Simple Calculator Application

Student Name: K.A.T.A. SANDARUWAN

Student ID: E107368 / NEG00302298

Course: Application Development

Instructor: Ms.Layanga Kaushani

Date: 10/11/2025

Contents

1.Introduction

The project documentation for my basic calculator application is presented in this document. This project's main goal is to use a Windows Form Application written in C# (by using Visual Studio) to design and implement a functional calculator that can perform four simple arithmetic operations.

2.Problem Analysis

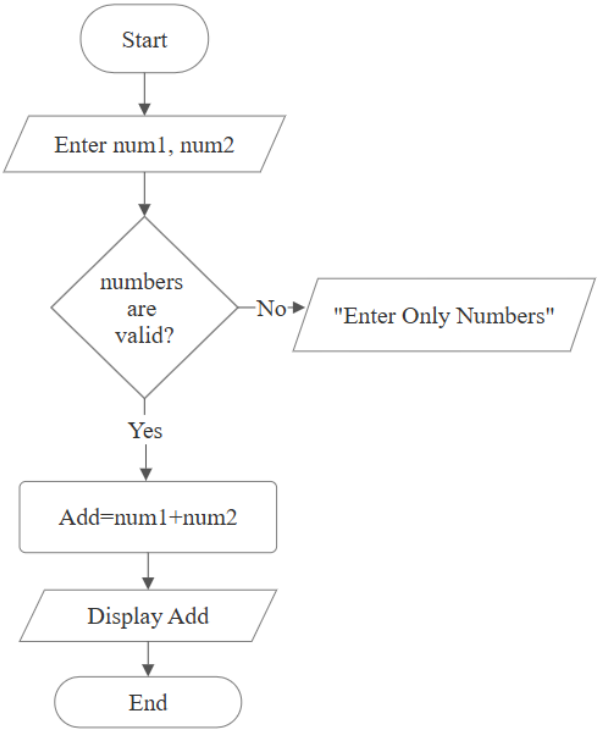
The calculator application's main functions are addition, subtraction, multiplication, and division. It is designed to take two numerical inputs from the user. After the user selects an operation (+, -, *, /), the application will process the inputs and display the calculated result.

Other than that we must recognize whether inputs have characters other than numbers and if not display message. Also we must address the issue when a number divided by zero.

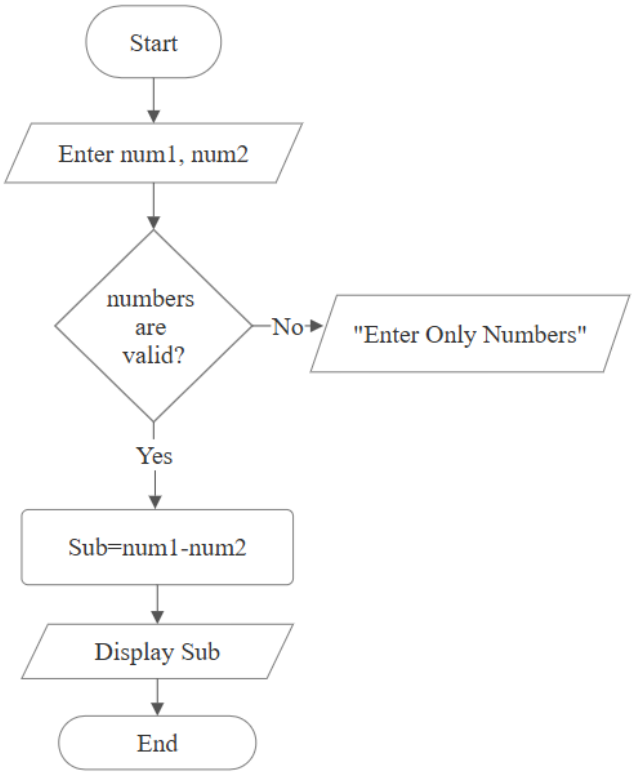
3.System Design

a. Flowchart

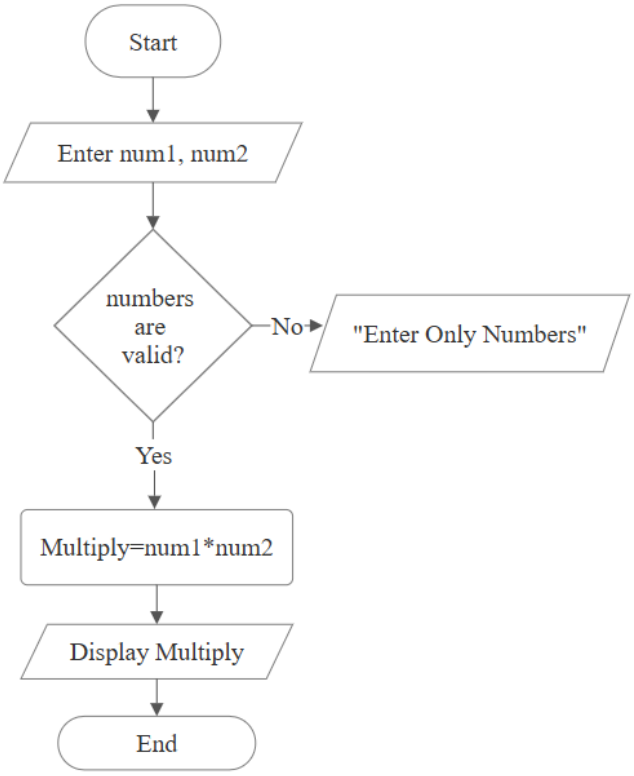
Flowchart for Addition



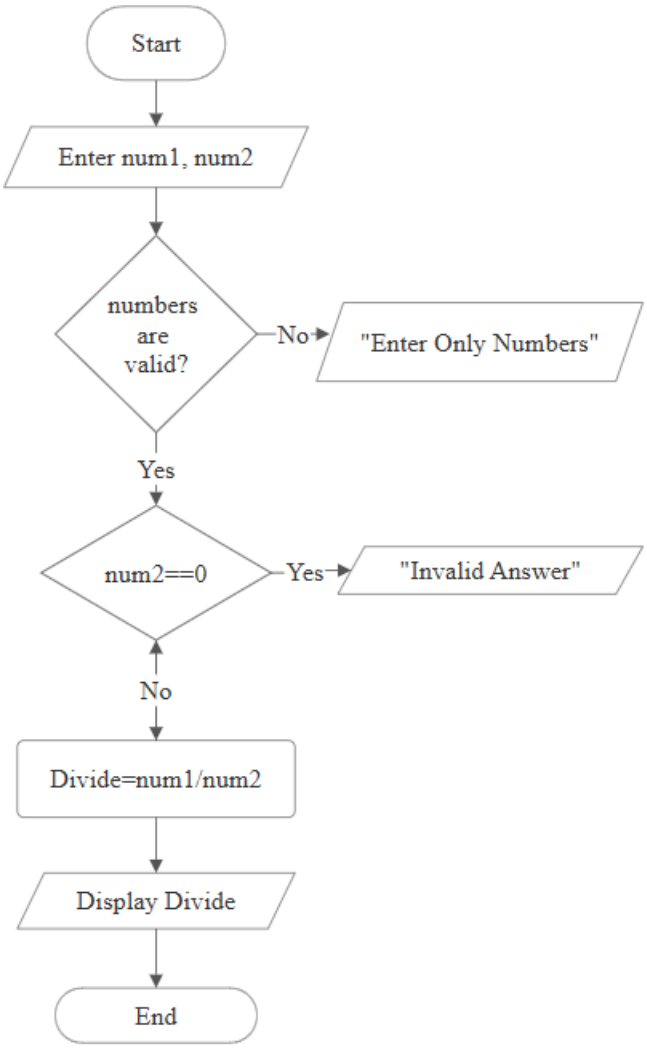
Flowchart for Subtraction



Flowchart for Multiply



Flowchart for Divide



b. Pseudocode

Pseudocode for Addition

BEGIN

READ num1 from txtnumber1

READ num2 from txtnumber2

IF either input is not a valid number THEN

```
    SHOW "Enter only numbers"
    STOP
  ENDIF

  Add = num1 + num2
  DISPLAY Add in txt_Answer
END
```

Pseudocode for Subtraction

```
BEGIN
  READ num1 from txtnumber1
  READ num2 from txtnumber2

  IF either input is not a valid number THEN
    SHOW "Enter only numbers"
    STOP
  ENDIF

  Subtract = num1 - num2
  DISPLAY Subtract in txt_Answer
END
```

Pseudocode for Multiply

```
BEGIN
  READ num1 from txtnumber1
  READ num2 from txtnumber2
```

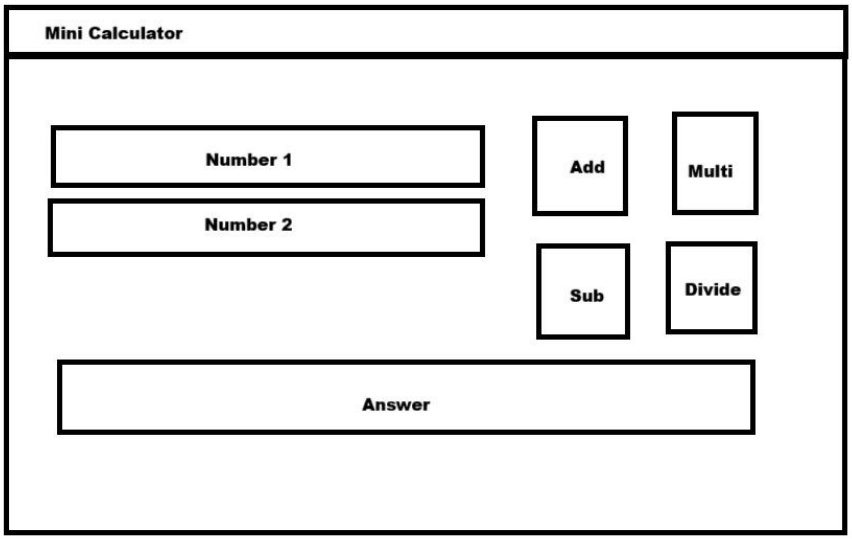
```
  IF either input is not a valid number THEN
     SHOW "Enter only numbers"
     STOP
  ENDIF

  Multiply = num1 * num2
  DISPLAY Multiply in txt_Answer
END
```

Pseudocode for Divide

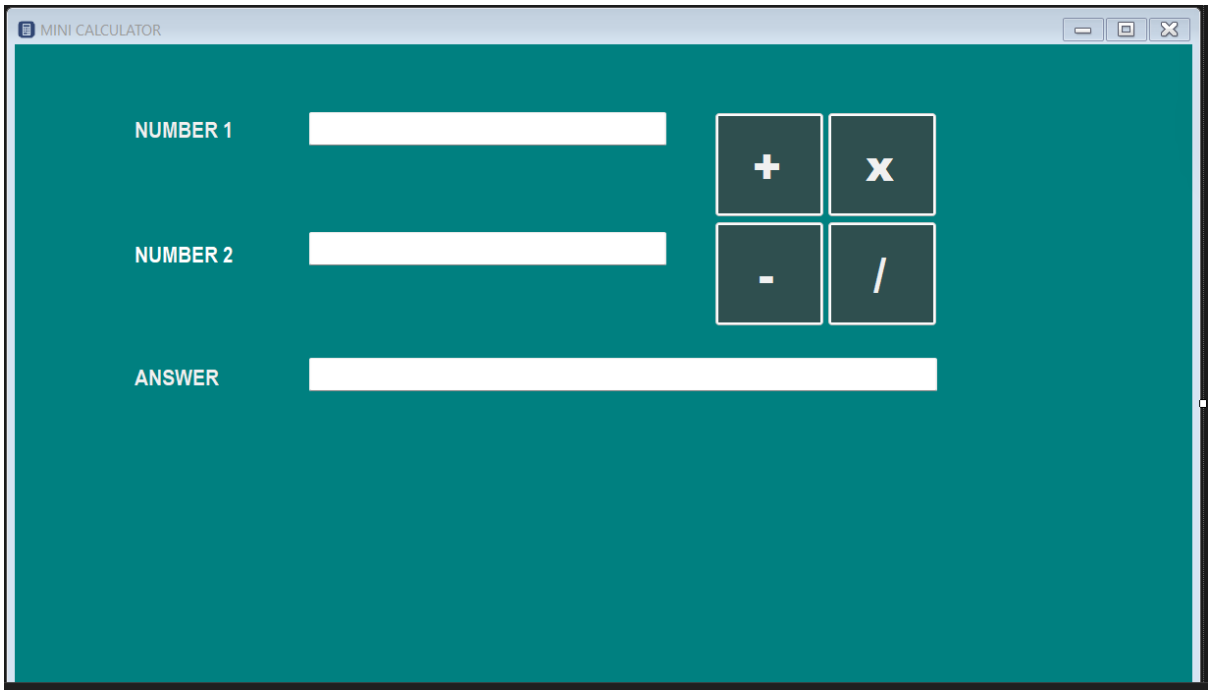
```
BEGIN
  READ num1 from txtnumber1
  READ num2 from txtnumber2

  IF either input is not a valid number THEN
     SHOW "Enter only numbers"
     STOP
  ENDIF

  IF num2 = 0 THEN
     SHOW "Cannot divide by zero"
     STOP
  ENDIF

  Divide = num1 / num2
  DISPLAY Divide in txt_Answer
END
```

c. Wire frame

Wire frame for the application



UI design



5.Implementation

IDE: Microsoft Visual Studio 2022 (or your version)

Framework: .NET Framework

Language: C#

Application Type: Windows Forms (.NET)

UI Controls

txt_number1: (TextBox) Used to input the first number.

txt_number2: (TextBox) Used to input the second number.

btn_add: (Button) Triggers the addition operation.

btn_sub: (Button) Triggers the subtraction operation.

btn_ultiply: (Button) Triggers the multiplication operation.

btn_ivide: (Button) Triggers the division operation.

txt_answer: (TextBox) Used to display the final result.

Application Logic

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace nov_9th
{
    4 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
```

```
        1 reference
        private void button1_Click(object sender, EventArgs e)

        {
            float num1, num2;
            if (!float.TryParse(txt_number1.Text, out num1) || !float.TryParse(txt_number2.Text, out num2))
                {
                MessageBox.Show("Enter only numbers");
                return;
            }
            float Add = num1 + num2;
            txt_answer.Text = Add.ToString();
        }


        1 reference
        private void label1_Click(object sender, EventArgs e)
        {


        }
```

The float is used to make sure the application is reliable. The TryParse() method is applied. This technique tries to safely turn the text in textboxes 1 and 2 into numerical values. It uses a return statement to halt the computation and show an error message if either conversion fails.

```
        1 reference
        public void txt_number2_TextChanged(object sender, EventArgs e)
        {
        }

        1 reference
        private void btn_sub_Click(object sender, EventArgs e)
        {
            float num1, num2;
            if (!float.TryParse(txt_number1.Text, out num1)||!float.TryParse(txt_number2.Text, out num2))
            {
                MessageBox.Show("Enter only numbers");
                return;
            }
            float Sub = num1 - num2;
            txt_answer.Text = Sub.ToString();
            {


            }
        }
```

```
1 reference
private void btn_multiply_Click(object sender, EventArgs e)
{
    float num1, num2;
    if (!float.TryParse(txt_number1.Text, out num1)||!float.TryParse(txt_number2.Text, out num2))
    {
        MessageBox.Show("Enter only numbers");
        return;
    }
    float multiply = num1 * num2;
    txt_answer.Text = multiply.ToString();
}

1 reference
private void btn_divide_Click(object sender, EventArgs e)
{
    float num1, num2;
    if (!float.TryParse(txt_number1.Text, out num1) || !float.TryParse(txt_number2.Text, out num2))
    {
        MessageBox.Show("Enter only numbers");
        return;
    }
        if
        (num2 == 0)
        {
        MessageBox.Show("Invalid Answer");
        return;
        }
    float Divide = num1 / num2;
    txt_answer.Text = Divide.ToString();
}

1 reference
private void lbl_number1_Click(object sender, EventArgs e)
{

}
}
}
```

A specific check is implemented in the buttonDivide_Click method to prevent a critical error. The code checks if the second number (num2) is zero. If it is, it displays a 'Invalid Answer' warning and stops the operation.

6.Testing

Test ID	Scenario	Test Steps	Expected Result	Actual Result	Status
T-01	Addition	1. Enter 10 in textBox1.2. Enter 5 in textBox2.3. Click buttonAdd.	The Result text should display 15.	The Result text displayed 15.	Pass
T-02	Subtraction	1. Enter 20 in textBox1.2. Enter 8 in textBox2.3. Click buttonSubtract.	The Result text should display 12.	The Result text displayed 12.	Pass

T-03	Multiplication	1. Enter 7 in textBox1.2. Enter 3 in textBox2.3. Click buttonMultiply.	The Result text should display 21.	The Result text displayed 21.	Pass
T-04	Division	1. Enter 100 in textBox1. 2. Enter 25 in textBox2. 3. Click button Divide.	The Result text should display 4.	The Result text displayed 4.	Pass
T-05	Test with Floats	1. Enter 7.5 in textBox1. 2. Enter 2 in textBox2. 3. Click button Multiply.	The Result text should display 15.	The Result text displayed 15.	Pass
T-06	Test with Negative Result	1. Enter 10 in textBox1. 2. Enter 20 in textBox2. 3. Click button Subtract.	The Result text should display -10.	The Result text displayed -10.	Pass
T-07	Error: Invalid Input (Text)	1. Enter abc in textBox1. 2. Enter 10 in textBox2. 3. Click button Add.	A MessageBox should appear with an error (e.g., "Please enter valid numbers").	MessageBox appeared with " Enter Only Numbers ".	Pass
T-08	Error: Invalid Input (Empty)	1. Leave textBox1 empty. 2. Enter 10 in textBox2.	A MessageBox should appear with an error (e.g., "Please enter valid numbers").	MessageBox appeared with "Enter Only Numbers".	Pass

		3. Click button Add.			
T-09	Error: Division by Zero	1. Enter 10 in textBox1. 2. Enter 0 in textBox2. 3. Click button Divide.	A MessageBox should appear with an error (e.g., "Invalid Answer").	MessageBox appeared with "Invalid Answer".	Pass

7.Conclution

This project was a success. I built the C# calculator application as planned. The final program meets all the requirements from the assignment. It correctly performs all four math operations: addition, subtraction, multiplication, and division. It also handles errors properly, showing a message for bad inputs (like "abc") and for "division by zero.

The complete project source code and documentation are available on GitHub: https://github.com/tharinda-ts/Tharinda_E107368_CalculatorAp