

# Variables, Input and Validation

## Description:

In this project, you will create a simple python that takes input from the user and performs some kind of interaction with a user. You will use flowcharts to plan out the program's logic and identify its purpose. You will also comment and document your code to make it more readable and understandable.

## Objective:

Design a program with the basic commands of the programming language Python.

## Essential Questions:

- What is the purpose of the program?
- How can we use flowcharts to plan out the program's logic and identify its purpose?
- How can we use variables to store the user's input and the results of the arithmetic operations?
- How can we use input to get the user's input and validate it to ensure that it is a number?
- How can we validate the user's input to ensure that it is a valid arithmetic operation?
- How can we use comments and documentation to make the code more readable and understandable?
- How can we apply the iterative steps of Define, Prepare, Prototype, and Test when creating the program?
- How can we handle errors and exceptions in the program?
- How can we test the program to ensure that it works correctly?

## Activities:

- Introduce the project and its objective.
- Brainstorm a project Idea.
- Discuss the purpose of the program and how it will work.
- Introduce flowcharts and how they can be used to plan out the program's logic.
- Create a flowchart for the simple program.
- Discuss and identify the variables needed for the program and how to validate the user's input.
- Have students document their flowchart and variable identification process.
- Review the documentation and provide feedback and revise documents.
- Program using python.
- Review the final documentation, program and get feedback, finalize.

## Reflection:

- What was the purpose of the program, and how did you identify it?
- How did you use flowcharts to plan out the program's logic, and what did you learn from this process?
- What variables did you use in the program, and how did you validate the user's input?
- How did you comment and document your code to make it more readable and understandable?
- What challenges did you encounter while working on the project, and how did you solve them?

- What resources did you use while working on the project, and which ones were especially helpful?
- How did you apply the iterative steps of Define, Prepare, Prototype, and Test when creating the program?
- How did you handle errors and exceptions in the program, and what did you learn from this process?
- How did you test the program to ensure that it works correctly, and what improvements could you make to create a more robust calculator?
- What did you learn about Python programming, variables, input, and validation from this project, and how will you apply this knowledge in future projects?

**Assessed:**

- CS.1.1: Program Design (Level 10)
- CS.2.1: Data Structures (Level 8)