CIS 103: Fundamentals of Programming

Lab 2: Building a Simple Calculator

Due Date: 09/14/2024 @ 11:59pm

Objective:

- Introduce students to the concepts of functions, user input, and control structures in Python.
- Guide students through the creation of a basic calculator that can perform arithmetic operations.

Lab Instructions:

In-Lab Portion: Building the Calculator

Estimated Time:

- 1 hour

Resources Required:

- Python IDE (e.g., Visual Studio Code, PyCharm, or any text editor)
- Access to a computer with Python installed

Instructions:

Create a New Python File:

- Open your Python IDE or text editor.
- Create a new Python file named `calculator.py`.

Define the Calculator Functions:

- Define four functions for the basic arithmetic operations: addition, subtraction, multiplication, and division.

Create a User Interface:

- Use the `input` function to prompt the user to select an operation and input numbers.
- Implement a loop that allows the user to perform multiple calculations until they choose to exit.

Test the Calculator:

- Run the program and test all operations to ensure they work as expected.
- Handle edge cases like division by zero.

Submission:

- Save the file as `calculator.py`
- Upload the file to your GitHub account.

Take-Home Portion: Enhancing the Calculator

Objective:

- Extend the functionality of the calculator by adding additional features and improving the user interface.

Estimated Time:

- 1-2 hours

Instructions:

1. **Add Additional Operations:**
    - Extend the calculator to include additional operations such as:
        - Exponentiation (`x^y`)
        - Modulus (`x % y`)
        - Square root (`√x`)
    - Update the user interface to allow the user to select these new operations.
2. **Implement Error Handling:**
    - Improve the calculator by adding error handling for invalid inputs (e.g., non-numeric input, invalid operation selection).
    - Ensure that the program does not crash and provides appropriate feedback to the user.
3. **Enhance User Interface:**
    - Refine the user interface by making it more user-friendly. For example:
        - Add clear instructions for the user.
        - Format the output more neatly.
        - Provide an option to clear the screen between operations.
4. **Document the Code:**
    - Add comments to your code to explain how each part of the program works.
    - Write a brief summary at the top of the file explaining what the program does and how to use it.
5. **Submission:**
    - Save the enhanced calculator as `enhanced_calculator.py`.
    - Upload the file to your GitHub account.

Final Submission:

- Compile both the in-lab and take-home portions into a single Word document or PDF named `CIS103_Lab2_LastName`
- Include:
    - The full Python code from the in-lab portion.
    - The full Python code from the take-home portion.
    - A short reflection (200-300 words) on what you learned from this lab, the challenges you faced, and how you overcame them.
- Submit the document on Brightspace along with your GitHub repository link.