

# Portfolio Assignment 1

ENDG 233 Fall 2021

Department of Electrical and Software Engineering

Schulich School of Engineering

Due: October 1, 2021 at 11:59 PM

The purpose of these assignments is to build a portfolio of successful programs that could be used to demonstrate your understanding of coding and Python. This assignment is worth 10% of your final grade.

## Portfolio Assignment #1 – Learning Outcomes

- Perform operations on dynamically-typed variables
- Accept and analyze user input
- Design control flow logic using data comparison
- Print formatted output according to given specifications

## Program Specifications

Design a terminal-based calculator application that meets the following design specifications:

- Prompt the user to enter a mathematical expression that includes three integer values and two operators. You may prompt for the input to be entered in any form or order you like, but be sure to give the user clear instructions.
- Your program must correctly calculate and print the answer to the terminal.
- The final calculation must show the original expression and the final answer. You must use the output formatting shown in the attached output example.
- Your application must handle the order of operations correctly! (i.e. \* and / should be calculated before + and - )
- Your answer should remain in integer format. Any division operations should use floor division.
- You may assume that only valid integers and operators will be given as input. You do not need to test for their validity.
- A screenshot with example terminal behaviour is included in this handout. You do not need to follow this format exactly when prompting for user input as long as all of the specifications are met. You must, however, follow the format exactly for displaying the expression and final result.
- Your code should include at least two separate if/else logic blocks (may be a single if statement, if/else, or if/elif/else/etc.).
- Your code must follow the conventions discussed so far in the course (names\_with\_underscores, four spaces for indentations, spaces between variables/operators, comments throughout, etc.)
- You may not use any built-in Python functions or external modules.
- Your code will be run by the TAs as your end user using VS Code and a minimum of Python 3.9.
- FAQs about the assignment will be answered on the D2L discussion boards. Please check the boards for any clarifications before submitting.
- The grading rubric is also included in this handout.

## Assignment Tasks

- Make sure to watch the video lessons for Weeks 1 – 3 and review the corresponding active learning content.
- Create a flowchart to plan your logic. You may use a digital drawing program or draw the diagram neatly by hand. Illegible diagrams will not be marked.
  - Popular free tools include draw.io (diagrams.net) and Lucidchart (free for students)
- Open VSCode and start a new terminal. Make sure that your virtual environment is activated.
- `calculator.py` is provided as a starting point. Fill in the header with your own information and write your program in this file.
- Remember to test your program execution via the terminal.
  - For example, use `python calculator.py` or the corresponding command for your operating system.
- Take a screenshot of your successful program execution that shows:
  - a) Your virtual environment is active when running the program
  - b) Your program successfully computes the given tests (see provided output example)
- Submit the following items to the Assignment 1 D2L dropbox:
  - Your final `calculator.py` file with your name in the comment header (do not change the file name)
  - Your flowchart as a PDF or PNG (e.g. `00121343\_assign1\_flowchart.pdf`)
  - Your execution screenshot (e.g. `00121343\_assign1\_output.png`)

### Assignment 1 Rubric (28 marks, 10% of overall grade)

Your code must successfully run to be given full marks. Code that does not execute may be given partial marks for some criterion listed below.

#### Flowchart and Planning (8 marks):

- (1) All flowchart symbols follow the correct conventions
- (2) Includes input of 3 integers and 2 operators
- (2) Includes output of the entered expression and the final result
- (1) Includes logic check for the order of operations
- (2) Includes branches for each operator option
- One mark will be deducted for each error or missing component, up to a maximum of 8 marks

#### Commenting and Syntax (5 marks):

- (1) Your name must be included in the file header
- (2) Comments must be included throughout the code to explain the functionality
- (1) All variables have clear and useful names that use lowercase words separated by an underscore
- (1) Code is clearly indented and spaces are included between variables and operators
- One mark will be deducted for each error or missing component, up to a maximum of 5 marks

#### Code Structure and Semantics (4 marks):

- (2) Solution receives both int input and string input
- (2) Solution contains at least two types of if/else branching (may be a single if statement, if/else, or if/elif/else/etc.).
- One mark will be deducted for each error or missing component, up to a maximum of 4 marks

#### User Interface and Functionality (5 marks):

- (1) User is given clear guidance on how to enter in the input values
- (1) Program accepts three integer values and two operators
- (1) Program includes logic for determining the correct order of operations
- (2) The program output includes the original input expression and the calculated answer while following the required format
- One mark will be deducted for each error or missing component, up to a maximum of 5 marks

#### Execution (6 marks):

- (3) Screenshot of successful execution is shown using the following parameters:
  - Your screenshot should include all successful functionality using the given demonstration values
  - Your screenshot must show that your virtual environment is active and that you are able to run your program directly from the terminal
  - Demonstration 1:  $10 + 2 / 5 = 10$
  - Demonstration 2:  $6 * 9 - 2 = 52$
- (3) Your program will be executed three times with varied integer/operator values to test if the output is calculated correctly
  - All students will have their code tested with the same input values (e.g.  $4 + 15 * 2$ )
- One mark will be deducted for each error or missing component, up to a maximum of 6 marks

## Assignment 1 Example Output

You may choose how to prompt for input as long as it is clear to the user.

You must display the expression and final answer using the following format and spacing:

Entered expression:  $10 + 2 / 5$

Your final answer = 10

```
(venv) C:\Users\eamarasc\Dropbox\Teaching\ENDG 233\Fall 2021\Portfolio\Assignment 1>python calculator.py
Enter the first value: 10
Enter the first operator: +
Enter the second value: 2
Enter the second operator: /
Enter the third value: 5
Entered expression: 10 + 2 / 5
Your final answer = 10

(venv) C:\Users\eamarasc\Dropbox\Teaching\ENDG 233\Fall 2021\Portfolio\Assignment 1>python calculator.py
Enter the first value: 6
Enter the first operator: *
Enter the second value: 9
Enter the second operator: -
Enter the third value: 2
Entered expression: 6 * 9 - 2
Your final answer = 52

(venv) C:\Users\eamarasc\Dropbox\Teaching\ENDG 233\Fall 2021\Portfolio\Assignment 1>python calculator.py
Enter the first value: 4
Enter the first operator: +
Enter the second value: 15
Enter the second operator: *
Enter the third value: 2
Entered expression: 4 + 15 * 2
Your final answer = 34
```