

COSC 101 Homework 1: Spring 2024

The due date for this homework is **Friday, February 2, 11pm EDT**.

Introduction

This assignment is designed to give you a first introduction to writing programs in Python! By completing this assignment, you will demonstrate that you understand a number of important concepts:

- built-in function calls, such as `print()`, `input()`, and `round()`
- working with different data types, including using type conversion functions and operators
- working with basic arithmetic

Important things to know about homework in COSC101

- **Match your output to the output given.** Precision is important in computer science, and we expect your answers to match the provided example output **exactly** for full credit.
- **Don't change file names.** You're generally given `.py` files to work in. Don't change the filenames of those files.
- **Start early.** You're given a full week to do the homework, and some of the more challenging homework assignments will require that full time.

Your assignment

Your task is to complete following steps:

1. Download the [hw1.zip](#) file from the course website and open it. You will see three python files, `hw1_types.py`, `hw1_brunch.py`, and `hw1_trip.py`, in the unzipped folder. You are expected to write your programs in these files.
2. Complete `hw1_types.py`. This file is used in **Part 1**.
3. Complete `hw1_brunch.py`. This file is used in **Part 2**.
4. Complete `hw1_trip.py`. This file is used in **Part 3**.
5. Review the grading criteria at the end of this assignment.
6. (OPTIONAL) Do the challenge problem for this homework.
7. Submit your completed programs.

Notice that each starter `.py` file has a header with some information for you to fill in. Please do so. Your feedback helps the instructors better understand your experiences doing the homeworks and where we can provide better assistance.

Part 1

Types and Casting

In the `hw1_types.py` file, several variables are already defined. You will sometimes receive starter files that already contain some code. This is an example of what that might look like. This starter code will be denoted by comments and instructions as to whether you should modify the code or not. In this case the code should not be modified. You should write the rest of your code in the area designated by the comments.

1. Write code to `print` the type of each of the existing variables in the same order in which they were assigned.
2. Write code to convert the value of `b` to a float, truncate divide it by `d`, and display the type of the result, *without* modifying the values of any of the variables.
3. Next, write code to convert `d` to a string, concatenate it with `c`, repeat by the integer value of `b`, and save the result to a new variable named `z`.
4. Print the value of the variable `z`.
5. Update the variable `a` to be the remainder when `e` is divided by `d`.
6. Print the new value of the variable `a`.

7. Print the value of the variable `c`.

The full output of your program should be exactly equivalent to the following:

```
<class 'str'>
<class 'str'>
<class 'str'>
<class 'int'>
<class 'int'>
<class 'float'>
4a4a4a
3
a
```

Notes

- Consult the output of the following code snippet `"a" * 4`

Part 2

Brunch Budget

On Sunday, you and your roommates like to go have brunch together. These have become extravagant, so you decided to put your newly acquired programming skills to good use and write a program that figures out the budget to host the brunch. In the `hw1_brunch.py` file, write a short program that asks for some information about the brunch and computes the total cost for food and coffee. The program must ask the user to input the following information:

- number of people invited
- number of bagels slices each guest is expected to eat
- number of cups of coffee each guest is expected to drink
- the cost of a whole bagel
- the cost of a box of coffee

Here are some examples of expected output. Make sure the text of your input prompts and result match these examples exactly.

```
How many friends are invited? 6
How many bagel slices does every friend eat? 4
How many cups of coffee does every friend drink? 2
How much does a bagel cost? 4.99
How much does a box of coffee cost? 19.99
The brunch will cost you $79.87
```

```
How many friends are invited? 13
How many bagel slices does every friend eat? 3
How many cups of coffee does every friend drink? 2
How much does a bagel cost? 2.99
How much does a box of coffee cost? 23.99
The brunch will cost you $107.78
```

Notes

- The key challenge for this problem is to convert each input to the correct data type before assigning it to its corresponding variable.
- Another challenge is computing the number of bagel slices and boxes of coffee correctly. You may use the `round` function to round to the nearest whole number.
- Notice that the `hw1_brunch.py` file includes variables representing important conversions. **DO NOT** change these.
- Notice that the total cost is always printed with two decimal places.

To get full credit on this problem you need to get those details correct.

Part 3

Road Trip

The semester has just started. However, you and your friends are already tired of the cold, brrr. You'd like to know how much time it takes to drive to a warmer place. In the `hw1_trip.py` file, write a program that asks for a driver's name, the distance (in miles) and the speed (in miles per hour) and reports how long it will take for the driver to arrive at the destination. The time should be reported in units of days, hours, minutes, and seconds, rounded to the nearest second. The printed numbers should be of type `int`, however, you may need to work with floats during the computation.

Your program's output should look like the examples provided below.

For example, San Francisco is roughly 2850 miles from Colgate. The snow makes you a timid driver, so your average speed will be 55.00 mph. Your program's output should look like this:

```
Who will be driving? George
How far away is the destination (in miles)? 2850
How fast will George be driving (in mph)? 55
It will take George 2 days 3 hours 49 minutes and 5 seconds.
```

Another example: Luiz is willing to drive you all to New Orleans. The distance between Colgate and New Orleans is roughly 1385 miles and Luiz will be driving at 77 mph on average. Your program's output should look like this:

```
Who will be driving? Luiz
How far away is the destination (in miles)? 1385
How fast will Luiz be driving (in mph)? 77
It will take Luiz 0 days 17 hours 59 minutes and 13 seconds.
```

Yet another example: Aditi will be driving you all to Charlotte. The distance between Colgate and Charlotte is roughly 704 miles, and Aditi will be driving at 60 mph on average. Your program's output should look like this:

```
Who will be driving? Aditi
How far away is the destination (in miles)? 704
How fast will Aditi be driving (in mph)? 60
It will take Aditi 0 days 11 hours 44 minutes and 0 seconds.
```

Notes

- The driver's name is used in the subsequent prompt about driving speed.
- One approach to solving this problem is to first figure out how long the trip will take in seconds. For example, to go 2000 miles at 100 mph will take 72,000 seconds! Given the total amount of time in seconds, break that number down into days, hours, minutes, and remaining seconds.
- To round to the nearest whole number, use the `round` function. Although the `round` function does round off the fractional part, it returns a number of type `float` whereas you probably want an `int`. Remember that you can use the `int` function to get an `int` from a `float`.

To get full credit on this problem you need to get those details correct.

Submission Instructions

Submit four Python files to the platform indicated in your class section:

- `hw1_types.py`
- `hw1_brunch.py`
- `hw1_trip.py`

Remember to complete the questions at the top of each file before submitting.

Grading

Your assignment will be graded on two criteria:

1. Correctness: this document contains several examples for each program. Be sure that you run your program once for each example and make sure it works correctly for each one! [90%]

The correctness part of your grade is broken down as follows:

Category	Portion of grade
Types	30%
Brunch	30%
Trip	30%

2. Program design and style [10%]: style and program design become increasingly important the more complex your program becomes. For these first programs, adhere to the following guidelines:

- Variable names should be meaningful
- Programs should contain at least a few descriptive comments. Each of the programs for this homework involves multiple conceptual steps and should have comments that indicate to a reader what's going on at a high level.

Challenge Problem (OPTIONAL)

Challenge problems are entirely optional extensions to the homework. If you complete them successfully, you are rewarded with a sense of accomplishment and a small number of extra points on the homework. Note that any extra points apply only to the homework on which the challenge problem is attempted, not to other homeworks or other grading categories. They are intended for students who want to explore a little further; only pursue the challenge problem after you have successfully completed the homework.

For this challenge problem, you should write a program that prompts the user for a floating point number which stands for gallons of gasoline. You will reprint that value along with other information about gasoline and gasoline usage. In particular, you should print:

- The equivalent number of liters
- The equivalent number of barrels of oil required
- The number of pounds of CO_2 produced

Some relevant conversion factors are:

- 1 gallon is equivalent to 3.7854 liters
- 1 barrel of oil produces 19.5 gallons of gasoline (approximately, can vary depending on the oil source).
- 1 gallon of gasoline produces approximately 20 pounds of CO_2 .

If you decide to do the challenge problem, please create a new file `hw1_challenge.py` and write your code inside that file.

An example run of your program might be:

```
Please enter the number of gallons of gasoline: 100
Original number of gallons is: 100.0
100.0 gallons is equivalent to 378.54 liters
100.0 gallons of gasoline requires 5.12820512821 barrels of oil
100.0 gallons of gasoline produces 2000.0 pounds of CO2
```