# Mini Project: Basic Calculator

## Project Description:

Create a basic calculator program that performs simple arithmetic operations based on user input. The calculator should be able to handle addition, subtraction, multiplication, and division. Additionally, the program should include error handling for invalid operations and division by zero.

## Objectives:

- Use input and output functions to interact with the user.
- Apply arithmetic operators to perform calculations.
- Utilize variables to store user input and results.
- Implement if-else-elif statements to control the flow of the program.
- Include error handling for specific cases (e.g., invalid input or division by zero).

## Instructions:

1. **User Input:**
    - Prompt the user to enter two numbers.
    - Prompt the user to choose an operation: addition (+), subtraction (-), multiplication (*), or division (/).
2. **Process User Input:**
    - Perform the chosen arithmetic operation on the two numbers.
3. **Output Result:**
    - Display the result of the calculation to the user.
4. **Error Handling:**
    - If the user chooses an invalid operation, display an error message.
    - Handle division by zero by displaying an appropriate message instead of crashing the program.
5. **Example Output:**

```
Enter the first number: 10
Enter the second number: 5
Choose an operation (+, -, *, /): +
Result: 10 + 5 = 15
```

6. **Additional Features (Optional):**
    - Allow the user to perform multiple calculations without restarting the program.
    - Add a feature to exit the program gracefully when the user is done.

## Implementation Hints:

- Use `input()` to gather user input.
- Store the input values in variables and convert them to the appropriate data types as needed.
- Use `if-elif-else` statements to determine the operation and calculate the result.
- Use `try-except` blocks to handle division by zero.

## Assessment Criteria:

- Correctness: The program performs calculations accurately based on user input.
- Error Handling: The program gracefully handles invalid input and division by zero.
- User Interaction: The program effectively communicates with the user through prompts and messages.
- Code Structure: The code is well-organized, with clear variable names and logical flow.