CS-119 Chapter 2 Lab

Student Learning Outcomes

- Gain more experience with the Python IDLE IDE
- Python applications
- Writing pseudocode
- Variables
- Comments
- Arithmetic calculations
- Sequential code
- Console input and output

Overview

This lab continues the introduction to programming in Python and writing *sequential code*. You will be creating 3 Python console applications using Python IDLE. It also gives you an opportunity to get a little more hands-on experience with the terminology and concepts presented in chapters 1 & 2 of the PLD text.

As we progress through these labs, the detailed step-by-step instructions will be omitted for things already covered and more focused on any new material being covered. Before you start this lab, be sure you are comfortable with the following:

- Starting Python IDLE
- Creating a pseudocode.txt file for your pseudocode.
- Creating Python code files
- Getting user input using the input() method
- Displaying output using the print() method

Review Lab 1 if you're unclear on any of the above.

Exercise 1: Text Message Cost Calculator

This exercise is based on exercise 6, page 61 of the PLD text. Besides pseudocode, you will write working Python code as well.

Develop an application that will allow the user to enter the number of text messages sent in the last month. Messages are 0.25 each and a 9% tax is added to the total.

Steps

1. Start Python IDLE.

2. Create your pseudocode.txt file. For all labs, you may put your pseudocode for all the exercises in a single file to save time and make things easier. Be sure to have comments or notes indicating which pseudocode is for which exercise. It will help you score some points!

3. Write your pseudocode. Once again, keep in mind there is no single "correct" way of doing it. What are the inputs? What processing and formulas are needed? What are the outputs?

4. Create a new file and save as TextMessage.py to your Lab 1 folder. Note that Python source files must have the .py extension.

5. Be sure to save your work!

6. Run your application. Select Run Module from the Run menu.

7. Choose test values where you can easily verify that your program is producing the correct output.

Exercise 2: Gas Mileage Calculator

In this exercise, you will create another Python application to compute gas mileage and gas cost per mile. Your application ***must*** allow the user to enter the number of miles driven, number of gallons put into the gas tank and cost in dollars per gallon of gas. Your outputs must include MPG, total fuel cost and cost per mile.

Key formulas:
mpg = milesDriven / gallons
totalFuelCost = gallons * costPerGallon
costPerMile = totalFuelCost / milesDriven

Before writing any code, write some pseudocode as it will help you break down the problem into understandable pieces. Also, it's worth points! To save time, you may add your pseudocode for this exercise to the pseudocode file you created in exercise 1.

You will create the Python file the same way you did in exercise 1.

Grading Criteria:

Deliverable	Points	Breakdown
Text message cost pseudocode	**10**	**Input, process and output clearly identified. Clear programming logic and calculation formulas.**
Text message cost program code	**10**	**Complete, code is clear, descriptive variable names, appropriate use of comments**
Text message cost run	**5**	**Runs, produces correct output.**
MPG pseudocode	**5**	**Input, process and output clearly identified. Clear programming logic and calculation formulas.**
MPG program code	**15**	**Complete, prompts for all 3 inputs are clear, code is clear, descriptive variable names, appropriate use of comments**
MPG run	**5**	**Runs, produces correct output, and displays mpg, total fuel cost and cost per mile.**
Lab Total	**50**	