes three integers as its input and returns whether a triangle with those side lengths is a right triangle or not. Hint: sort is your friend.