

## Zelle 3e Chapter 3 Coding Assignment

### General Instructions

My expectations for your work on coding assignment exercises will grow as we progress through the course. In addition to applying any new programming techniques that have been covered in the current chapter, I will be expecting you to follow all of the good programming practices that we have adopted in the preceding weeks. Here is a quick summary of good practices that we have covered so far:

- Include a single-line comment with name of program file.
- Include a single-line comment that describes the intent of the program.
- Place your highest-level code in a function named `main`.
- Include a final line of code in the program that executes the `main` function.
- Follow all PEP-8 Python coding style guidelines enforced by the PyCharm Editor. For example, place two blank lines between the code making up a function and the code surrounding that function.
- Choose names for your variables that are properly descriptive.
- Model your solution after the code that I demonstrate in the tutorial videos.
- Remember to test your program thoroughly before submitting your work.

### Exercise 1

Create a program named *integer\_division\_2*. This program will Calculate both the quotient and remainder for two integers provided by the user. This program will use Python integer division operator (`//`) and the Python modulo operator (`%`). Please use the program demonstrated in the tutorial video (*integer\_division\_1*) as a guide for your code.

For a refresher on the terms used in division, please consult <http://www.math-only-math.com/terms-used-in-division.html> .

When this program is run, console sessions should look like this:

```
Please enter an integer for the dividend: 33
Please enter an integer for the divisor: 22
The quotient is 1 , the remainder is 11
```

## Exercise 2

Create a program named *remainder\_accumulator*. This program will use the Python modulo operator (%) to calculate the remainders on each of a series of five pairs of user-supplied integers. The program will also accumulate and print the sum of all five remainders. Please use the program demonstrated in the tutorial video (*quotient\_accumulator*) as a guide for your code.

For a refresher on the terms used in division, please consult <http://www.math-only-math.com/terms-used-in-division.html>.

When this program is run, console sessions should look like this:

```
Now collecting integer pair 1
Please enter an integer for the dividend: 33
Please enter an integer for the divisor: 22
This remainder is 11
Now collecting integer pair 2
Please enter an integer for the dividend: 44
Please enter an integer for the divisor: 33
This remainder is 11
Now collecting integer pair 3
Please enter an integer for the dividend: 55
Please enter an integer for the divisor: 44
This remainder is 11
Now collecting integer pair 4
Please enter an integer for the dividend: 66
Please enter an integer for the divisor: 55
This remainder is 11
Now collecting integer pair 5
Please enter an integer for the dividend: 77
Please enter an integer for the divisor: 66
This remainder is 11
The accumulated sum of the remainders is 55
```

### Exercise 3

Create a program named *right\_int\_mixed\_type\_divide\_and\_accumulate*. This program will use the Python division operator (/) to calculate the quotients on each of a series of five pairs of user-supplied values. Each set of user-supplied values will include a float value (the first, or left, entry) and an int value (the second, or right, entry). The program will also accumulate and print the sum of all five quotients. Please use the program demonstrated in the tutorial video (*left\_int\_mixed\_type\_divide\_and\_accumulate*) as a guide for your code.

For a refresher on the terms used in division, please consult <http://www.math-only-math.com/terms-used-in-division.html> .

When this program is run, console sessions should look like this:

```
Now collecting mixed-type pair 1
Please enter a float for the dividend: 3.0
Please enter an integer for the divisor: 2
This quotient is 1.5
Now collecting mixed-type pair 2
Please enter a float for the dividend: 4
Please enter an integer for the divisor: 3
This quotient is 1.3333333333333333
Now collecting mixed-type pair 3
Please enter a float for the dividend: 5.00
Please enter an integer for the divisor: 4
This quotient is 1.25
Now collecting mixed-type pair 4
Please enter a float for the dividend: 6.000
Please enter an integer for the divisor: 5
This quotient is 1.2
Now collecting mixed-type pair 5
Please enter a float for the dividend: 7
Please enter an integer for the divisor: 6
This quotient is 1.1666666666666667
The accumulated sum of the quotients is 6.45
```

## Tools

Use PyCharm to create and test all Python programs.

## Submission Method

Follow the process that I demonstrated in the tutorial video on submitting your work. This involves:

- Locating the properly named directory associated with your project in the file system.
- Compressing that directory into a single .ZIP file using a utility program.
- Submitting the properly named zip file to the submission activity for this assignment.

## File and Directory Naming

Please name your Python program files as instructed in each exercise. Please use the following naming scheme for naming your PyCharm project:

```
surname_givenname_exercises_zelle_3e_chapter_03
```

If this were my own project, I would name my PyCharm project as follows:

```
trainor_kevin_exercises_zelle_3e_chapter_03
```

Use a zip utility to create one zip file that contain the PyCharm project directory. The zip file should be named according to the following scheme:

```
surname_givenname_exercises_zelle_3e_chapter_03.zip
```

If this were my own project, I would name the zip file as follows:

```
trainor_kevin_exercises_zelle_3e_chapter_03.zip
```

## Due By

Please submit this assignment by the date and time shown in the Weekly Schedule.

Last Revised

2021-09-08