CSCI 220L

Lab 2

Learning Objectives

- Develop simple Python programs that read input, produce output, and do arithmetic.
- Use the Python `math` module.
- Create simple for loops.

Activities

Complete the following activities in the given Python file. For each activity, demonstrate your working solution to one of the instructors. Make sure to include meaningful comments and output messages to receive full credit.

1. Shooting percentages

   The players on a basketball team want to know their shooting percentages (percentage of successful shots). Complete the function `get_shooting_percent()`, which should ask the user how many shots were attempted and how many were successful, and then display the corresponding shooting percentage. For example, if you attempt 10 shots and make 5 of them, your percentage is 50%.

2. Volume

   Find the function `calc_volume()` in the given file and modify it to calculate the volume of a rectangular solid. The function should ask the user to provide the solid's length, width, and height. Here's a possible example of how your function's output may look (you don't have to follow this format exactly):

   ```
   This function calculates the volume of a rectangular solid.
   Enter the length: 25
   Enter the width: 12
   Enter the height: 18
   The volume of the solid is 5400
   ```

3. One-line volume

   Modify the previous function to accept all three numbers on one line, separated by commas. Hint: Use simultaneous assignment.

4. Name and age

   Write a new function, `greet_better()`, that asks the user for their name and age, and then tells them their age in five years. For example:

   ```
   This function tells you your age in five years.
   Enter your name: Elessar
   Enter your age: 87
   Hi Elessar! In five years, you will be 92.
   ```

5. Coffee shop accounting

   Write a function, `compute_coffee_cost()`, to determine the price of shipping coffee orders. You know that coffee costs $7.50 per pound, shipping costs $0.44 per pound, and there is a fixed cost of $3.00 per order for overhead. Your function should ask the user how many pounds of coffee were purchased and use it to display the total cost.

6. Area of a triangle

   Your friend Bill has been hired to paint the interior of a new museum in downtown Charleston. The building is modern themed and features a multitude of triangular walls and ceilings that need to be painted. Bill wants to know the square footage of each surface in order to buy the correct amount of paint, so you have agreed to help him by writing a program to calculate the area of each triangular surface. Bill will measure the side lengths of each triangle and enter them into your program, which should calculate and display the corresponding area in square feet.

   You recall that the area of any triangle can be found using Heron's Formula:

   $$p = \frac{a+b+c}{2}, \qquad area = \sqrt{p(p-a)(p-b)(p-c)}$$

   (where *a*, *b*, and *c* represent the lengths of the three sides)

   Write a function, `measure_triangle_area()`, that uses the above formulas to calculate the area of a single triangle. The function should ask the user for the length of the three sides. Your solution should use the `math.sqrt()` function to compute the square root (there are other ways, but this will give you practice with the `math` module). See the attached appendix for a review of how to use modules in Python.

   Note: Not every combination of three side lengths represents a possible triangle. For example, you can't make a triangle with side lengths 1, 2, and 3. In chapter 7, you will learn how to make a program that rejects invalid inputs, but for now this isn't a concern.

7. Sum of squares

   Write a function, `sum_squares()`, that calculates the sum of the squares of the integers in a particular range. The function should allow the user to input the starting and ending numbers of the range. For example, if the user enters 3 and 5, your function should compute $3^2 + 4^2 + 5^2$ and display the sum of 50.

When you're done, upload your final Python file to the OAKS dropbox.

Appendix: Python Modules

In addition to the variety of built-in functions that you've already been using (see the full list here), Python also includes a large selection of *modules* that offer specialized tools. A module is a collection of pre-defined variables and functions that you can import into your programs. Python ships with about 200 built-in modules. Here are a few you may find interesting:

- `calendar` – tools for working with dates and times
- `email` – tools for reading and sending emails
- `math` – common math variables and functions, like *pi*, *sin(x)*, *factorial(x)*, etc.
- `statistics` – common statistical functions
- `string` – tools for working with text
- `threading` – tools for allowing multiple programs to run simultaneously
- `urllib` – tools for connecting to the Internet (the name is short for "URL library")

You can read about all the built-in modules here: docs.python.org/3/py-modindex.html.

To get access to the contents of a particular module, use an *import statement*. For example, to use the `math` module, you would write:

```
import math
# all lines after this can now use the math module
print(math.factorial(8))
print(math.sqrt(16))
```

By convention, you should place all of your import statements at the top of your program. You can even import multiple modules on a single line:

```
import math, statistics
```

You can read about more ways to import modules in Python here: www.geeksforgeeks.org/import-module-python/.