

IO, Types, and Operators

You must get checked out by your lab CA prior to leaving early. **If you leave without being checked out, you will receive 0 credits for the lab.**

Restrictions

The Python structures that you use in this lab should be restricted to those you have learned in lecture so far. Please check with your course assistants in case you are unsure whether something is or is not allowed!

If you do not have Python running on your computer please go back to Lab 0 and set it up before moving on

For Problem 1, write your answers on paper or in a text editor on your computer.

Problem 1: Constantly Variable

Part 1A: Constant or Not:

State whether each of the following variables is a constant or not.

1. `FEET_IN_oNE_mILE` = "five thousand two hundred and eighty feet"
2. `anotherVariable` = 828.4
3. `INCHES_IN_ONE_FOOT` = 12
4. `THINGS_MORE_FUN_THAN_LAB` = 0
5. `descriptive_variable_name` = True

Part 1B: Data Types:

State the data type for each variable shown in Part 1A.

Create a new python file for each of the following questions.

Problem 2: Baking Scones?

You have been wanting to bake scones and have found a recipe, but the recipe uses the metric system and you only have measuring cups. Let's convert the metric measures to customary measurements.

The following are the metric measures for **10 scones**:

```
75 g salted butter  
350 g flour  
150 ml milk
```

Here are some conversion factors you can use:

```
75 grams butter = 1/3 cup of butter  
150 gram flour = 1 cup flour  
100 ml milk = 1/2 cup milk
```

Your program should take user input for the number of scones they want to make and print the quantity of each ingredient in customary measurements.

Your code should output the following (disregard small floating point differences):

```
Enter the number of scones you want to make: 25
To make 25 scones use 0.8333333333333334 cups butter,
5.833333333333333 cups flour, and 1.875 cups milk
```

Problem 3: Time in seconds

This program will ask the user for **four inputs**: a number of days, number of hours, number of minutes, and number of seconds. This may look something like:

```
How many days do you have?
How many hours do you have?
How many minutes do you have?
How many seconds do you have?
```

You may assume that the user will always input a positive whole number. After getting the four inputs, the function should calculate how many seconds in total are in the given number of days, hours, minutes and seconds, and output the result. The final output of the program should be something like this:

```
How many days do you have? 3
How many hours do you have? 7
How many minutes do you have? 41
How many seconds do you have? 16
3 Days 7 Hours 41 Minutes and 16 Seconds results in a total of
286876 Seconds.
```

Problem 4: Expression Marathon

In this problem, you will look at the variables defined, and then evaluate them. All changes made to variables remain when moving onto the next expression!

```
A = True  
B = True  
C = False  
D = ""  
E = 0  
F = not C
```

```
(A or B) and (not C and A)
```

```
(C and B) or (A or C)
```

```
(C and B) or (A or C)
```

```
(A and D) or (E == 5 - 5)
```

```
not (F and B and E or A)
```

```
(64 == 2**6) and (E or F)
```

```
D = "a"  
six = (D and B) and (F == E)
```

```
E += 1  
seven = (7 % 2 == 0) or (E and F)
```

```
(not D) or (not F)
```

Problem 5: All Roads Lead To Rome

This problem will convert a decimal number to Roman Numerals.

1. Ask the user to input a decimal integer less than 100.
2. Convert the input to Roman numbers and print to the screen.

Roman numerals and decimals follow the table below.

Decimal	Roman
1	I
5	V
10	X
50	L

For this problem there is no need to consider subtractions.

i.e., 4 = IIII, 9 = VIIII instead of IV and IX respectively.

Explanation:

- *We start by taking the input from the user.*
- *Using integer division and modulus:*
- *Calculate how many L's are needed (number // 50).*
- *Update number with the remainder (number % 50).*
- *Repeat this process for X (10), V (5), and I (1).*
- *Concatenate each Roman numeral symbol the appropriate number of times.*
- *Print the final Roman numeral string.*

Example Outputs:

```
Input: 83
Output: LXXXIII
Input: 44
Output: XXXIIII
Input: 9
Output: VIIII
```