**📝 Project Report**

**Title: Simple Calculator in Python**

**Task Number: Task 2**

### 🔖 ****Objective:****

To create a Python-based command-line calculator that performs a variety of arithmetic operations including addition, subtraction, multiplication, division, modulus, power, and floor division, while handling user input and errors gracefully.

### 🧰 ****Technology Used:****

* **Language:** Python 3.x
* **IDE:** Visual Studio Code
* **Execution:** Terminal (VS Code / Command Prompt)

**⚙️ Features Implemented:**

1. Supports the following operations:
   * ➕ Addition (+)
   * ➖ Subtraction (−)
   * ✖️ Multiplication (×)
   * ➗ Division (÷)
   * 🧮 Modulus (%)
   * 🔼 Power (\*\*)
   * ⬇️ Floor Division (//)
2. Takes two numeric inputs from the user.
3. Displays results with proper formatting.
4. Validates input for non-numeric values.
5. Handles division by zero with error messages.
6. Allows user to perform multiple calculations using a loop.
7. Gracefully exits on user request.

**🔐 Sample Validation:**

* If the user enters text instead of numbers → Error message is shown.
* If division or modulus is attempted with 0 → Division by zero is prevented.

**💡 Working Logic:**

1. The program starts with a welcome message and shows the list of operations.
2. It prompts the user to enter two numbers.
3. Then the user is asked to enter the desired arithmetic operation.
4. The calculator performs the selected operation and displays the result.
5. The program then asks if the user wants to perform another calculation or exit.
6. The process repeats until the user chooses to exit.

**📈 Enhancements Made:**

Compared to a basic calculator, this version includes:

* **7 different operations**
* **Looping logic** to avoid restarting
* **Exception handling** for robustness
* **Clear user guidance and formatting**
* **Graceful exit**

### ✅ ****Conclusion:****

This Python calculator project demonstrates control flow, user input, exception handling, and basic operations in a real-world interactive script. It is functional, user-friendly, and extensible for more features such as trigonometric functions or a graphical interface.