



LARANA, INC.

INTERNSHIP PROJECT

CONTENT

01

INTRODUCTION

02

INSTRUCTION

03

PROBLEMS

04

SOLUTION OVERVIEW

05

CHALLENGES AND RESOLUTION

INSTRUCTIONS

- Setting Up Your Environment
- Install Required Software:
- C++ Compiler: Install GCC or any other C++ compiler.
- IDE/Text Editor: Use Visual Studio, Code::Blocks, Visual Studio Code, or any preferred IDE/text editor.
- Create a New Project:
- Name the project BasicCalculator.

INSTRUCTIONS

Detailed Steps

1. Function Definitions:

- addition(double a, double b): Returns the sum of a and b.
- subtraction(double a, double b): Returns the difference between a and b.
- multiplication(double a, double b): Returns the product of a and b.
- division(double a, double b): Returns the quotient of a and b, checks for division by zero.

2. displayMenu():

- Shows the menu options to the user.

3. main():

- Contains an infinite loop that continues until the user chooses to exit.
- Displays the menu and gets the user's choice.
- Validates the user's choice and handles invalid inputs.
- Prompts the user for two numbers and validates the input.
- Calls the appropriate arithmetic function based on the user's choice and displays the result.

INSTRUCTIONS

Step 3: Testing Your Calculator

- Test Addition: Try various pairs of numbers, including negative and large values.
- Test Subtraction: Similar tests as addition.
- Test Multiplication: Verify with small, large, and negative numbers.
- Test Division: Check normal cases and handle division by zero.
- Test Invalid Inputs: Enter non-numeric values and ensure the program handles them gracefully.
- Test Menu Choices: Ensure the program handles invalid menu choices properly.

Compiling and Running the Program

- Save your file as BasicCalculator.cpp.
- Open your terminal or command prompt.
- Navigate to the directory where your file is saved.
- Compile the program:
- sh

PROBLEMS

PROBLEM STATEMENT:

The objective is to create a command-line calculator that performs basic arithmetic operations (addition, subtraction, multiplication, division). The calculator should provide a menu for the user to choose an operation, accept two numeric inputs, display the result, and handle invalid inputs gracefully.

PROBLEMS

Division by zero can cause the program to crash or behave unexpectedly.

The user might enter a choice that is not in the menu

The user might enter a non-numeric value when prompted for numbers, causing cin to fail..

If the user enters an invalid input, the program might get stuck in an infinite loop.

The program might not exit gracefully when the user wants to quit.

CHALLENGES AND RESOLUTIONS

The solution involves:

- Check if the divisor is zero before performing the division.

Use `cin.fail()` to check if the input is valid and clear the input stream if it is not.

Ensure that the loop condition and input handling are robust enough to break out of the loop if necessary.

Make sure the loop breaks and the program provides a friendly exit message.

SOLUTION OVERVIEW

The solution involves:

1. Setting up the development environment.
2. Defining functions for each arithmetic operation.
3. Implementing a main function that interacts with the user, handles input validation, and displays results.
4. Testing the calculator to ensure it handles various inputs correctly.