

CSCI 141
Spring 2019

Assignment 1: Variables, `print`, `input`, operators
Due Date: See Canvas

Introduction

This first homework has 2 parts. For the first part please answer the questions on Canvas assigned for this homework, and for the second portion you will complete a single programming task.

Getting Started

Refer to lab 1, as well as the lecture slides, to review. In this and future assignments, you may not have seen all the topics in lecture before the assignment is released, but they will be covered in before the deadline. As usual, seek help early if you get stuck: come talk to me or the TAs during office hours, or visit the CS mentors for help. Please keep track of approximately how much time you spend on both portions of this assignment. You will be asked to report your estimate on Canvas after you submit.

Collaboration and Academic Honesty

The answers to the questions and programming solution **MUST** be your own. You can discuss the problems with your peers, but these discussions must happen away from computers and you should take a break before returning to work on them to help ensure that you truly understand the answers. You may not copy another person's code, or have another person tell you what code to type. If you have any questions, or are unsure about whether a specific sort of collaboration violates academic honesty, please come talk to me.

1 Questions: 16 points

Please answer the questions in the A1 Written quiz on Canvas. The questions on Canvas have been configured so that there is no time limit, but you have only 2 attempts to submit your answers. The score that is recorded in Canvas is the score that is the latest (most recent submission) of your attempts.

2 Programming Task: 20 points

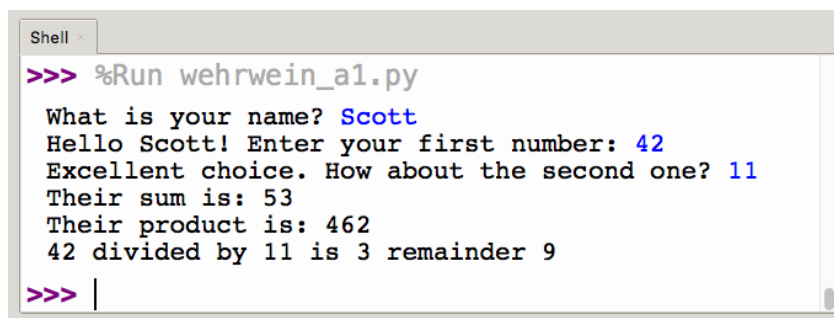
Congratulations! You've just been hired as a Python programmer at an education start-up company. Your first task is to develop a prototype of a program that kindergarten students will use to check their homework assignments which involve addition, multiplication, and division problems.

Program Specification

The program begins with a series of prompts, then prints a few lines to the screen in response. In total there are 6 lines that are printed each time the program is run:

1. Prompt the user for their name
2. Greet the user and ask them to supply the first integer
3. Prompt the user for a second integer
4. Output the sum of the two numbers
5. Output the product of the two numbers
6. Rephrase the division question, and output the whole number and remainder. All numerical outputs on the 6th line of output must be integers (whole numbers, without decimals).

A sample invocation of the program is shown in Figure 1:



```
Shell >
>>> %Run wehrwein_a1.py
What is your name? Scott
Hello Scott! Enter your first number: 42
Excellent choice. How about the second one? 11
Their sum is: 53
Their product is: 462
42 divided by 11 is 3 remainder 9
>>> |
```

Figure 1: Sample Output

Although this is a simple set of steps, there are many, many different Python programs that can achieve it. The text of your prompts does not need to match the example exactly. However, your solution **must** follow the the instructions above exactly as specified. For example:

- Both the greeting and the prompt for the first number must be printed on the second line of output.
- The last (6th) line of output must rephrase the division question and output the whole number and remainder portions of the calculation on a single line.

Valid Input and Error Checking

You should assume that the user provides all requested inputs (via the keyboard) as instructed, and assume that all integers are positive numbers. Your program is *not* required to check the input or behave in any specific way if the above conditions are not met.

Testing Your Program

Testing is a major component in the process of writing software. Often, testing (detecting errors) and debugging (locating and fixing errors) takes way more effort than writing the code did in the first place. We'll talk more about testing as the quarter progresses; in the meantime, the following table provides some helpful test cases that you can use to see if your program is working correctly. Try your code out with the given pairs of integers and see if your output matches the sum, product, and division result.

First Integer	Second Integer	Sum	Product	Division
7	5	12	35	1 remainder 2
5	7	12	35	0 remainder 5
3	3	6	9	1 remainder 0
1	678	679	678	0 remainder 1
8364724	9738	8374462	81455682312	858 remainder 9520

Submission

Double check that your program works according to the specification. Take a look through the rubric below and make sure you won't lose points for reasons that could easily be foreseen and fixed. When you're finished, submit your program to Canvas as a single .py file named `arithmetic.py`. Finally, fill out the A1 Hours quiz with an estimate of the number of hours you spent on A1 (include both the written and programming portions in your estimate).

Rubric

Canvas questions	16 points
Author, date, and program description given in comments at the top of the file	1 point
Program prompts for user's name on the first line	4 points
Greeting on second line includes user's name	4 points
First integer prompt also appears on second line	2 points
Correct sum output on fourth line	2 points
Correct product output on fifth line	2 points
Division question is rephrased, quotient and remainder are printed on sixth line	3 points
Code is commented adequately and variables are appropriately named	2 points
Total	36 points

3 Optional Challenge Problem

Some assignments will come with an optional challenge problem. In general, these problems will be worth very small amounts of extra credit: this one is worth one point. Though the grade payoff is small, you may find them interesting to work on and test your skills in Python and algorithm development. The skills and knowledge needed to solve these problems are not intended to go beyond those needed for the base assignment, but less guidance is provided and more decisions are left up to you. The A1 challenge problem is as follows:

Many online real estate websites have mortgage calculator features¹. These calculators ask for some information, such as the price of a home, the down payment (amount of the home price you'd pay up front), and the interest rate, then calculate the amount you'd have to pay monthly on a loan for the home.

According to NerdWallet², the formula used to calculate the monthly payment based on these inputs is as follows:

$$M = (P - D) \frac{r(1 + r)^N}{(1 + r)^N - 1}$$

Where:

M = The monthly payment

P = The price of the home

D = The down payment amount

N = The number of months over which the loan will be paid off

$r = R/12$, the monthly interest rate, which is the yearly rate divided by 12

Write a program that asks the user to enter P , D , N , and R , then outputs the monthly payment amount M . Notice that you will prompt the user for R , the annual interest rate, but the formula uses r , the monthly interest rate.

3.1 Submission

Upload your submission to Canvas in a file called `challenge.py`.

¹See <https://www.zillow.com/mortgage-calculator/> for an example

²Go to <https://www.nerdwallet.com/mortgages/mortgage-calculator/calculate-mortgage-payment> and click "How to calculate your mortgage payment" for the source of the formula