Lab 1: Python Basics I

INFSCI 0201
Intermediate Programming (Spring 2025)

Topics Reviewed

1. Variables
2. Constants
3. Data Types
4. Typecasting
5. Handling user input
6. Working with strings
7. Control structurals

Grading and Submission

You are to write a complete Java program that meets the requirements outlined in the Lab 1 Tasks section.

Once you have completed the program, you should push your solution to GitHub.

There will be three groups of requirements for this lab:

1. Pythagorean Theorem Solution
2. Area and Perimeter Solution
3. Unit Convertor

Note that if your program is not named properly, it cannot be tested.

Lab 1 Tasks

- Create a new Python project folder in the IDE of your choice named *lab1*. (this should be in the lab01 folder of your git repo!)
    - So, the path to your codes should be in “labs/lab01/”
- Python files: there are 3 problems in this assignment, and each of them has to be written in a different Python file (.py file). All files should be within the lab1 package in the *lab1* project!
- Do not use numeric literals – all numbers must be assigned to variables or constants! Any fixed factor used in a computation should be defined as constant.
- Follow the naming conventions for naming your variables and constants. All three problems can be solved with what has been covered in the first 2 weeks of class. Be aware that if you search for answers, you might be given other solutions that involve more advanced stuff. For this reason, we are giving precise enough instructions and tips.

Problem 1: Solve Pythagorean theorem

- Write your solution to this problem in the main method of a file named "**pythagorean.py**"
- Using the *input()* function, ask the user to input 2 numbers, corresponding to the sides of the triangle. Consider asking twice (one number each time) and storing them in string variables.
- Convert those variables to numbers. Assume the user is kind and will enter a number (you do not have to check the input before converting it to a number. We will see how to do this later in the course.)
- Create a variable for computing the hypotenuse and its value to the square root of the sum of the squares of the other two sides. For example, if your variable for the hypotenuse is "c" and the sides are "a" and "b", then:
    - Use the sqrt method of Python's math library to calculate the square root
    - Use the ** operand to calculate the squares of the sides
    - Use the *print()* function to display your result. In the output, give a complete sentence. For example, "The hypotenuse is 5.0".
- You might get an output with many decimal numbers - you need to shorten the number to a certain amount of decimals. Research the solution on your own - the output/result should be limited to 2 decimal points.

Problem 2: Area and perimeter of a circle

- Write your solution to this problem in a Python file named “**circle.py**”
- Ask the user to input the radius of a circle and convert the input to a numeric variable. (Be aware of random input from users)
- Compute the perimeter and the area. Use the predefined constant *math.pi*.
- Give an output in complete sentences ("The circle with radius 10 has an area of 314.159 and a perimeter of 62.83"). If you have many decimals, shorten the number of decimals to 2.

Problem 3: Unit Convertor

In this task, you are going to develop a simple metric/imperial system converter to convert:

- centimeters to inches (cm to in)
- inches to centimeters (in to cm)
- yards to meters (yd to m)
- meters to yards (m to yd)
- ounces to grams (oz to gm)
- grams to ounces (gm to oz)
- pounds to kilograms (lb to kg)
- kilograms to pounds (kg to lb)

1. Write your solution to this problem in a Python file named "***unitconverter.py***"
2. Using the input() function, ask the user to input a distance or a weight amount. Assume that the input string will contain a number (with or without decimals) followed by a space followed by a unit of measurement that can be one of the following: in, cm, yd, m, oz, g, kg, lb.

   Here are eight examples of valid inputs:
   a. "111.5 in"
   b. "19.342124 cm"
   c. "55 yd"
   d. "10.7 m"
   e. "10 oz"
   f. "8.3 g"
   g. "23.2 kg"
   h. "172.3 lb"
3. Process the input to store the number and the unit separately in two variables. Remember to name the variables following the naming conventions covered in class. **TIP**: you know that the value and the unit of measurement are separated by a space (one space character). It might help to look at Python string operations!

4. The number should be transformed to the other measurement system depending on the unit value. Here are those conversion factors:
   1 in = 2.54 cm
   1 yd = 0.9144 m
   1 oz = 28.349523125 g
   1 lb = 0.45359237 kg

   You should be able to handle each of these conversions (in both directions).

5. Display the result of the conversion. Your output should contain both the original value and units and the converted value and units.

   For example, if the input is "19.342124 cm", the system should output:

   19.342124 cm = 7.62 in

   Note: As in the sample output, you must round your converted value to include only 2 values after the decimal point (you can use the same trick you used in the previous lab…), but you should not change the user's input.