

Zelle 3e Chapter 6 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.
- Output printed by the program (both prompts and results) should be polite and descriptive.
- Choose names for your variables that are properly descriptive.
- Choose names for your functions that are properly descriptive.
- Close all files before the conclusion of the program.
- 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 *miles_to_kilometers*. Prompt the user to enter a measurement in miles at the console (float). Use a separate function to convert the measurement from miles to kilometers. The conversion function should do only the conversion calculation. It should not print the result. At the conclusion, print the result rounded to 1 decimal place. Use my tutorial example as a model for your code.

Use the following factor to convert miles to kilometers: 1.60934

When running your test, you should expect input/output similar to the following on your console:

```
Please enter the measurement in miles: 2.0
The measurement in kilometers is 3.2
```

Exercise 2

Create a program named *function_cannot_change_float_parameter*. This program should demonstrate that an immutable value object passed to a called function as a parameter CANNOT be changed within the context of the calling code. Use a float value for your demonstration. Use my tutorial example as a model for your code. When running your test, you should expect the following output on your console:

```
Starting value in main is 25.99

Starting value in function is 25.99
Ending value in function is 111.22

Ending value in main is 25.99
```

Exercise 3

Create a program named *function_can_change_list_parameter*. This program should demonstrate that a mutable value object passed to a called function as a parameter CAN be changed within the context of the calling code. Use a list of integers as the value object for your demonstration. In the called function, replace each integer in the list with its square. To calculate the square, you may use any available method, including multiplication ($x * x$), standard Python exponentiation ($x**2$), or exponentiation using the Python math library ($\text{math.pow}(x, 2)$). Use my tutorial example as a model for your code. When running your test, you should expect the following output on your console:

```
Starting value of list in main is [9, 7, 5, 3, 1]

Starting value of list in function is [9, 7, 5, 3, 1]
Ending value of list in function is [81, 49, 25, 9, 1]

Ending value of list in main is [81, 49, 25, 9, 1]
```

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_givename_exercises_zelle_3e_chapter_06
```

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

```
trainor_kevin_exercises_zelle_3e_chapter_06
```

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_givename_exercises_zelle_3e_chapter_06.zip
```

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

```
trainor_kevin_exercises_zelle_3e_chapter_06.zip
```

Due By

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

Last Revised

2021-02-14