CS 160 – Exploring Computer Science

Assignment 1: Input and Output

Academic Integrity

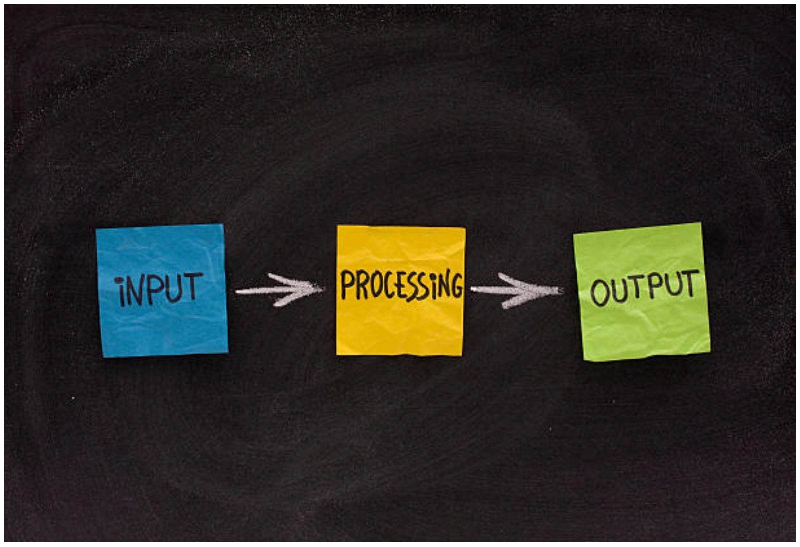
You may NOT, under any circumstances, begin a programming assignment by looking for completed code on StackOverflow or Chegg or any such website, which you can claim as your own. Please check out the Student Code of Conduct at PCC.

The only way to learn to code is to do it yourself. The assignments will be built from examples during the lectures, so ask for clarification during class if something seems confusing. If you start with code from another source and just change the variable names or other content to make it look original, you will receive a zero on the assignment.

I may ask you to explain your assignment verbally. If you cannot satisfactorily explain what your code does, and answer questions about why you wrote it in a particular way, then you should also expect a zero.

I understand that acts of academic dishonesty may be penalized to the full extent allowed by the Portland Community College Student Conduct Code, including receiving a failing grade for this assignment. I recognize that I am responsible for understanding the provisions of the PCC Student Conduct Code as they relate to this Course.

Your name here.



Purpose

For this assignment, you will create your first program using the Python programming language to demonstrate your understanding of some of the programming concepts we covered this week. It is also an opportunity to show off your creativity because you get to decide what you will do! Remember, all students are at different levels in the class - don't compare yourself with other students - be proud of your creation!

After completing this assignment you will be able to:

- Design an algorithm for your program.
- Write a simple python program with basic user input and output.
- Do some basic calculations in Python.

Task

- ❏ Before you get started:
    - ❏ Look at this Assignment 01 Sample for an example and tutorial video.
- ❏ Open the Algorithmic Design Document, make a copy, and follow the steps to create your algorithm.
- ❏ You must express your algorithm as **pseudocode.**
- ❏ Print a welcome message for your program.
- ❏ Write an interactive application in Python to get input from the user, do some calculation, and output the results to the user.
- ❏ Here are some ideas to get you started:
    - ❏ Any measurement conversions - cups to ounces, meters to kms, or grams to kg etc.
    - ❏ Any calculation involving money - tax rates, loans, interest rates, tuition increases etc.
    - ❏ Any calculation involving speed, distance, and time.
- ❏ Algorithmic Design Document Requirements:
    - ❏ Make a copy of the Algorithmic Design Document.
    - ❏ Start with a sample run to help with the design process.
    - ❏ Plan the logic steps for your algorithm. **You must answer the following questions with numbered steps and clear instructions to solve the problem.**
        - ❏ Identify all of the user input. What are the data types of the inputs? Define the input variables.
        - ❏ Describe the program output. What is displayed to the user? What are the data types of the output? Define the output variables.
        - ❏ What calculations do you need to do to transform inputs into outputs? List all formulas needed, if applicable. If there are no calculations needed, state there are no calculations for this algorithm.
        - ❏ Describe the process needed to transform inputs to outputs *using numbered steps*. Here is where you would use conditionals, loops, functions or array constructs (if applicable) and explain the process in transforming inputs into outputs.

        - ❏ Do not use any code or syntax here, just plain English sentences.
- ❏ Check the Criteria for Success section to make sure you have completed all requirements and to see how to submit your assignment.
- ❏ Check the Additional Support section for more resources to help you complete the assignment.

Criteria for Success

- ☐ Look at the Assignment Rubric. You will be graded on the following coding constructs:
    - ☐ variables
    - ☐ input/output
    - ☐ mathematical computation
    - ☐ main() function
- ☐ Include the required comment header (see style guide) and any explanatory comments in your code (always explain any mathematical calculations).
- ☐ Run your program and test with different user input. Do your prompts explain what the user should enter?
- ☐ Take a look at your user interface (UI) and user experience (UX). Did you provide a welcome message letting the user know what to expect? Is the UI nicely spaced with no typos?
- ☐ Your program must use a main() function and include the correct program comment header.
- ☐ Download your Algorithmic Design Document as a PDF (File -> Download -> PDF), rename it to `a01.pdf`, and upload it to the D2L assignment folder.
- ☐ Upload your `a01.py` Python source file to the D2L assignment by **Sunday**.
- ☐ Do your own work. Consult the syllabus for more information about academic integrity.

Additional Support

- ❏ Post a question for the instructor in the Ask Questions! area of the Course Lobby.
- ❏ Look at this Assignment 01 Sample for an example and tutorial video.
- ❏ Runestone Academy Resources:
    - ❏ Start with the video in Section 2.1. Variables, Expressions and Statements from Chapter 2, Simple Python Data in the Runestone Academy textbook.
    - ❏ Read through Sections 2.2, Values and Data Types from the textbook. This section talks about the different Data Types (Numeric Data and Strings). To check your understanding, make sure to try some programs in the workspace provided.
    - ❏ Read through Sections 2.4 and 2.5 from Chapter 2, Simple Python Data from this textbook. Try some programs in the workspace provided in Section 2.4.
    - ❏ Read through Section 2.10 that talks about the assignment operator and how reassignment of variables work. These are all small sections but will really help you understand how the syntax works.
    - ❏ Read through Sections 2.6 and 2.7 from Chapter 2, Simple Python Data from this textbook. Try some programs in the workspace provided.

- ❏ Section 2.6 talks about Python Statements and Expressions.
- ❏ Section 2.7 talks about operators, order of operations, etc.
- ❏ Now is also a good time to read Section 2.3 to understand Type Conversion Functions.
- ❏ Read through Section 2.7 from Chapter 2, Simple Python Data from this textbook. This section talks about the different operators including integer division and the modulus operators. Try some programs in the workspace provided in Section 2.7.
- ❏ Read through Section 2.8 from Chapter 2, Simple Python Data from this textbook. This section has a video and some really good examples for user input. Try some programs in the workspace provided.