

01_Calculator

1. What is the primary purpose of this project?

- **Answer:** The primary purpose of this project is to create a basic calculator that allows users to perform mathematical operations like addition, subtraction, multiplication, and division. It also teaches how to handle user input, implement menus, and use control structures like loops and switch cases in C++.
-

2. Why is the `displayMenu()` function used separately?

- **Answer:** The `displayMenu()` function is used separately to make the code modular and reusable. By keeping the menu logic in a separate function, the main code becomes more organized and easier to understand.
-

3. How does the program handle invalid inputs?

- **Answer:** The program checks if the user's choice is outside the valid range (1-5). If an invalid choice is entered, it displays an error message (`Invalid choice. Please try again.`) and uses the `continue` statement to restart the loop.
-

4. How does the program prevent division by zero?

- **Answer:** Before performing division, the program checks if the second number (`num2`) is zero. If it is zero, it displays an error message (`Error: Division by zero is not allowed.`) and skips the division.
-

5. Why is a `while (true)` loop used?

- **Answer:** The `while (true)` loop ensures that the calculator keeps running until the user explicitly chooses to exit (by entering choice 5). This provides continuous functionality without restarting the program.
-

6. How does the program terminate gracefully?

- **Answer:** If the user enters choice `5`, the program displays a goodbye message (`Exiting the calculator. Goodbye!`) and uses the `break` statement to exit the loop, terminating the program.
-

7. What happens if the user enters a character instead of a number?

- **Answer:** If the user enters a character or invalid input instead of a number, the program might behave unexpectedly since there is no explicit input validation for non-numeric inputs. Adding input validation can handle such cases.
-

8. What is the role of the `switch` statement in this program?

- **Answer:** The `switch` statement is used to execute different operations (addition, subtraction, multiplication, division) based on the user's choice. It makes the code more structured and easy to extend for additional operations in the future.
-

9. Why does the program ask for two numbers after every valid choice?

- **Answer:** For every operation, the calculator requires two inputs (`num1` and `num2`) to perform the selected operation. This ensures that the user provides the necessary data for the calculation.
-

10. How can this project be extended in the future?

- **Answer:**
 - Adding more operations like modulus (`%`), square root, or power.
 - Implementing input validation to handle non-numeric inputs.
 - Creating a graphical user interface (GUI) to make the calculator more user-friendly.
 - Adding memory functionality to store and recall previous results.
-