

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 6<sup>th</sup> 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 6<sup>th</sup> 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
Explanation of program.cs .....	1
Form1.cs.....	2
Explanation of form1.cs .....	4
Output:.....	6

## Program.cs

```
namespace C_sharp_Calculator
{
    internal static class Program
    {
        static void Main()
        {
            ApplicationConfiguration.Initialize();
            Application.Run(new Form1());
        }
    }
}
```

## Explanation

### of program.cs

1. Program.cs is a entry point of this calculator software.
2. The namespace declaration specifies the logical grouping of related classes and types. In this case, the namespace is named C\_sharp\_Calculator.
3. Inside the namespace, we have a static class called Program. Static classes cannot be instantiated and are often used for utility methods or initialization tasks.
4. The Main() method is the entry point of the program. It is automatically executed when the application starts.
5. Within the Main() method:
  - ApplicationConfiguration.Initialize();: This line likely initializes some application-specific configuration settings. Unfortunately, without further context, I cannot provide specifics about what it does.
  - Application.Run(new Form1());: This line runs an instance of a Form1 class. The Form1 class is not defined in the provided code snippet, so I cannot explain its functionality without additional information.

## Form1.cs

```
namespace C_sharp_Calculator{
    public partial class Form1 : Form{
        int value1;
        int value2;
        double result = 0;
        string sign;
        public Form1(){
            InitializeComponent();
        }

        private void textBox1_TextChanged(object sender, EventArgs e){}

        private void button1_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "1";
        }

        private void button2_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "2";
        }

        private void button3_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "3";
        }

        private void button5_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "4";
        }

        private void button6_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "5";
        }

        private void button7_Click(object sender, EventArgs e) {
            textBox1.Text = textBox1.Text + "6";
        }

        private void button9_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "7";
        }

        private void button10_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "8";
        }

        private void button11_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "9";
        }

        private void button14_Click(object sender, EventArgs e){
            textBox1.Text = textBox1.Text + "0";
        }

        private void button13_Click(object sender, EventArgs e) {
            textBox1.Text = "";
            label1.Text = "";
            label2.Text = "";
        }
    }
}
```

```

    }

    private void button16_Click(object sender, EventArgs e){
        value1 = Convert.ToInt32(textBox1.Text);
        sign = "+";
        label1.Text = textBox1.Text + sign;
        textBox1.Text = "";
    }

    private void button4_Click(object sender, EventArgs e){
        value1 = Convert.ToInt32(textBox1.Text);
        sign = "-";
        label1.Text = textBox1.Text + sign;
        textBox1.Text = "";
    }

    private void button8_Click(object sender, EventArgs e){
        value1 = Convert.ToInt32(textBox1.Text);
        sign = "*";
        label1.Text = textBox1.Text + sign;
        textBox1.Text = "";
    }

    private void button12_Click(object sender, EventArgs e) {
        value1 = Convert.ToInt32(textBox1.Text);
        sign = "/";
        label1.Text = textBox1.Text + sign;
        textBox1.Text = "";
    }

    private void button18_Click(object sender, EventArgs e) {
        value2 = Convert.ToInt32(textBox1.Text);
        label2.Text = textBox1.Text;
        if (sign == "+"){
            result = value1 + value2;
            textBox1.Text = Convert.ToString(result);
        }
        else if (sign == "-"){
            result = value1 - value2;
            textBox1.Text = Convert.ToString(result);
        }
        else if (sign == "*"){
            result = value1 * value2;
            textBox1.Text = Convert.ToString(result);
        }
        else if (sign == "/"){
            if (value2 == 0){
                textBox1.Text = Convert.ToString("");
                textBox1.Text = "infinity";
            }
            Else{
                result = value1 / value2;
                textBox1.Text = Convert.ToString(result);
            }
        }
    }
}

```

## Explanation of form1.cs

### 1. Namespace and Class:

- The code is part of the C\_sharp\_Calculator namespace.
- The Form1 class is a partial class that inherits from the Form class. It represents the main form of your calculator application.

### 2. Fields and Variables:

- int value1; and int value2;; These integer fields are used to store numeric values for calculations.
- double result = 0;; This field holds the result of calculations (initialized to 0).
- string sign;; This string field stores the mathematical operation (e.g., addition, subtraction).

### 3. Constructor:

- The public Form1() constructor initializes the form by calling InitializeComponent(). This method sets up the form's visual components and event handlers.

### 4. Event Handlers:

- textBox1\_TextChanged(object sender, EventArgs e): This event handler is triggered when the text in textBox1 (presumably the display area) changes. However, the provided code does not contain any specific logic for this event.
- Numeric Button Click Handlers (e.g., button1\_Click, button2\_Click, etc.):
  - These methods are executed when the corresponding numeric buttons (1 to 9) are clicked.
  - They append the clicked number to the text displayed in textBox1.
- button14\_Click(object sender, EventArgs e): This method handles the "0" button click. It appends "0" to the text in textBox1.
- button13\_Click(object sender, EventArgs e): This method clears the text in textBox1 and resets the labels (label1 and label2) to empty strings.

### 5. User Interface Elements:

- textBox1: Represents the display area where the user sees the entered numbers and results.
- Buttons (e.g., button1, button2, etc.): These buttons allow the user to input numeric values.

### 6. Labels

- label1 and label2: These labels are presumably used for additional information or display purposes. However, their purpose is not explicitly defined in the provided code snippet.
- private void button16\_Click(object sender, EventArgs e): This method handles the "16" button click. It converts the text in textBox1 to an integer and assigns it to

the value1 field. It also sets the sign field to “+” and displays it in label1. Then, it clears the text in textBox1.

- `private void button4_Click(object sender, EventArgs e)`: This method handles the “4” button click. It does the same thing as the previous method, but with a different value for the sign field.
- `private void button8_Click(object sender, EventArgs e)`: This method handles the “8” button click. It does the same thing as the previous methods, but with a different value for both fields.
- `private void button12_Click(object sender, EventArgs e)`: This method handles the “12” button click. It does the same thing as the previous methods, but with a different value for both fields.
- `private void button18_Click(object sender, EventArgs e)`: This method handles the “18” button click. It converts the text in textBox1 to an integer and assigns it to the value2 field. It also sets the label2 field to display the same text as textBox1.
- It then checks the value of the sign field, which is presumably a variable that stores the mathematical operation (e.g., “+”, “-”, “\*”, or “/”). If the sign is “+”, it adds value1 and value2 and assigns the result to a new variable called result. It then displays this result in textBox1.
- If the sign is “-”, it subtracts value1 and value2 and assigns the result to a new variable called result. It then displays this result in textBox1.
- If the sign is “\*”, it multiplies value1 and value2 and assigns the result to a new variable called result. It then displays this result in textBox1.
- If the sign is “/”, it performs a conditional check on whether value2 is zero or not.
  - If value2 is zero, it clears both text fields (textBox1) and displays “infinity” in label2.
  - If value2 is not zero, it divides value1 and value2 and assigns the result to a new variable called result. It then displays this result in textBox1.



## Output

