.NET C# Term Work

Sadik Saifi

Nov 28, 2022

Contents

1 Create a C# program to perform arithmetic Operations.

Aim

To make a calculator using .NET C#.

Algorithm

1. Start
2. Read input from textBox1 and textBox2.
3. Then perform the calculation according to button pressed by user.
4. If user click on add/sub/mul/div button then display 'textBox1+textBox2'/ 'textBox1-textBox2'/'textBox1*textBox2'/'textBox1/textBox2' in textBox3.
5. End.

Source Code

```
using System;
using System.Windows.Forms;

namespace calculator_cs
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            int num = int.Parse(textBox1.Text);
            int num2 = int.Parse(textBox2.Text);
            int sum = num + num2;
            textBox3.Text = sum.ToString();
        }
        private void button2_Click_1(object sender, EventArgs e)
        {
            int num = int.Parse(textBox1.Text);
            int num2 = int.Parse(textBox2.Text);
            int sub = num - num2;
            textBox3.Text = sub.ToString();
        }
        private void button3_Click_1(object sender, EventArgs e)
        {
            int num = int.Parse(textBox1.Text);
            int num2 = int.Parse(textBox2.Text);
            double mul = num * num2;
            textBox3.Text = mul.ToString();
        }
        private void button4_Click_1(object sender, EventArgs e)
```

```
        {
            int num = int.Parse(textBox1.Text);
            int num2 = int.Parse(textBox2.Text);
            float div = num / num2;
            textBox3.Text = div.ToString();
        }

    }
}
```

Output

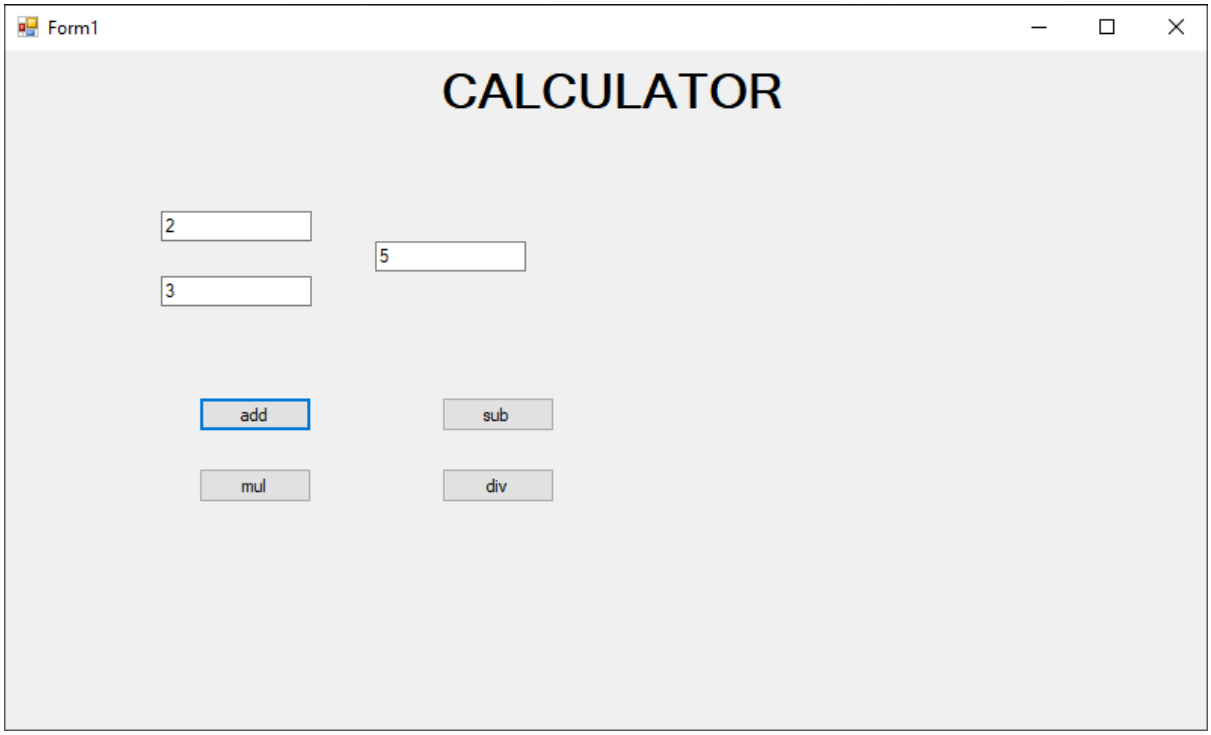

Figure 1: yo

2 Create a C# program to calculate Simple interest and compound Interest.

Aim

To make a simple interest and compound interest calculator.

Algorithm

1. Start
2. Read all the values from user like principle amount, annual rate, period.
3. Add a CALCULATE button.
4. Inside calculate button find the interest earned with the respective formula for S.I./C.I.
   - si = (principle amount x rate x years)/100
   - ci = amount - principle amount
     - amount = principle amount x (1 + r/100)^years
5. Display the interest and total amount earned in respective textBox for S.I./C.I.
6. End

Source Code

```
using System;
using System.Windows.Forms;

namespace simple_compound_interest_cs
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int si = (int.Parse(textBox1.Text) * int.Parse(textBox2.Text) *
            int.Parse(textBox3.Text)) / 100;

            int totalValue = si + int.Parse(textBox1.Text);

            textBox4.Text = si.ToString();
            textBox5.Text = totalValue.ToString();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            float p = float.Parse(textBox6.Text);
            float r = float.Parse(textBox7.Text);
            float t = float.Parse(textBox8.Text);
```

```
            double a = p * Math.Pow((1+r/100), t);

            double ci = a - p;

            textBox10.Text = ci.ToString();
            textBox11.Text = a.ToString();
        }
    }
}
```

Output

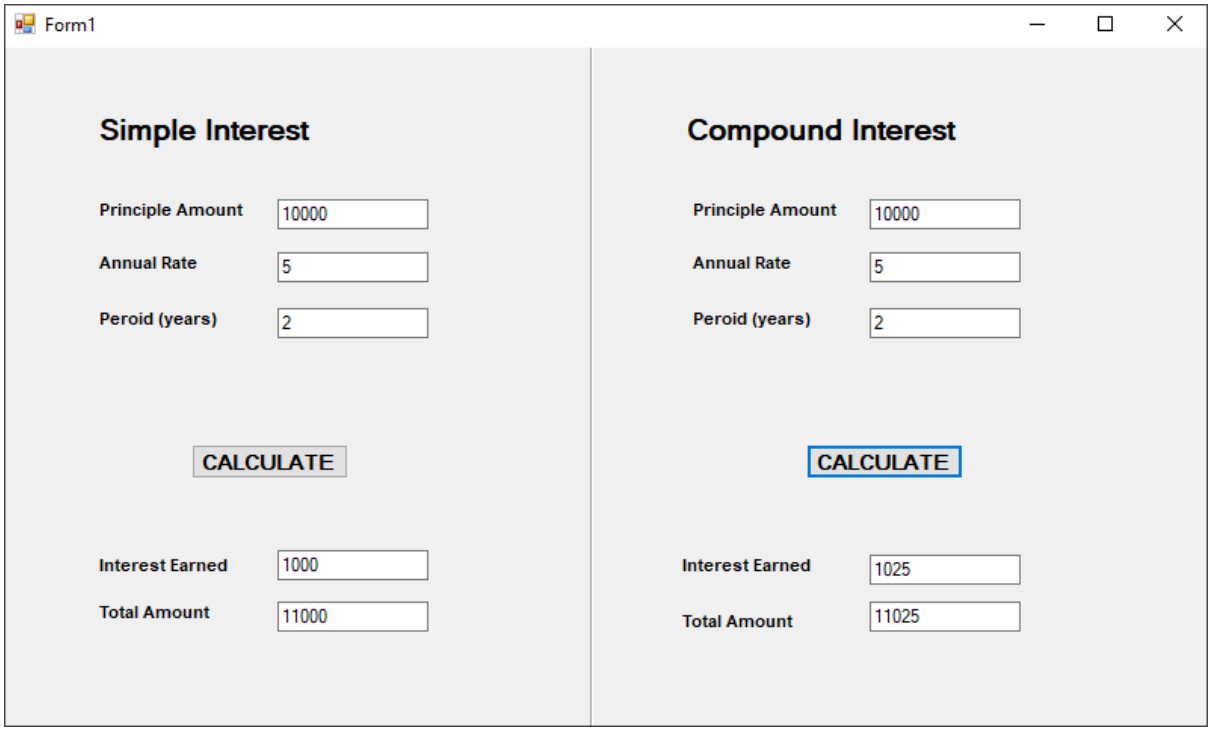

Figure 2: Interest Calculator

3 Create a C# program to create a STACK. Perform all the operations of Stack

Aim

To illustrate stack using listBox in .NET C#.

Algorithm

1. Start
2. First we will create an object of Stack.
3. Then will push some items in a stack at runtime.
4. Then we will create five buttons for five task which are :-
    - Push
    - Peek
    - Pop
    - Count
    - Clear
5. For every task we will use their respective methods provided by the stack object which are stack.Push(), stack.Peek(), stack.Pop(), stack.Count(), stack.clear();
6. Display the respective task.
7. End.

Source Code

```
using System;
using System.Windows.Forms;
using System.Collections;

namespace stack_implementation
{
    public partial class Form1 : Form
    {

        Stack days = new Stack();

        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            days.Push("Sunday");
            days.Push("Monday");
            days.Push("Tuesday");
            days.Push("Wednesday");
            days.Push("Thrusday");
            days.Push("Friday");
            days.Push("Saturday");
```

```
            foreach (String day in days)
                    listBox1.Items.Add(day);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            days.Push(textBox1.Text);
        }
        private void button2_Click(object sender, EventArgs e)
        {
            days.Peek();        }
        private void button3_Click(object sender, EventArgs e)
        {
            days.Pop();        }
        private void button4_Click(object sender, EventArgs e)
        {
           MessageBox.Show(days.Count.ToString());
        }
        private void button6_Click(object sender, EventArgs e)
        {
            days.Clear();
        }

    }
}
```

Output

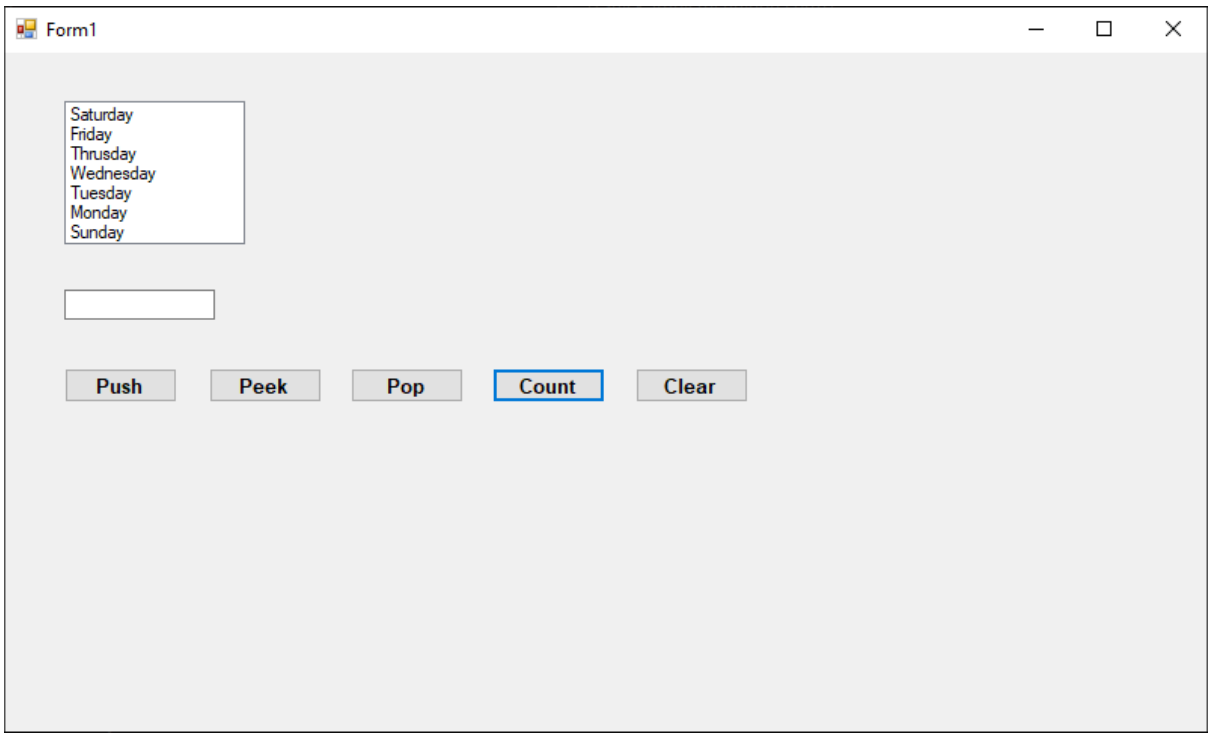

Figure 3: Stack Implementation

4 Create a C# program to implement Array List Perform any five operations in

Array List.

Aim

To illustrate array list using comboBox.

Algorithm

1. Start
2. First we will add comboBox to canvas.
3. By using comboBox method comboBox.Item.Add we will add several items.
4. Then we will create four buttons for respective tasks which are add, remove, RemoveAt(works on index), clear.
5. Methods for respective tasks are:
   - comboBox.Items.Add()
   - comboBox.Items.Remove()
   - comboBox.Items.RemoveAt()
   - comboBox.Items.Clear()
6. End.

Source Code

```
using System;
using System.Windows.Forms;

namespace comboBox_cs
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            comboBox1.Items.Add("A");
            comboBox1.Items.Add("B");
            comboBox1.Items.Add("C");
            comboBox1.Items.Add("D");
            comboBox1.Items.Add("E");
        }
        private void button1_Click(object sender, EventArgs e)
        {
            comboBox1.Items.Add(textBox1.Text);
        }
        private void button2_Click(object sender, EventArgs e)
        {
            comboBox1.Items.Remove(textBox1.Text);
```

```
        }
        private void button3_Click(object sender, EventArgs e)
        {
            comboBox1.Items.RemoveAt(int.Parse(textBox1.Text));
        }
        private void button4_Click(object sender, EventArgs e)
        {
            comboBox1.Items.Clear();
        }
    }
}
```

Output

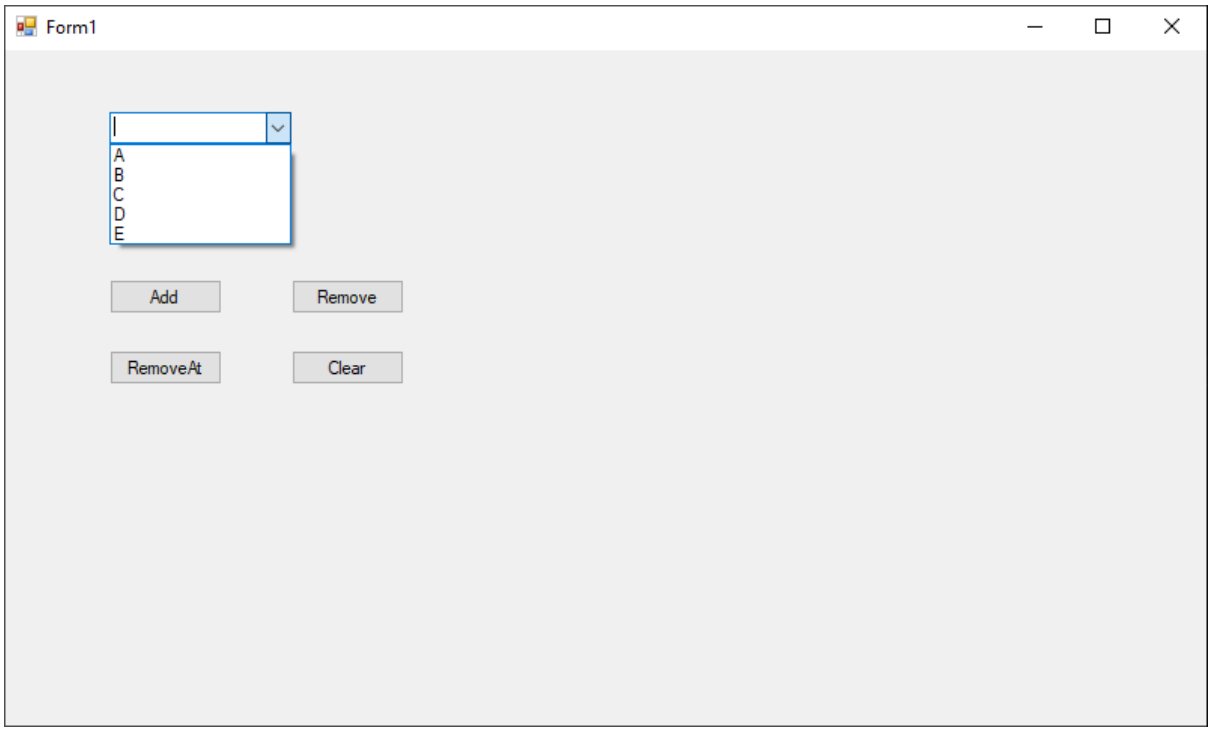

Figure 4: Array List using ComboBox

5 Create a program to calculate the total marks of the student and print the

grade using Select Case.

Aim

To learn about labels, textBoxes, and several other tools in .NET framework.

Algorithm

1. Start
2. Read all values for internal and external marks for respective subjects.
3. Display the total of respective subjects in other textBox.
4. Calculate the total marks of the student by adding total marks of each subject and display it textBox.
5. Calculate Percentage = (totalMarksObtainted / 400) x 100 and display it.
6. Then display the grade of the student according to percentage student got.
7. Display a message whether the student has pass/failed.
8. End

Source Code

```
using System;
using System.Windows.Forms;

namespace student_mark_list_cs
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            float engInt = float.Parse(textBox1.Text);
            float engExt = float.Parse(textBox2.Text);

            float hindiInt = float.Parse(textBox3.Text);
            float hindiExt = float.Parse(textBox4.Text);

            float mathInt = float.Parse(textBox5.Text);
            float mathExt = float.Parse(textBox6.Text);

            float phyInt = float.Parse(textBox7.Text);
            float phyExt = float.Parse(textBox8.Text);

            float engTotal = engInt + engExt;
```

```
float hindiTotal = hindiInt + hindiExt;

float mathTotal = mathInt + mathExt;

float phyTotal = phyInt + phyExt;

textBox9.Text = engTotal.ToString();

textBox10.Text = hindiTotal.ToString();

textBox11.Text = mathTotal.ToString();

textBox12.Text = phyTotal.ToString();

// Display Total Marks Obtained
float totalMarksObtainted = engTotal + hindiTotal + mathTotal +
phyTotal;
textBox15.Text = totalMarksObtainted.ToString();

// Display percentage
float per = (totalMarksObtainted / 400) * 100;
textBox13.Text = per.ToString();

String grade = "";

if (per > 90)
    grade = "A";
else if (per > 80 && per < 90)
    grade = "B";
else if (per > 70 && per < 80)
    grade = "C";
else if (per > 60 && per < 70)
    grade = "D";
else if (per > 50 && per < 60)
    grade = "E";
else if (per < 50)
    grade = "F";

// Displaying the grade
textBox14.Text = grade;

// Giving a message whether student has passed or failed.

String result = "";

if (grade == "F")
    result = "Fail";
else
    result = "Pass";

// Display the message
```

```
            MessageBox.Show(result);
        }
    }
}
```

Output

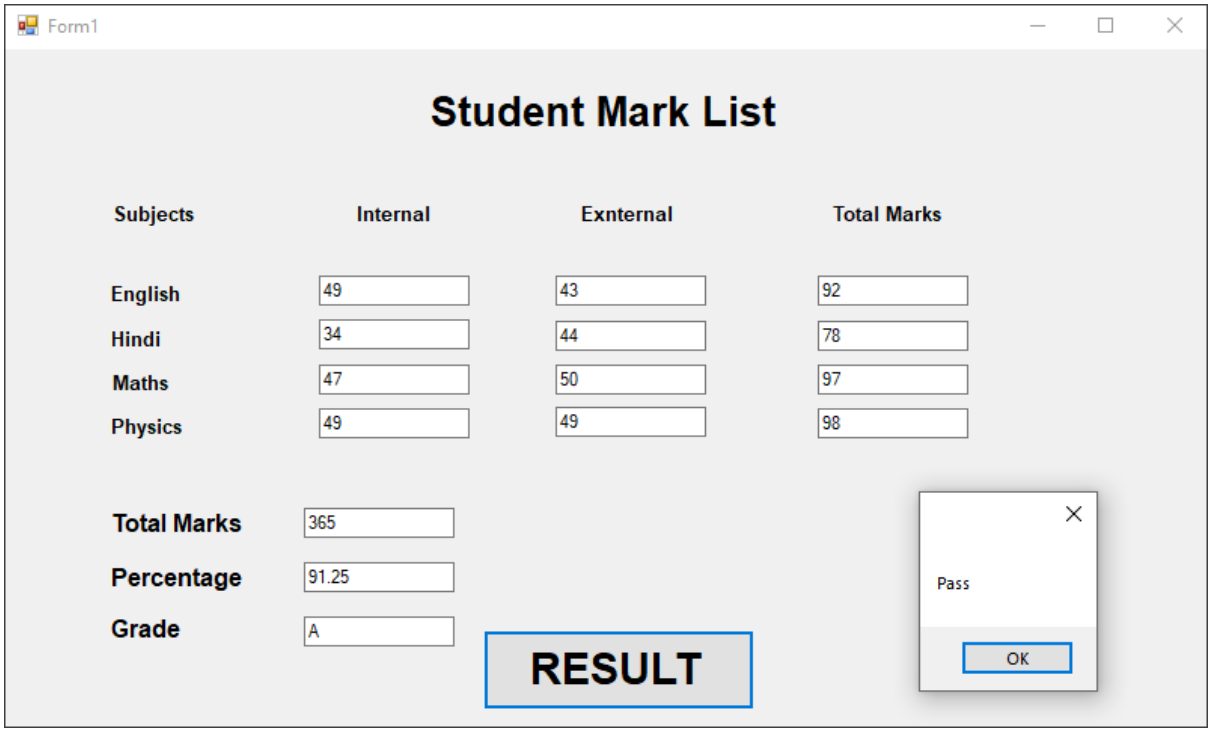

Figure 5: Student Marks List

6 Create a C# program to implement Menu. Fill the forms with different colors.

Aim

To make use of menustrip to design main menu to add different color and images to window.

Algorithm

1. Start
2. We will add a menu strip with respective attributes File, Color(Red, Green, Blue), BackgroundImage(IMG1, IMG2, IMG3).
3. We will create a object of Form2.
4. Then we will change the background color of the form by using frm2.BackColor variable which available in frm2 object.
5. Just like background color we also have frm2.BackgroundImage variable to set a different background so we will use that to change background.
6. End

Source Code

```
namespace Menu_Program
{
    public partial class Form1 : Form
    {
        Form frm2 = new Form2();
        public Form1()
        {
            InitializeComponent();
        }
        private void newToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frm2.MdiParent = this;
            frm2.Show();
        }
        private void redToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frm2.BackColor = Color.Red;
        }
        private void greenToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frm2.BackColor= Color.Green;
        }
        private void blueToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frm2.BackColor=Color.Blue;
        }
        private void iMG1ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frm2.BackgroundImage = imageList1.Images[0];
```

```
        }
        private void iMG2ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frm2.BackgroundImage = imageList1.Images[1];
        }
        private void iMG3ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frm2.BackgroundImage = imageList1.Images[2];
        }
        private void closeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            frm2.Close();
        }
    }
}
```

Output

Figure 6: Menu Strip

7 Create a program to animate the picture using Timer Control.

Aim

To make use of timer to add animations using V.B.

Algorithm

1. Start
2. First we will initialize xchange and ychange as 100.
3. We will add picture box and also add a picture.
4. Then we will add a timer control.
5. We will displace the position of PictureBox1 by xchange and ychange values to the left and top.
6. Then we will animate the picture by using If else statement by comparing with width and height of the picture.
7. End

Source Code

```
Public Class Form1
    Dim xchange As Short
    Dim ychange As Short
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        xchange = 100
        ychange = 100
    End Sub
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        Me.WindowState = FormWindowState.Normal
        PictureBox1.Left = PictureBox1.Left + xchange
        PictureBox1.Top = PictureBox1.Top + ychange
        If PictureBox1.Left + PictureBox1.Top > width Then
            xchange = xchange * -1
            PictureBox1.BackColor = Color.Sienna
            PictureBox1.Width = 50
        End If
        If PictureBox1.Left < 0 Then
            xchange = xchange * -1
            PictureBox1.BackColor = Color.Violet
            PictureBox1.Width = 100
        End If
        If PictureBox1.Top + PictureBox1.Height > Height Then
            ychange = ychange * -1
            PictureBox1.BackColor = Color.Tomato
            PictureBox1.Width = 150
        End If
        If PictureBox1.Top < 0 Then
            ychange = ychange * -1
            PictureBox1.BackColor = Color.SeaGreen
            PictureBox1.Width = 200
```

```
        End If
    End Sub

        Private Sub MouseClick(sender As Object, e As EventArgs) Handles
        MyBase.Click
        End
    End Sub
End Class
```

Output

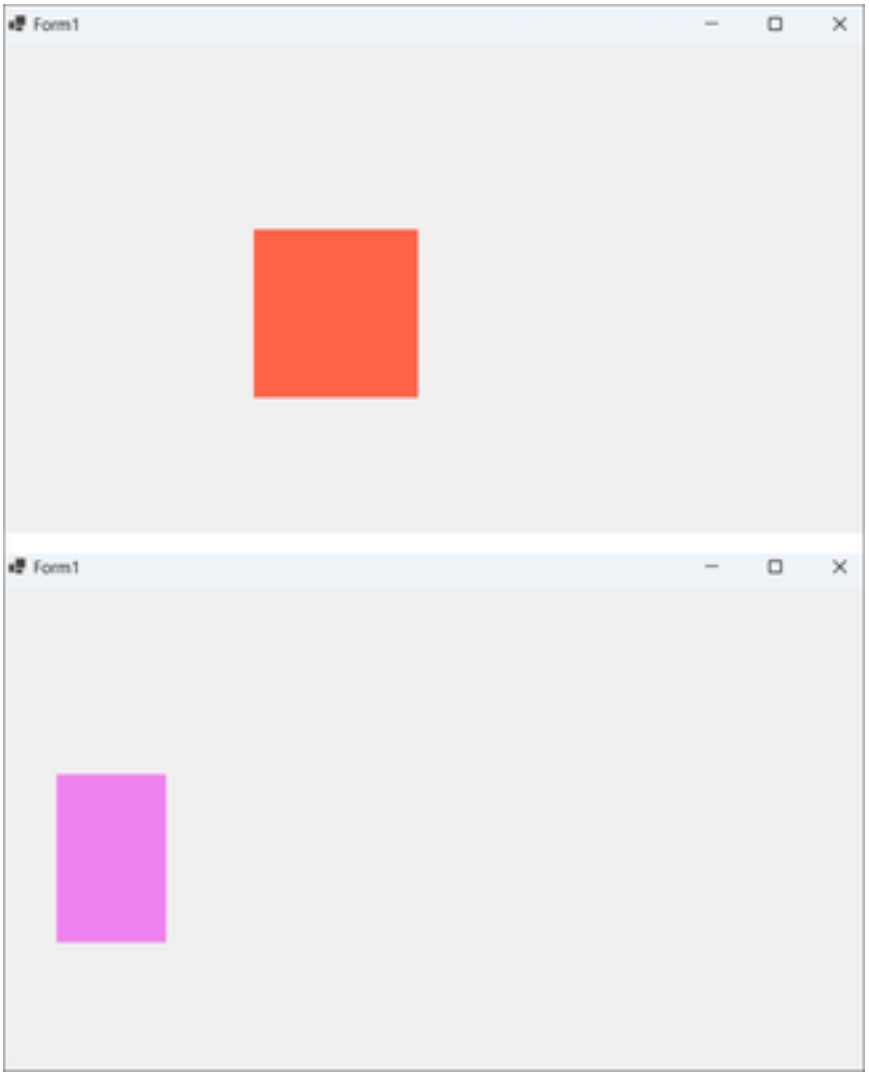

Figure 7: Animated Image

8 Create a C# Program to implement Vowels Program using Select Case.

Aim

To check an alphabet is vowel or consonant.

Algorithm

1. Start
2. Read the alphabet from user.
3. We will use a switch case statement to check whether an alphabet is vowel or consonant
4. If alphabet is vowel then we will display vowel and vice-versa.
5. End

Source Code

```
using System;
using System.Windows.Forms;

namespace vowels_cs
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            String word = textBox1.Text;
            switch (word)
            {
                case "a":
                    MessageBox.Show("Vowel");
                    break;
                case "e":
                    MessageBox.Show("Vowel");
                    break;
                case "i":
                    MessageBox.Show("Vowel");
                    break;
                case "o":
                    MessageBox.Show("Vowel");
                    break;
                case "u":
                    MessageBox.Show("Vowel");
                    break;
                default:
```

```
                    MessageBox.Show("Consonet");
                    break;
                }
            }
        }
    }
```

Output

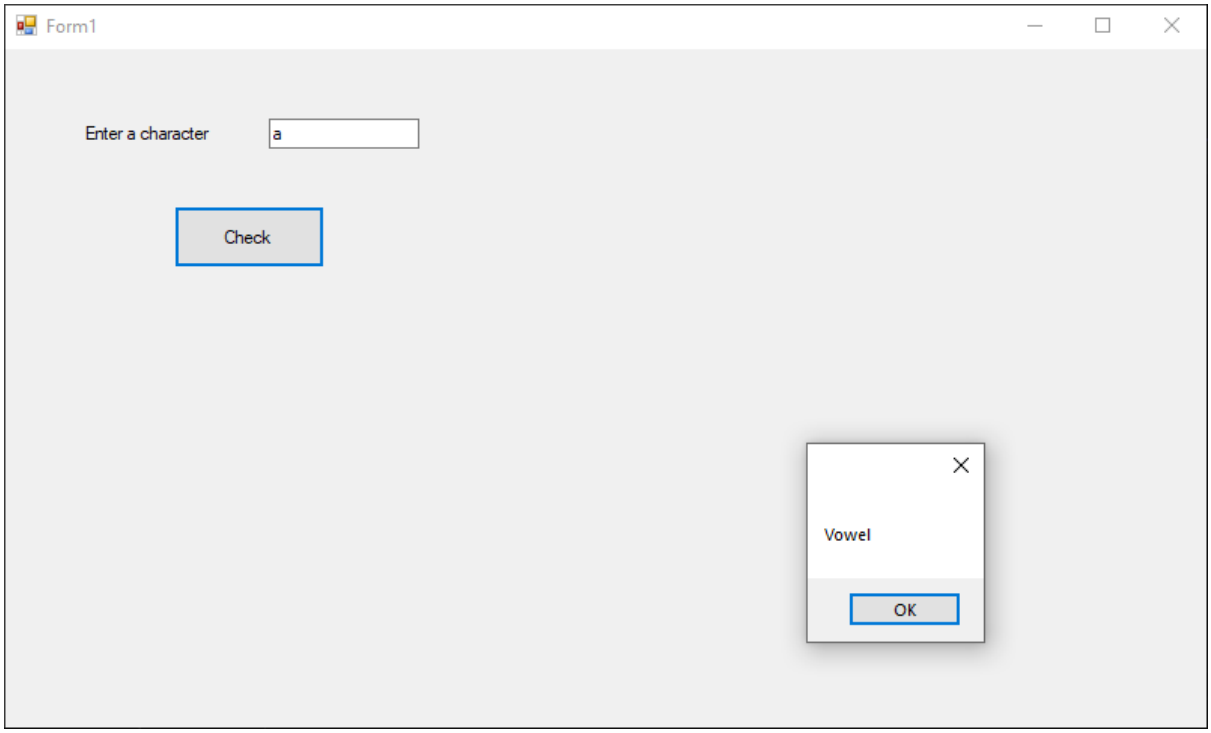

Figure 8: Check Vowels and Consonet

9 Create a C# Console Application program to print the area and cost of a rectangle using Inheritance.

Aim

To learn about inheritance in C#.

Algorithm

1. Start
2. Read height and width.
3. Calculate area by multiplying height and width.
4. Calculate cost by multiplying area * money;
5. Display area and cost.
6. End

Source Code

```
using System;
class Rectangle
{
    public int height, width;
    public void getData()
    {
        Console.WriteLine("Enter height and width of the rectangle : ");
        height = Convert.ToInt32(Console.ReadLine());
        width = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine();
    }
    public int area()
    {
        int area = height * width;
        return area;
    }
class Cost : Rectangle
    {
        public int cost;

        public int getCost()
        {
            cost = area() * 50;
            return cost;
        }
        public void display()
        {
        Console.Write("Area of Rectangle : " + area());
        Console.WriteLine();
        Console.Write("Area of Rectangle : " + getCost());
        }
public static void Main(string[] args)
        {
```

```
            Cost cost = new Cost();
            cost.getData();
            cost.display();
        }
    }
}
```

Output

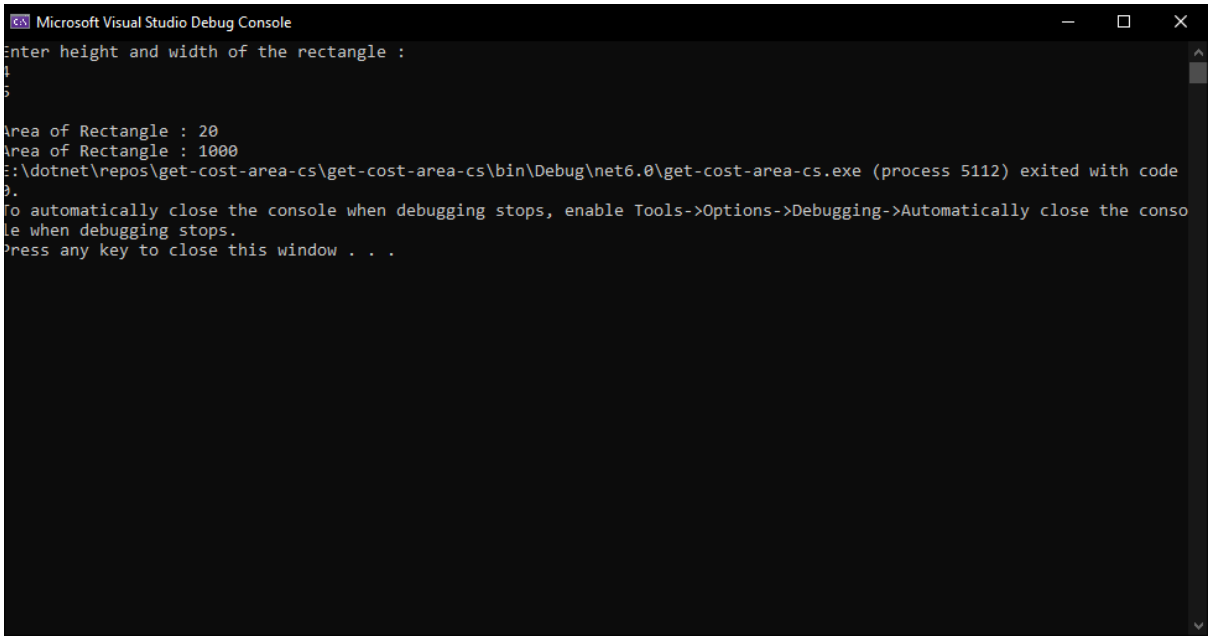

Figure 9: Area and Cost Of Rectangle

10 Create a C# program to implement Notepad.

Aim

To make use of dialog boxes to create Notepad.

Algorithm

1. Start
2. We will add a menu strip.
3. We create an object of OpenFileDialog as ofd.
4. We will use respective method to do different task like ofd.Title, ofd.Filter, to change title and to filter files.
5. We will check if ShowDialog = DialogResult then we will open file and also read the data using StreamReader by creating its object.
6. We will create a tool save in menustrip.
7. We will create an object of SaveFileDialog class and we will use its methods like .Title and .Filter.
8. Check if this.Text = untitled_notepad then .ShowDialog() that name = .File-Name. else we will use StreamWriter class to save the file by using its .Write-Line(), Flush(), Close() methods.
9. Then we will create several tools like close, font, color, date, undo, redo, cut, copy, paste and we will use respective classes and methods for them which are this.dialog, fontDialog.ShowDialog(), fontDialog.Font, colorDialog.ForeColor, System.DateTime.Now, .Undo(), .Redo(), .Cut(), .Copy(), .Paste().
10. End

Source

```
using System;
using System.Windows.Forms;

namespace notepad
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void newToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Form1 frm = new Form1();
            frm.Show();

            richTextBox1.Clear();
            this.Text = "untitled_notepad";
        }

        private void openToolStripMenuItem_Click(object sender, EventArgs e)
        {
```

```
    OpenFileDialog ofd = new OpenFileDialog();

    ofd.Title = "OPEN";
    ofd.Filter = "Text Document(.*txt)|*.txt|All files(*.*)|*.*";

    if (ofd.ShowDialog() == DialogResult.OK)
    {
        System.IO.Stream contains = ofd.OpenFile();
        StreamReader m_StreamReader = new StreamReader(contains);
        int check = m_StreamReader.Read();
        String str = "";

        while (check != -1)
        {
            str += (char)check;
            check = m_StreamReader.Read();
        }

        richTextBox1.Text = str;
        this.Text = ofd.FileName;
        contains.Close();
    }
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    string str = richTextBox1.Text;
    string name;
    SaveFileDialog sfd = new SaveFileDialog();
    sfd.Title = "save";
    sfd.Filter = "Text Document(.*txt)|*.txt|All files(*.*)|*.*";

    if (this.Text == "untitled_notepad")
    {
        sfd.ShowDialog();
        name = sfd.FileName;
    }
    else
        name = this.Text;

    StreamWriter m_writer = new StreamWriter(name);

    m_writer.WriteLine(str);
    m_writer.Flush();
    m_writer.Close();
}

private void Form1_Load(object sender, EventArgs e)
{
    this.Text = "untitled_notepad";
}
```

```
        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void fontToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FontDialog fd = new FontDialog();
            fd.ShowDialog();
            richTextBox1.Font = fd.Font;
        }

        private void colorToolStripMenuItem_Click(object sender, EventArgs e)
        {
            ColorDialog cd = new ColorDialog();
            cd.ShowDialog();
            richTextBox1.ForeColor = cd.Color;
        }

        private void dateTimeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            richTextBox1.Text += " "+System.DateTime.Now.ToString();
        }

        private void undoToolStripMenuItem_Click(object sender, EventArgs e)
        {
            richTextBox1.Undo();
        }

        private void redoToolStripMenuItem_Click(object sender, EventArgs e)
        {
            richTextBox1.Redo();
        }

        private void cutToolStripMenuItem_Click(object sender, EventArgs e)
        {
            richTextBox1.Cut();
        }

        private void copyToolStripMenuItem_Click(object sender, EventArgs e)
        {
            richTextBox1.Copy();
        }

        private void pasteToolStripMenuItem_Click(object sender, EventArgs e)
        {
            richTextBox1.Paste();
        }
    }
}
```

Output

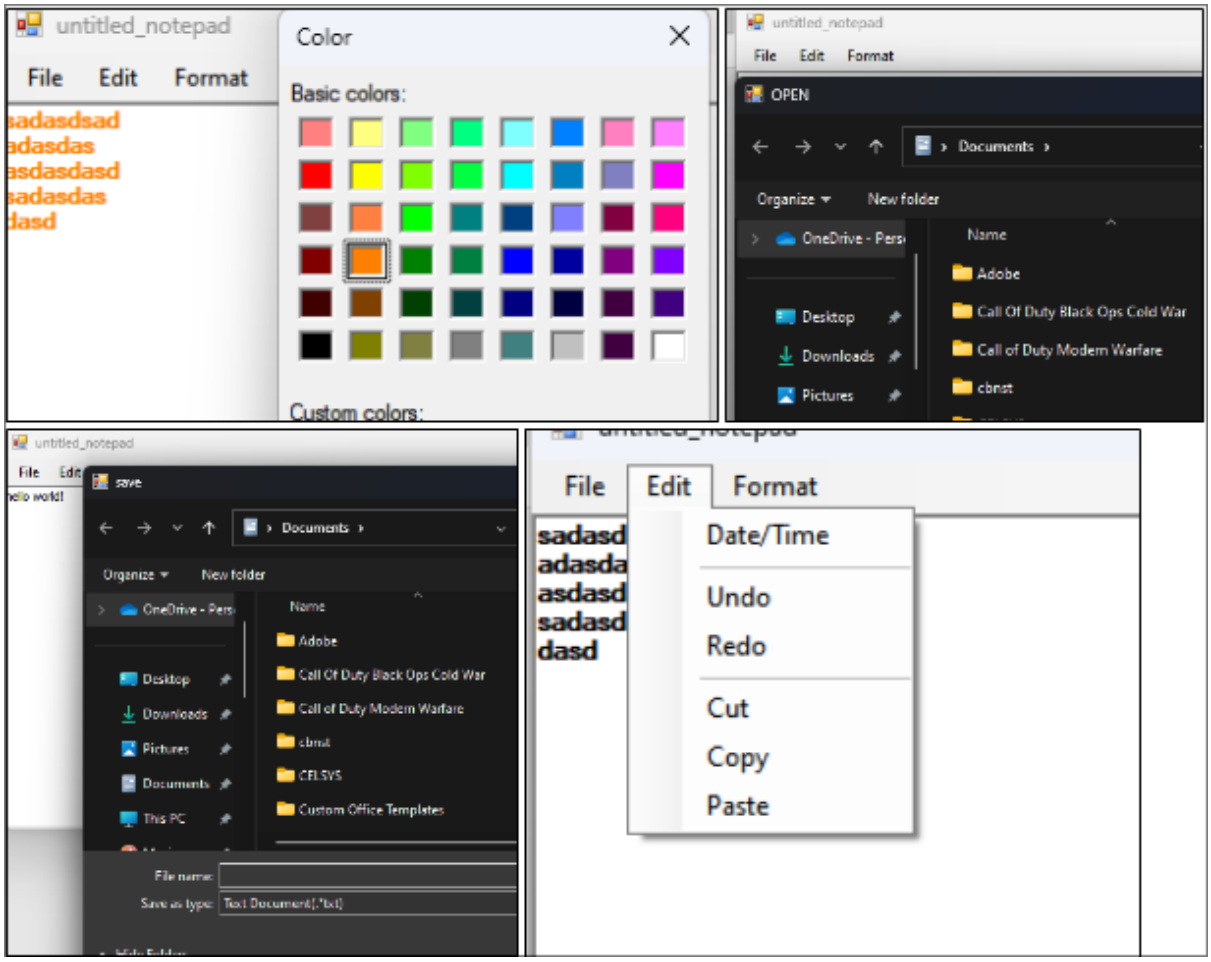

Figure 10: Notepad Demonstration

11 Create a C# Console Application program to implement operator overloading.

Aim

To learn about operator overloading.

Algorithm

1. Start
2. We will initialize three variables x, y, z for class Negate.
3. We create a function to assign values of x,y,x by a,b,c.
4. Then we use Negate operator to make values negative.
5. We will display the values.
6. End

Source Code

```
using System;
namespace OperatorOverloading
{
    class Negate
    {
        private int x, y, z;
        public Negate (int a, int b, int c)
        {
            x = a;
            y = b;
            z = c;
        }
        public void display()
        {
            Console.WriteLine("x={0}\ny={1}\nz={2}", x, y, z);
        }
        public static Negate operator - (Negate c)
        {
            c.x = -c.x;
            c.y = -c.y;
            c.z = -c.z;
            return c;
        }
    }
    class exe
    {
        public static void Main(String []args)
        {
            Negate obj = new Negate(2,-3,4);
            Console.WriteLine("Elements bofore negating:");
            obj.display();
            obj=-obj;
            Console.WriteLine("Elements ofter negating");
```

```
            obj.display();
        }
    }
}
```

Output

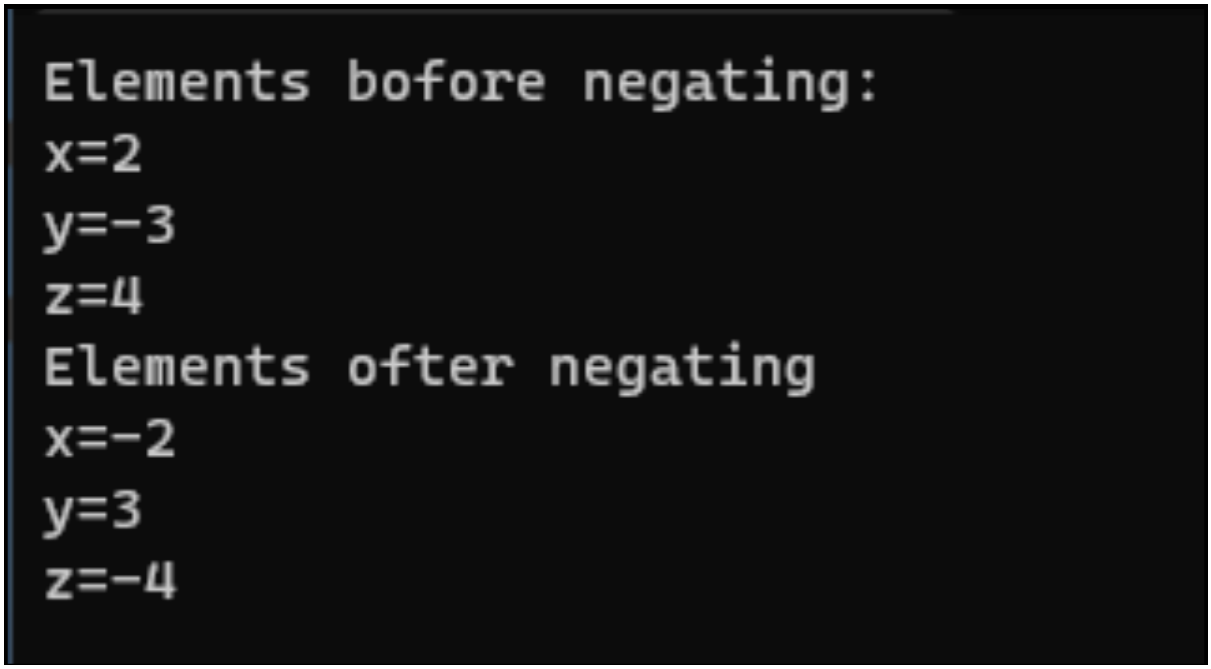

Figure 11: Operator overloading

12 Create a C# program to implement TreeView and ListView Control.

Aim

To make use of Tree view and List View in C#.

Algorithm

1. Start
2. We will place a listView.
3. Initially we will add some items with listView1.Columns.Add() method.
4. Then we will create an object of string[] array as str and read the values.
5. We fead the array in listView with a button.
6. For adding a node we will treeView1.Nodes.Add() method with button.
7. For adding a selecting a node we will treeView1.SelectNodes.Add() method with button.
8. We will create another button number of nodes.
9. End

Source Code

```
using System;
using System.Windows.Forms;

namespace TreeView
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {
            listView1.View = View.Details;
            listView1.GridLines = true;
            listView1.Columns.Add("ROLL", 70);
            listView1.Columns.Add("NAME", 100);
            listView1.Columns.Add("Email", 150);
            listView1.Columns.Add("COURSE", 100);
            listView1.BackColor = Color.LightGreen;
        }

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

```
        {
            string[] str = new string[4];
            str[0] = textBox1.Text;
            str[1] = textBox2.Text;
            str[2] = textBox3.Text;
            str[3] = textBox4.Text;
            listView1.Items.Add(new ListViewItem(str));
        }

        private void button2_Click(object sender, EventArgs e)
        {
            treeView1.Nodes.Add(textBox5.Text);
        }

        private void button3_Click(object sender, EventArgs e)
        {
            treeView1.SelectedNode.Nodes.Add(textBox6.Text);
        }

        private void button4_Click(object sender, EventArgs e)
        {
            MessageBox.Show("total nodes =
            "+treeView1.GetNodeCount(true).ToString(),"node count");
        }
    }
}
```

Output

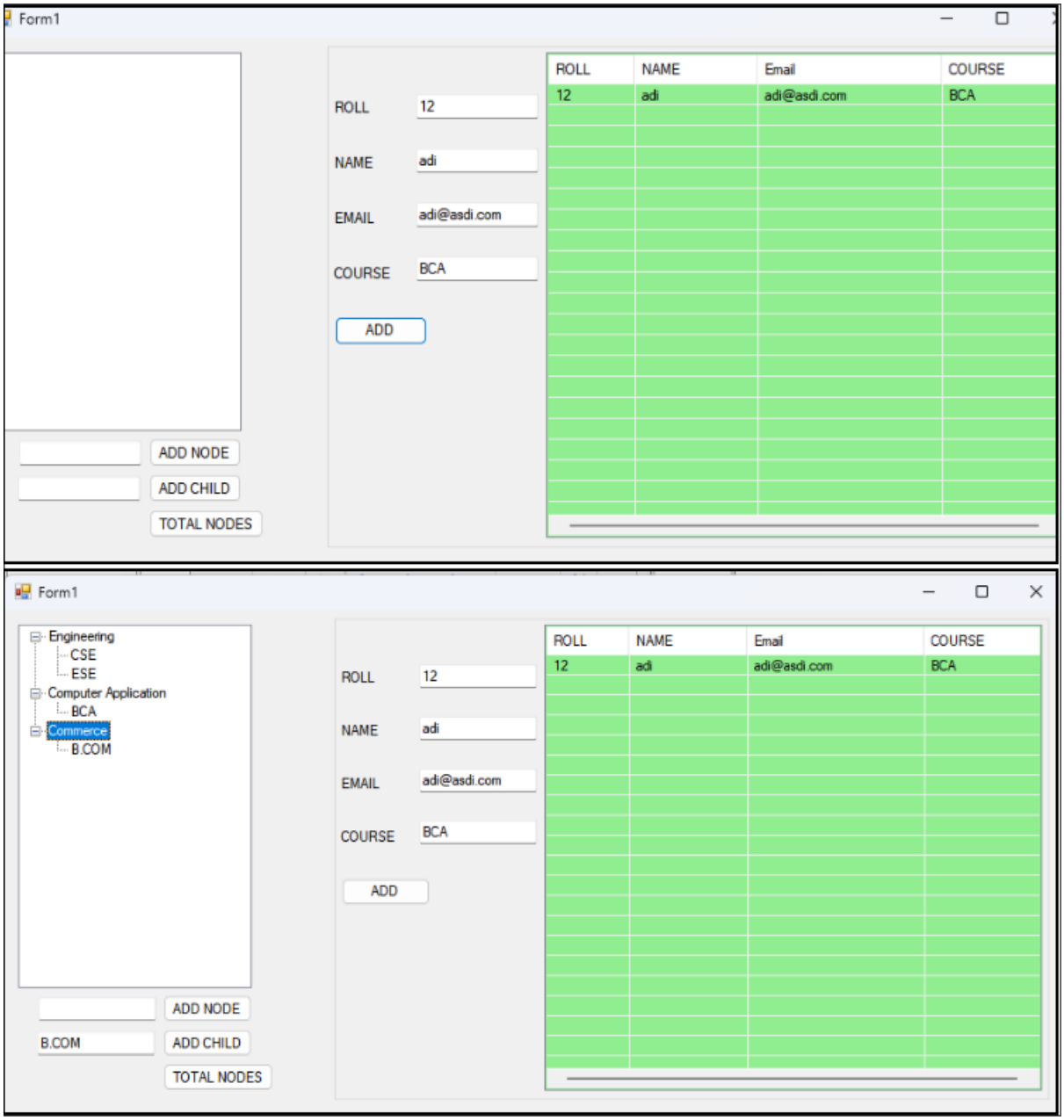

Figure 12: List and Tree View

13 Create a program to create and manipulate a File.

Aim

To implement read and write file.

Algorithm

1. Start
2. We will create three buttons as Read, Write, Cancel.
3. We will StreamReader's object as filereader to read the file.
4. For Write we will StreamWriter's object as filewriter to write the file.
5. For closing the file we will use .Close() method.
6. End

Source Code

```
Imports System.IO
Public Class Form1
    Private Sub cmdread_Click(sender As Object, e As EventArgs)
    Handles cmdread.Click
        Dim filereader As StreamReader
        Dim result As DialogResult
        result = OpenFileDialog1.ShowDialog
        If result = DialogResult.OK Then
            filereader = New StreamReader(OpenFileDialog1.FileName)
            RichTextBox1.Text = filereader.ReadToEnd
            filereader.Close()
        End If
    End Sub
    Private Sub cmdwrite_Click(sender As Object, e As EventArgs)
    Handles cmdwrite.Click
        Dim filewriter As StreamWriter
        Dim result As DialogResult
        result = SaveFileDialog1.ShowDialog
        If result = DialogResult.OK Then
           filewriter = New StreamWriter(SaveFileDialog1.FileName, False)
            filewriter.Write(RichTextBox1.Text)
            filewriter.Close()
        End If
    End Sub
End Class
```

Output

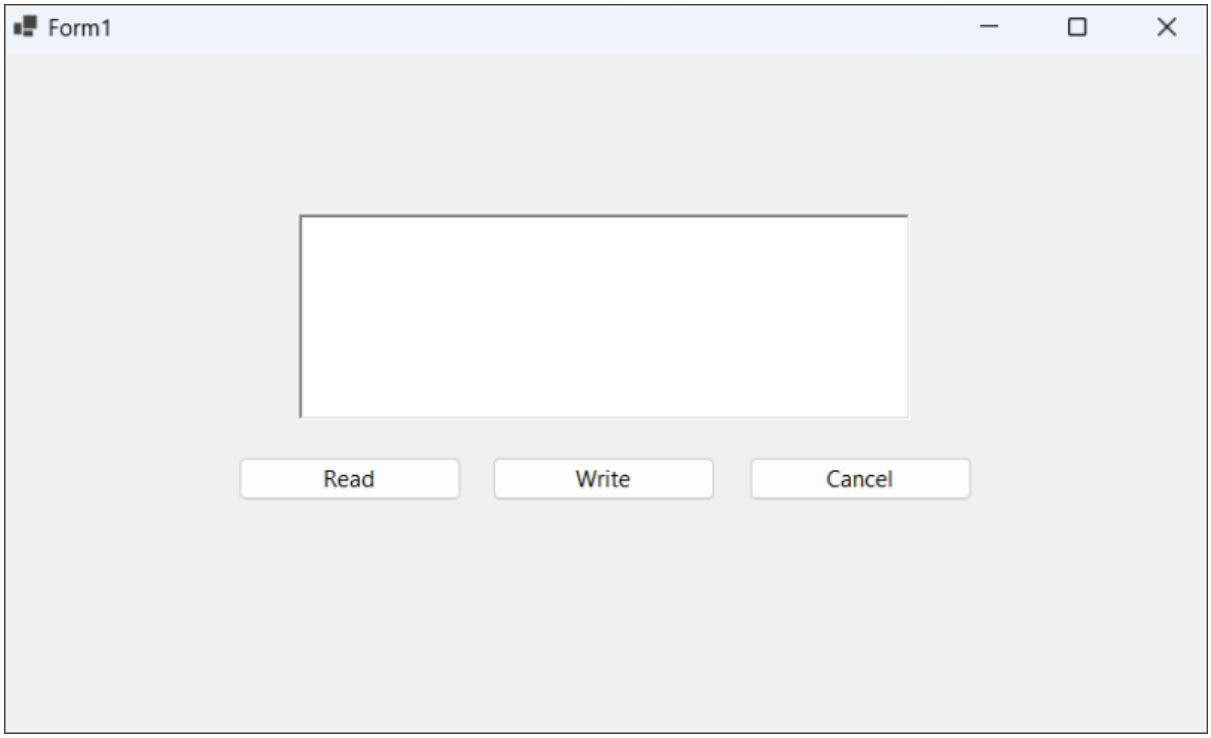

Figure 13: File Manipulation

14 Create a C# Program to implement Employee Information System with Backend.

Aim

To learn data base connectivity in .NET framework.

Algorithm

1. Start
2. We will create objects for ADDOB.Connection, ADODB.Command, OleDb.OleDbConnection as db, cmd, cn
3. We will open our database with by db.Open();
4. We will establish the connection.
5. With the help of cmd.CommandText we insert all the data
6. End

Source Code

```
Public Class Form1
    Dim db As New ADODB.Connection
    Dim cmd As New ADODB.Command
    Dim str, cnstr, sql As String
    Dim cn As OleDb.OleDbConnection

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
            db.Open("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=
           D:\12itu077\employeedb.mdb")
    End Sub

    Private Sub cmdadd_Click(sender As Object, e As EventArgs)
        Handles cmdadd.Click
        str = "insert into employee values('" + Txtempno.Text + "','" +
        Txtname.Text + "','" +
       Txtaddr.Text + "','" + Txtdes.Text + "','" + Txtbp.Text + "','" +
        Txtda.Text + "','" + Txthra.Text + "','" + Txtpf.Text + "','" +
        Txtgp.Text + "','" +
       Txtnp.Text + "')"
        cmd = New ADODB.Command
        cmd.ActiveConnection = db
        cmd.CommandText = str
        cmd.Execute(MsgBox("ADD"))
        cmd.Cancel()
    End Sub

    Private Sub cmddel_Click(sender As Object, e As EventArgs)
        Handles cmddel.Click
        str = "Delete * from employee where empno=" + ComboBox1.Text + ""
        cmd = New ADODB.Command
        cmd.ActiveConnection = db
        cmd.CommandText = str
```

```
        cmd.Execute(MsgBox("RECORD DELETED"))
        cmd.Cancel()
    End Sub

    Private Sub Txtbp_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
   Txtbp.KeyPress
        If Asc(e.KeyChar) = 13 Then
            Txtbp.Text = Val(Txtbp.Text)
            Txtda.Text = Val(Txtbp.Text) * 0.2
            Txthra.Text = Val(Txtbp.Text) * 0.05
            Txtpf.Text = Val(Txtbp.Text) * 0.08
            Txtgp.Text = Val(Txtbp.Text) + Val(Txtda.Text) + Val(Txthra.Text) +
            Val(Txtpf.Text)
            Txtnp.Text = Val(Txtgp.Text) - Val(Txtpf.Text)
        End If
    End Sub
    Private Sub ComboBox1_GotFocus(sender As Object, e As EventArgs) Handles
   ComboBox1.GotFocus
        ComboBox1.Items.Clear()
        cnstr =
        "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:\12itu077\employeedb.mdb"
        cn = New OleDb.OleDbConnection(cnstr)
        cn.Open()
        sql = "select empno from employee"
        Dim ocmd As New OleDb.OleDbCommand(sql, cn)
        Dim odatareader As OleDb.OleDbDataReader = ocmd.executereader
        While odatareader.Read
            ComboBox1.Items.Add(odatareader.GetValue(0).ToString())
        End While
        odatareader.Close()
        cn.Close()
    End Sub
End Class
```

Output

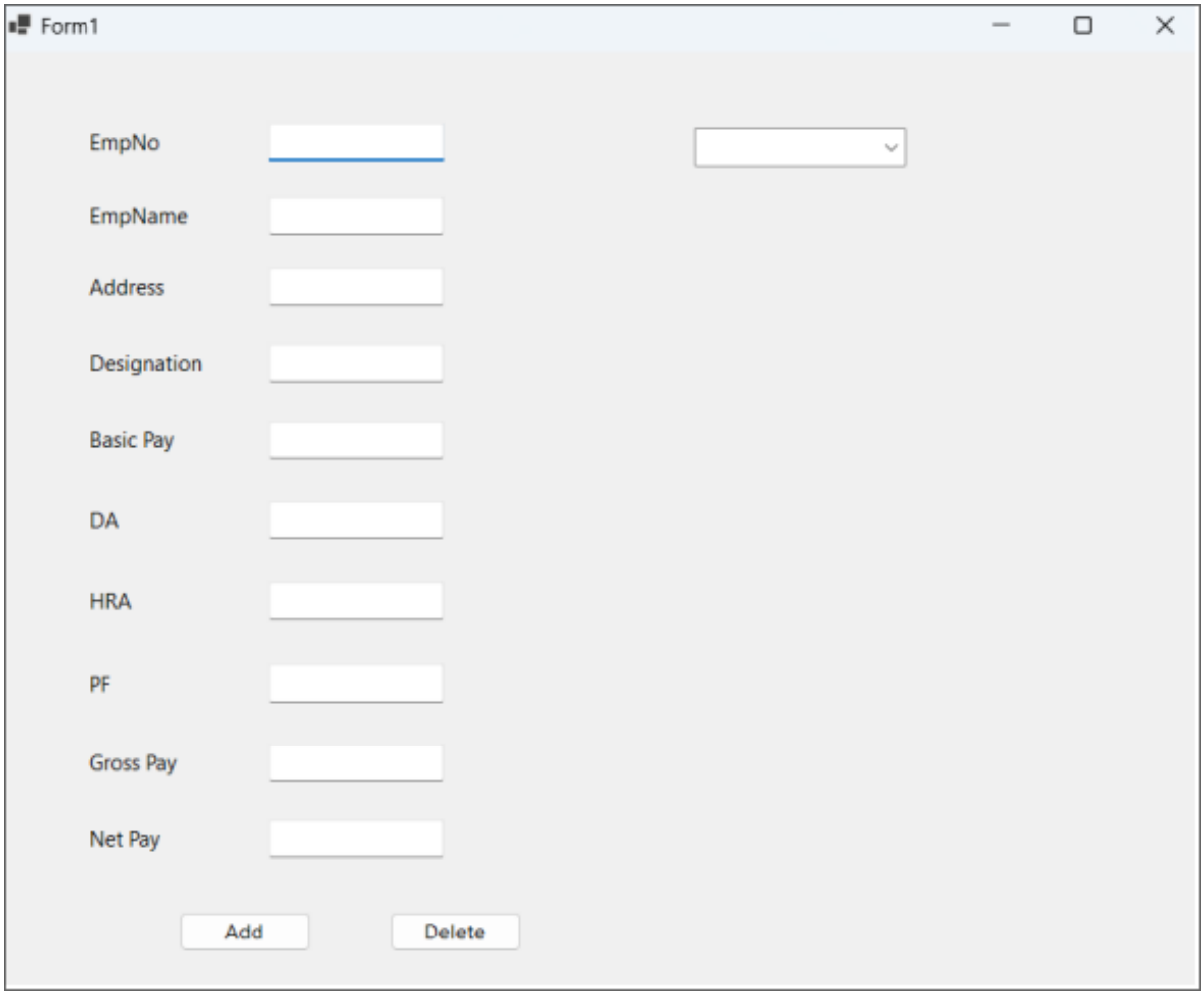

Figure 14: Database

15 Create a C# program to maintain the details of doctors in a hospital with their specialization.

Aim

To learn about database connectivity in .NET framework.

Algorithm

1. Start
2. We will create objects for ADDOB.Connection, ADODB.Command, OleDb.OleDbConnection as db, cmd, cn
3. We will open our database with by db.Open();
4. We will establish the connection.
5. With the help of cmd.CommandText we insert all the data
6. End

Source Code

```
Imports System.Windows.Forms.VisualStyles.VisualStyleElement
Public Class Form1
    Public Class Form1
        Dim db As New ADODB.Connection
        Dim cmd As New ADODB.Command
        Dim str, cnstr, sql As String
        Dim cn As OleDb.OleDbConnection
        Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load
            db.Open("Provider=Microsoft.Jet.OLEDB.4.0;Data
            Source=D:\12itu077\doctor.mdb")
        End Sub
        Private Sub cmddel_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles cmddel.Click
            str = "Delete * from doctable where ID=" + ComboBox1.Text
            cmd = New ADODB.Command
            cmd.ActiveConnection = db
            cmd.CommandText = str
            cmd.Execute(MsgBox("RECORD DELETED"))
            cmd.Cancel()
        End Sub
        Private Sub cmdadd_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles cmdadd.Click
            str = "insert into doctable values('" + Txtname.Text
            + "','" + Txtid.Text + "','" + Txtaddr.Text + "','" + Txtphn.Text
            + "','" + Txtdoj.Text + "','" + Txtbas.Text + "','" + Txtadd.Text
            + "','" + Txtexp.Text + "','" + Txtspl.Text + "','" + Txthos.Text
            + "','" + Txttime.Text + "','" + Txtcladdr.Text + "')"
            cmd.ActiveConnection = db
            cmd.CommandText = str
            cmd.Execute()
            MsgBox("ADD")
```

```
            cmd.Cancel()
        End Sub
        Private Sub ComboBox1_GotFocus(ByVal sender As Object, ByVal e
        As System.EventArgs) Handles ComboBox1.GotFocus
            ComboBox1.Items.Clear()
            cnstr = "Provider=Microsoft.Jet.OLEDB.4.0;Data
            Source=D:\12itu077\doctor.mdb"
            cn = New OleDb.OleDbConnection(cnstr)
            cn.Open()
            sql = "select ID from doctable"
            Dim ocmd As New OleDb.OleDbCommand(sql, cn)
            Dim odatareader As OleDb.OleDbDataReader = ocmd.ExecuteReader
            While odatareader.Read
                ComboBox1.Items.Add(odatareader.GetValue(0).ToString())
            End While
            odatareader.Close()
            cn.Close()
        End Sub
    End Class
End Class
```

Figure 15: Database