Laboratory Practical Report

of

**Visual Programming with C#**

**(ICT ED 465)**

Submitted To

**TRIBHUVAN UNIVERSITY**

In Partial Fulfillment of the Requirements of the course

**B.Ed. ICTE 6th Semester**

Submitted By

Sanam Tamang

Symbol No.: 76214020

T.U. Regd. No.: 9-2-214-54-2019

Under the guidance of

**Er. Santosh Dahal**

Lecturer

Sukuna Multiple Campus

**SUKUNA MULTIPLE CAMPUS**

Sundarharaincha-12, Morang, Nepal

2080

**CERTIFICATE**

This is to certify that the Laboratory Practical Report

of

**Visual Programming with C#**

**(ICT ED 465)**

In Partial Fulfillment of the Requirements of the course

**B.Ed. ICTE 6th Semester**

Submitted By

Sanam Tamang

Symbol No.: 76214020

T.U. Regd. No.: 9-2-214-54-2019

is a bonafide record of experiments carried out by him/her under by guidance.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Er. Santosh Dahal

Lecturer

Sukuna Multiple Campus

Sundarharaincha-12, Morang

(Internal Examiner)

Submitted for the Final Examination on: 2080/08/20

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lecturer

(External Examiner

**Table of Contents**

[**Program.cs** 1](#_Toc152559507)

[**Explanation of program.cs** 1](#_Toc152559508)

[**Form1.cs** 2](#_Toc152559509)

[**Explanation of form1.cs** 3](#_Toc152559510)

[**Output:** 5](#_Toc152559511)

# **Program.cs**

using System;

using System.Windows.Forms;

namespace Calculator

{

static class Program

{

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

## **Explanation of program.cs**

1. **Using Directives**:
   * using System: This directive allows the program to use types and members from the System namespace.
   * using System.Windows.Forms;: This directive enables the use of Windows Forms classes and controls.
2. **Namespace**:
   * The code is inside a namespace called Calculator. Namespaces are used to organize code and prevent naming conflicts.
3. **Static Class**Program:
   * The Program class is marked as static, which means it cannot be instantiated. It serves as the entry point for the application.
   * The [STAThread] attribute indicates that the application uses a single-threaded apartment model for COM interop.
4. **Main Method**:
   * The Main method is the starting point of the application.
   * Inside Main:
     + Application.EnableVisualStyles();: This line enables visual styles for the application, ensuring consistent appearance across different Windows versions.
     + Application.SetCompatibleTextRenderingDefault(false);: This line sets text rendering compatibility to false, which improves text rendering performance.
     + Application.Run(new Form1());: This runs the main form of the application, which is an instance of Form1.
5. **Form1**:
   * The application runs an instance of Form1. You’ll find the actual UI and logic for your calculator in the Form1 class.

# **Form1.cs**

using System;

using System.Windows.Forms;

namespace Calculator

{

public partial class Form1 : Form

{

Double resultValue = 0;

String operationPerformed = "";

bool isOperationPerformed = false;

public Form1()

{

InitializeComponent();

}

private void button\_click(object sender, EventArgs e)

{

if ((textBox\_Result.Text == "0") || (isOperationPerformed))

textBox\_Result.Clear();

isOperationPerformed = false;

Button button = (Button)sender;

if (button.Text == ".")

{

if(!textBox\_Result.Text.Contains("."))

textBox\_Result.Text = textBox\_Result.Text + button.Text;

}else

textBox\_Result.Text = textBox\_Result.Text + button.Text;

}

private void operator\_click(object sender, EventArgs e)

{

Button button = (Button)sender;

if (resultValue != 0)

{

button15.PerformClick();

operationPerformed = button.Text;

labelCurrentOperation.Text = resultValue + " " + operationPerformed;

isOperationPerformed = true;

}

else

{

operationPerformed = button.Text;

resultValue = Double.Parse(textBox\_Result.Text);

labelCurrentOperation.Text = resultValue + " " + operationPerformed;

isOperationPerformed = true;

}

}

private void button4\_Click(object sender, EventArgs e)

{

textBox\_Result.Text = "0";

}

private void button5\_Click(object sender, EventArgs e)

{

textBox\_Result.Text = "0";

resultValue = 0;

}

private void button15\_Click(object sender, EventArgs e)

{

switch (operationPerformed)

{

case "+":

textBox\_Result.Text = (resultValue + Double.Parse(textBox\_Result.Text)).ToString();

break;

case "-":

textBox\_Result.Text = (resultValue - Double.Parse(textBox\_Result.Text)).ToString();

break;

case "\*":

textBox\_Result.Text = (resultValue \* Double.Parse(textBox\_Result.Text)).ToString();

break;

case "/":

textBox\_Result.Text = (resultValue / Double.Parse(textBox\_Result.Text)).ToString();

break;

default:

break;

}

resultValue = Double.Parse(textBox\_Result.Text);

labelCurrentOperation.Text = "";

}

}

}

## **Explanation of form1.cs**

1. **Namespace and Class**:
   * The code is inside the Calculator namespace.
   * The Form1 class is defined, which represents the main form of the calculator application.
   * The class inherits from Form, which is a base class for creating Windows Forms.
2. **Fields and Variables**:
   * Double resultValue = 0;: This field stores the current result value of calculations.
   * String operationPerformed = "";: This field keeps track of the current mathematical operation being performed.
   * bool isOperationPerformed = false;: This boolean flag indicates whether an operation has been performed.
3. **Constructor (**public Form1()**)**:
   * Initializes the main form of the calculator.
   * Calls InitializeComponent() to set up the form’s components (buttons, labels, etc.).
4. **Button Click Event (**button\_click**)**:
   * Handles button clicks (numeric buttons, decimal point, etc.).
   * If the current display shows “0” or an operation was just performed, clears the display.
   * Appends the clicked button’s text to the display.
5. **Operator Click Event (**operator\_click**)**:
   * Handles operator buttons (addition, subtraction, etc.).
   * If a previous result exists, calculates it by simulating a click on the equals button (button15).
   * Updates the operationPerformed field and displays the current operation.
   * Sets the isOperationPerformed flag to true.
6. **Clear Buttons (**button4\_Click**and**button5\_Click**)**:
   * Resets the display to “0” and clears the result value.
7. **Equals Button (**button15\_Click**)**:
   * Evaluates the result based on the stored operation.
   * Uses a switch statement to handle different operations (addition, subtraction, etc.).
8. The switch statement checks the value of operationPerformed (which represents the current mathematical operation).
9. Depending on the operation, it performs the corresponding calculation (addition, subtraction, multiplication, or division) using the resultValue and the current value displayed in the textBox\_Result.
10. The updated result is then displayed in the textBox\_Result.
11. Finally, the resultValue is updated with the new calculated value, and the label showing the current operation is cleared.

# **Output:**

