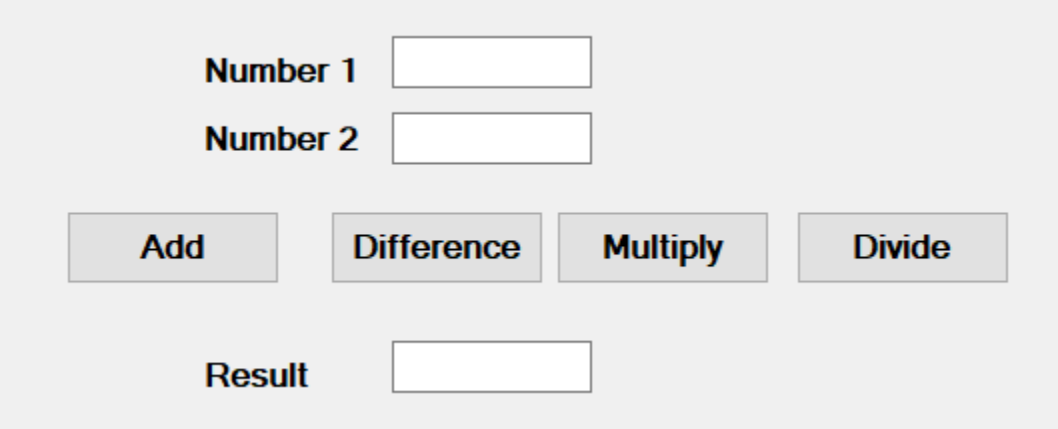
Experiment No . 1

Name : Bilal Mirje **Rollno**: 64 **Div:** A **Batch:** A3

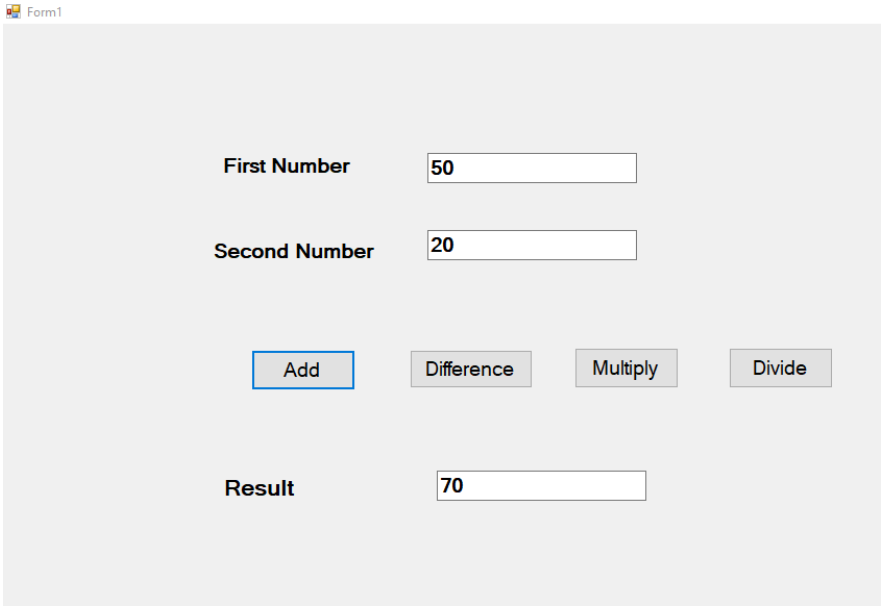
1. Write a single code which will work for all the buttons.




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

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

        private void button1_Click(object sender, EventArgs e)
        {
            int n1, n2, ans = 0;
            Button b = (Button)sender;
            n1 = Convert.ToInt16(textBox1.Text);
            n2 = Convert.ToInt16(textBox2.Text);

            switch (b.Text)
            {

                case "Add" :
                    ans = n1 + n2;
                    break;

                case "Difference":
                    ans = n1 - n2;
                    if (ans < 0)
                        ans *= -1;
                    break;
```

```
                case
                    "Multiply"
                    : ans = n1
                    * n2;
                    break;

                case "Divide":
                    ans = n1 /
                    n2; break;

            }
            textBox3.Text = ans.ToString();

        }

    }
}
```

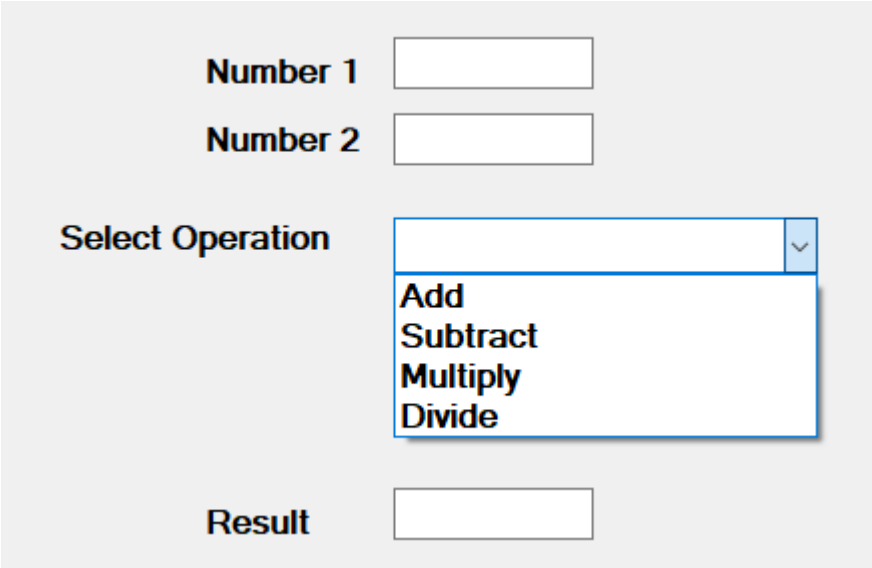
Output :



Experiment No . 2

Name: Bilal Mirje **Rollno:** 64 **Div:** A **Batch:** A3

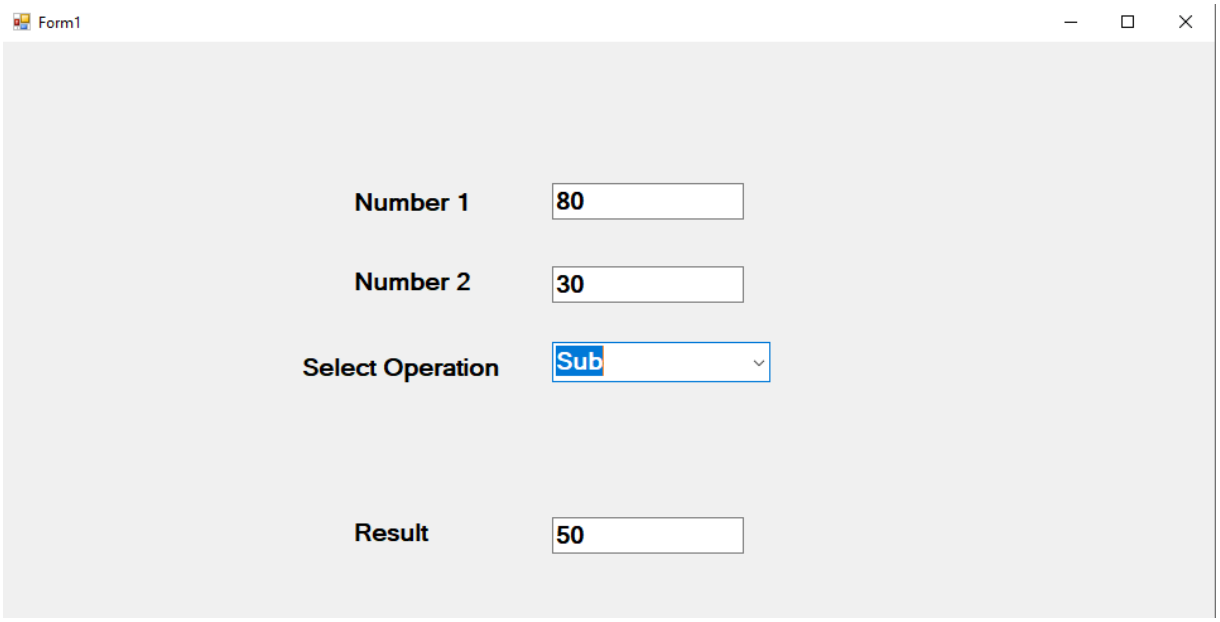
2. Write the code for :




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

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

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            int n1, n2, ans = 0;
            n1 = Convert.ToInt16(textBox1.Text);
            n2 = Convert.ToInt16(textBox2.Text);

            switch (comboBox1.SelectedItem.ToString())
            {
                case "Add":
                    ans = n1 + n2;
                    break;
```

```
                case "Sub":
                    ans = n1 - n2;
                    if (ans < 0)
                        ans *= -1;
                    break;

                case "Mul":
                    ans = n1 * n2;
                    break;

                case "Div":

                    ans = n1 / n2;
                    break;
            }
               textBox3.Text = ans.ToString();
        }


    }
}
```

Output:-

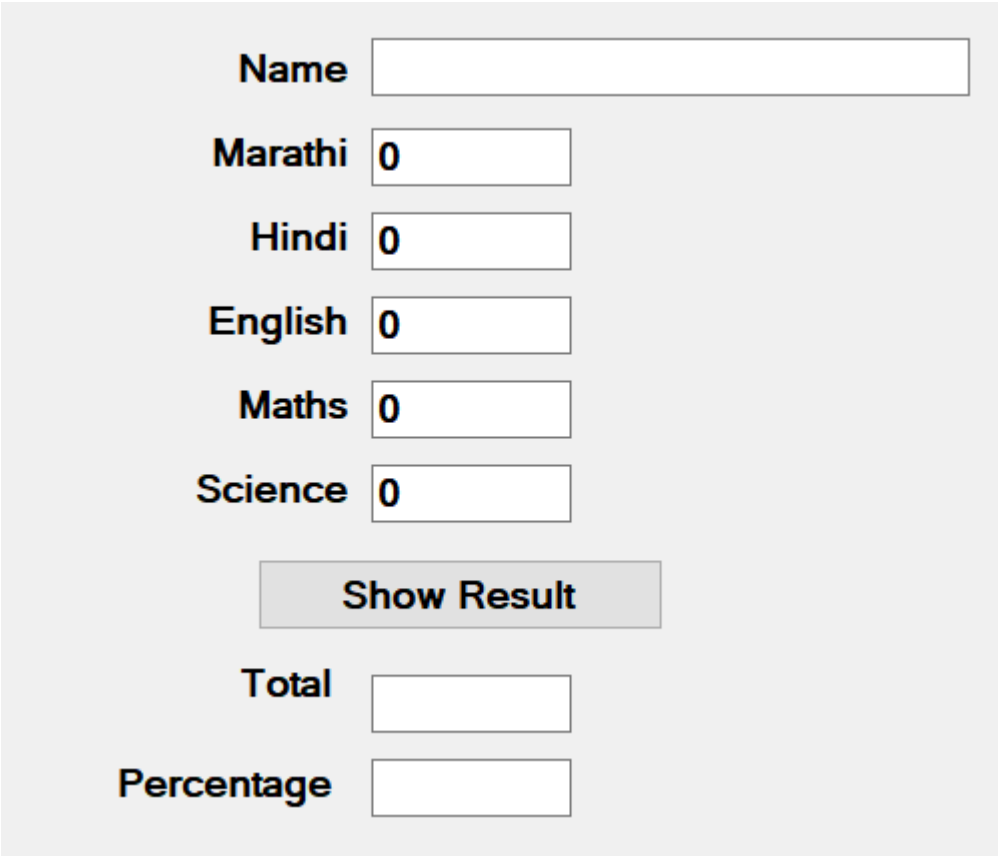


Experiment No. 3

Name : Bilal Mirje **Rollno** : 64 **Div:** A **Batch :** A3

3.Write the code for :

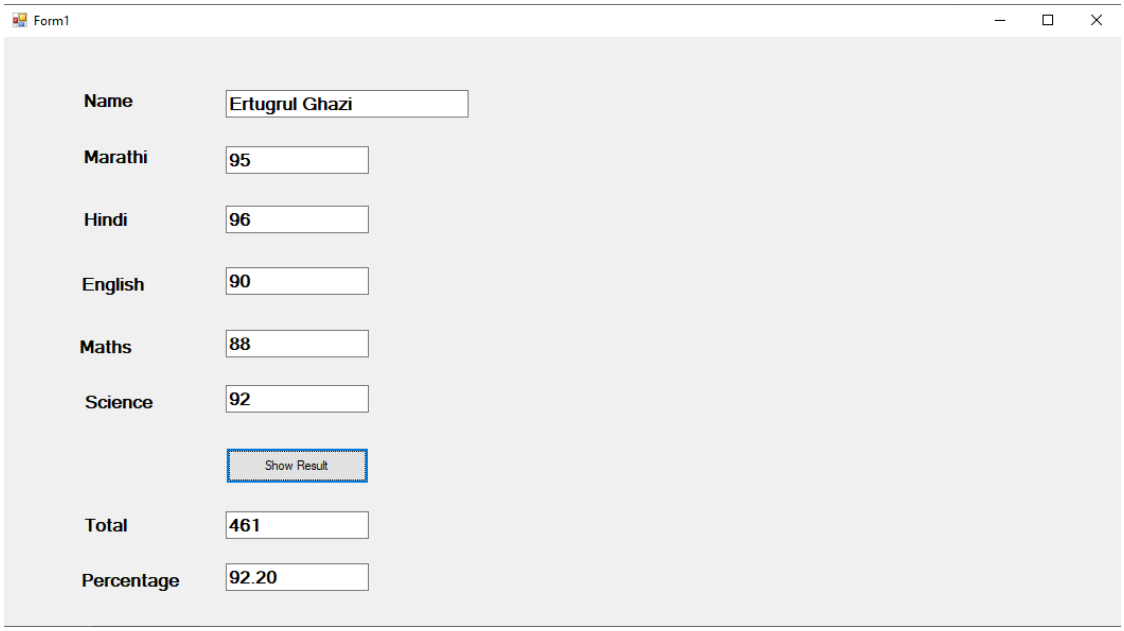
a. **Read name properly (alphabets , blank space and dot)**
b. **Read only numbers/ digits for marks**
c. **Marks must be between 0 to 100**
d. **Calculate total & percentage**




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

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

```
        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            string pattern = @"^[A-Za-z.\s]*$";
            if (!Regex.IsMatch(textBox1.Text, pattern))
            {
                MessageBox.Show("Only letters, spaces, and dots are allowed in the name.", "Invalid
Input", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                textBox1.Text = Regex.Replace(textBox1.Text, @"[^A-Za-z.\s]", "");
                textBox1.SelectionStart = textBox1.Text.Length;

            }
        }

        private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (e.KeyChar == '\b')
                return;
            if (e.KeyChar == '\r')
                SendKeys.Send("{TAB}");

            if (!((e.KeyChar >= '0') & (e.KeyChar <= '9')))
                e.Handled = true;
        }

        private void button1_Click(object sender, EventArgs e)
        {

            if (textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "" || textBox8.Text ==
"" || textBox5.Text == "")
            {
                MessageBox.Show("Please enter marks for all subjects.");
                return;
            }

            int marathi = int.Parse(textBox2.Text);
            int hindi = int.Parse(textBox3.Text);
            int english = int.Parse(textBox4.Text);
            int maths = int.Parse(textBox8.Text);
            int science = int.Parse(textBox5.Text);

            if (!IsValid(marathi) || !IsValid(hindi) || !IsValid(english) || !IsValid(maths) ||
!IsValid(science))
            {
                MessageBox.Show("Marks must be between 0 and 100.");
                return;
            }

                        int total = marathi + hindi + english + maths + science;
            float percentage = total / 5f;

            textBox6.Text = total.ToString();
            textBox7.Text = percentage.ToString("0.00");
        }

        private bool IsValid(int marks)
        {
            return marks >= 0 && marks <= 100;
        }

    }

 }
```

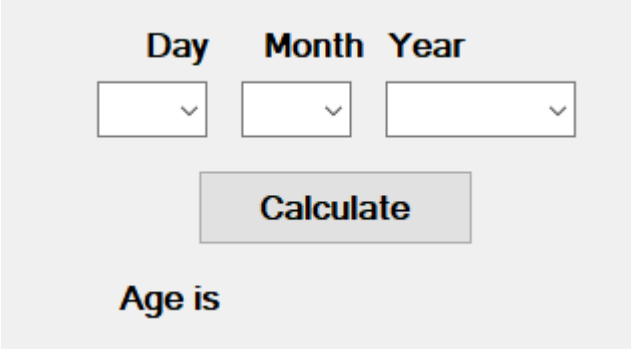
Output:-


Form1
Name
Ertugrul Ghazi
Marathi
95
Hindi
96
English
90
Maths
88
Science
92
Show Result
Total
461
Percentage
92.20

Experiment No. 4

Name : Bilal Mirje **Rollno :** 64 **Div:** A **Batch :** A3

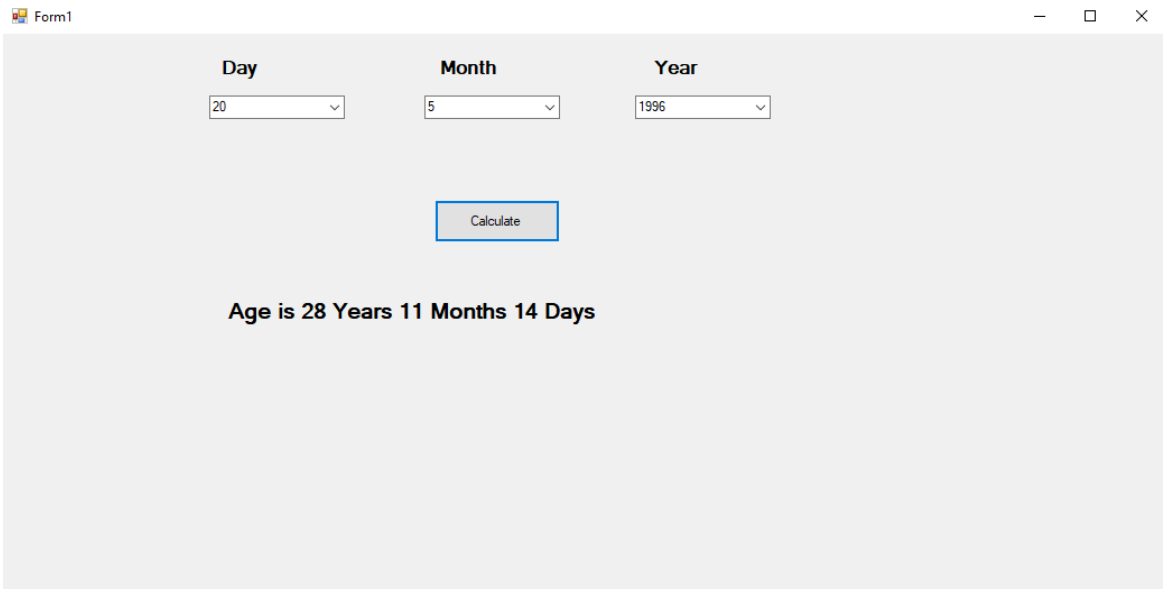
4. Write a program to calculate age :




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

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

        private void Form1_Activated(object sender, EventArgs e)
        {
            comboBox1.Items.Clear();
            comboBox2.Items.Clear();
            comboBox3.Items.Clear();

            for (int n = 1; n <= 31; n++)
                comboBox1.Items.Add(n);

            for (int n = 1; n <= 12; n++)
                comboBox2.Items.Add(n);

            for (int n = 2025; n >= 1945; n--)
                comboBox3.Items.Add(n);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int bd,bm,by,cd,cm,cy,ad,am,ay;
```

```
            bd = Convert.ToInt32(comboBox1.SelectedItem);
            bm = Convert.ToInt32(comboBox2.SelectedItem);
            by = Convert.ToInt32(comboBox3.SelectedItem);

            cd = DateTime.Today.Day;
            cm = DateTime.Today.Month;
            cy = DateTime.Today.Year;

            ad = cd - bd;
            am = cm - bm;
            ay = cy - by;

            if (ad < 0)
            {
                am -= 1;
                ad += 30;

            }
            if (am < 0)
            {
                ay -= 1;
                am += 12;
            }
            label1.Text = " Age is " + ay.ToString() + " Years " + am.ToString() + " Months " +
ad.ToString() + " Days ";

        }


    }
}
```

Output:-

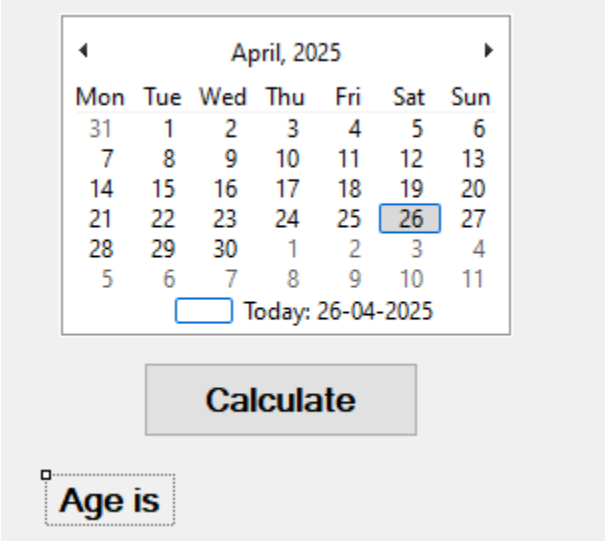


Experiment No. 5

Name : Bilal Mirje **Rollno :** 64 **Div:** A **Batch :** A3

5. Write a program to calculate age by creating class

a. Class must contain properties



//Class Code:-

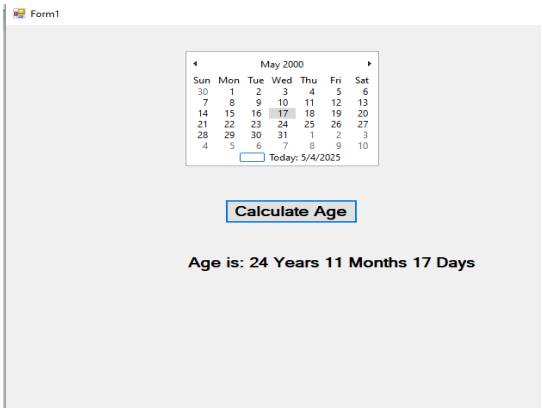
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Agecalculationusingclass
{
    class AgeCalculator
    {
        public int Years { get; private set; }
        public int Months { get; private set; }
        public int Days { get; private set; }

        public AgeCalculator(DateTime birthDate, DateTime currentDate)
        {
            int bd = birthDate.Day;
            int bm = birthDate.Month;
            int by = birthDate.Year;

            int cd = currentDate.Day;
            int cm = currentDate.Month;
            int cy = currentDate.Year;

            Days = cd - bd;
            Months = cm - bm;
            Years = cy - by;
```

```
            if (Days < 0)
            {
                Months -= 1;
                Days += 30; // Approximate month length
            }

            if (Months < 0)
            {
                Years -= 1;

                Months += 12;
            }
        }
    }
}
```

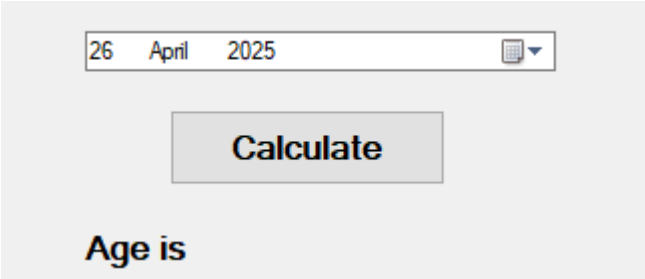
//Main Code:-

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

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

        private void button1_Click(object sender, EventArgs e)
        {
            DateTime birthDate = monthCalendar1.SelectionStart;
            DateTime currentDate = DateTime.Today;

            AgeCalculator age = new AgeCalculator(birthDate, currentDate);
            label1.Text = "Age is: " + age.Years + " Years " + age.Months + " Months " + age.Days + "
Days";
        }
    }
}
```

Output:-



Also for:



//ClassCode:

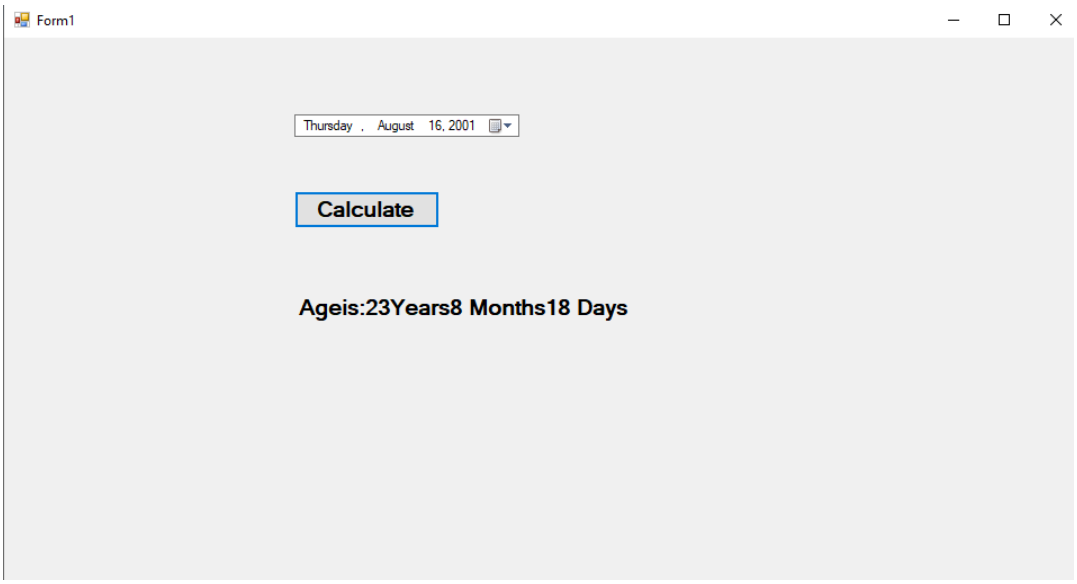
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Agecalculationusingdatetime
{
    class ClassAge
    {
        private string age = "";
        private DateTime dob;
        private int bd, bm, by;

        public DateTime DOB
        {
            set
            {
                dob = value;
            }
        }

        public string AGE
        {
            get
            {
                CalculateAge();
                return age;
            }
        }

        private void CalculateAge()
        {
            int cd = DateTime.Today.Day;
            int cm = DateTime.Today.Month;
            int cy = DateTime.Today.Year;

            bd = dob.Day;
            bm = dob.Month;
            by = dob.Year;

            int ad = cd - bd;
            int am = cm - bm;
            int ay = cy - by;

            if (ad < 0)
            {
                am -= 1;
                ad += DateTime.DaysInMonth(cy, (cm == 1) ? 12 : cm - 1);
            }
```

```
            if (am < 0)
            {
                ay -= 1;
                am += 12;
            }

            age = "Ageis:" + ay.ToString() + "Years" + am.ToString() + " Months" +
            ad.ToString() + " Days";
        }
    }
}
```

//Main Code

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

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

        private void button1_Click(object sender, EventArgs e)
        {
            DateTime dob = dateTimePicker1.Value;
            ClassAge obj = new ClassAge();
            obj.DOB = dob;
            label1.Text = obj.AGE;
        }
    }
}
```

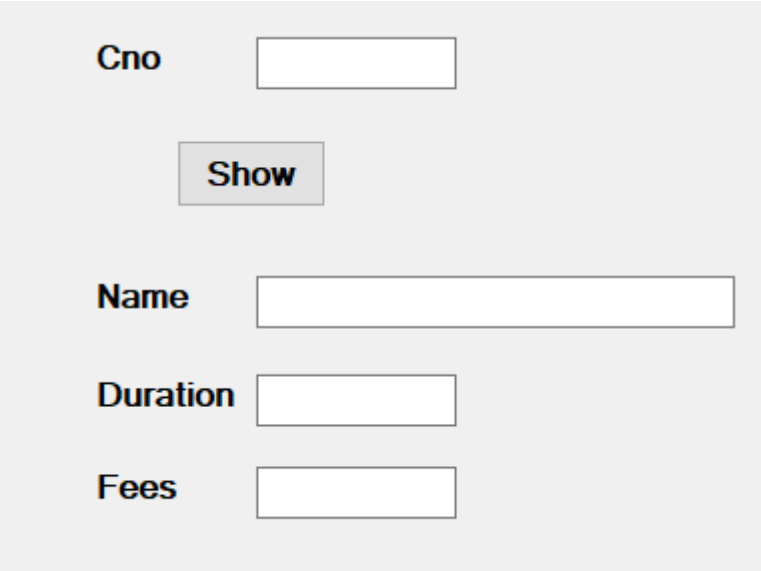
Output:-



Experiment No. 6

Name :- Bilal Mirje Roll no :- 64 Div:- A Batch:- A3

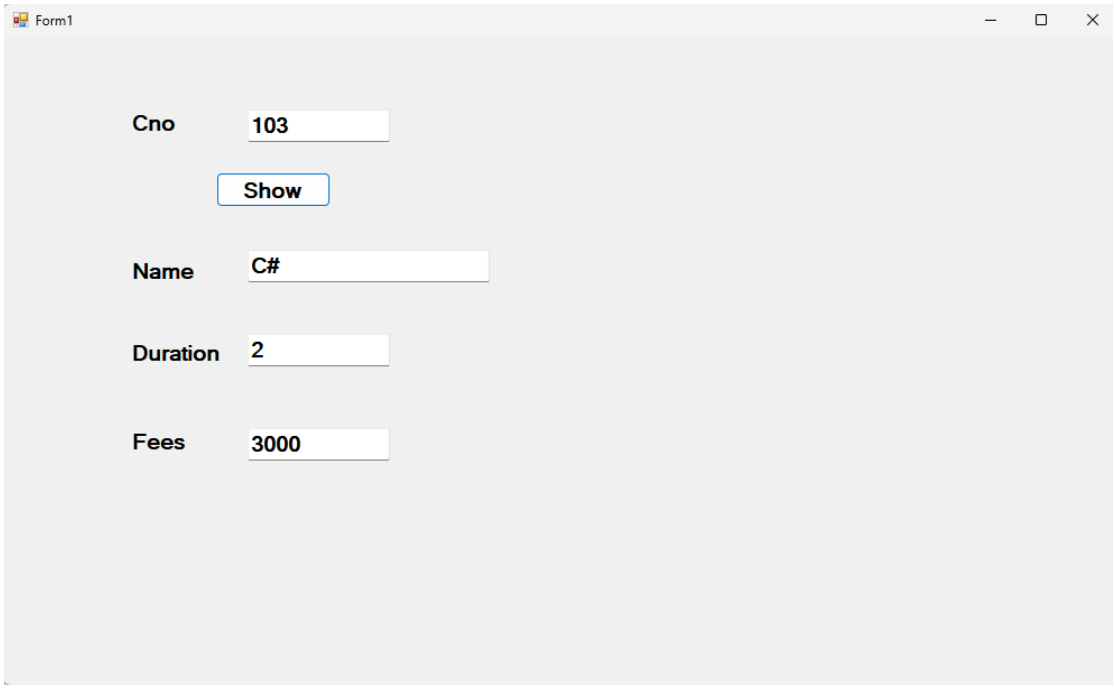
6. Write a program considering “MS – ACCESS” as backend




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
namespace Experiment6
{
    public partial class Form1 : Form
    {
        OleDbConnection cn;
        OleDbCommand co;
        OleDbDataReader dr;
        public Form1()
        {
            InitializeComponent();
        }


        private void button1_Click(object sender, EventArgs e)
        {
            cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\DELL\Documents\schooldb.accdb");
            cn.Open();
```

```
            co = cn.CreateCommand();
            co.CommandText = "SELECT * FROM course WHERE crno=" + textBox1.Text;

            dr = co.ExecuteReader();

            if (dr.HasRows)
            {
                dr.Read();
                textBox2.Text = dr[2].ToString();
                textBox3.Text = dr[3].ToString();
                textBox4.Text = dr[4].ToString();
                dr.Close();
                cn.Close();

            }
            else
            {
                MessageBox.Show("No data found", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                textBox1.Clear();
                textBox2.Clear();
                textBox3.Clear();
                textBox4.Clear();
                textBox1.Focus();
            }
        }


    }
}
```

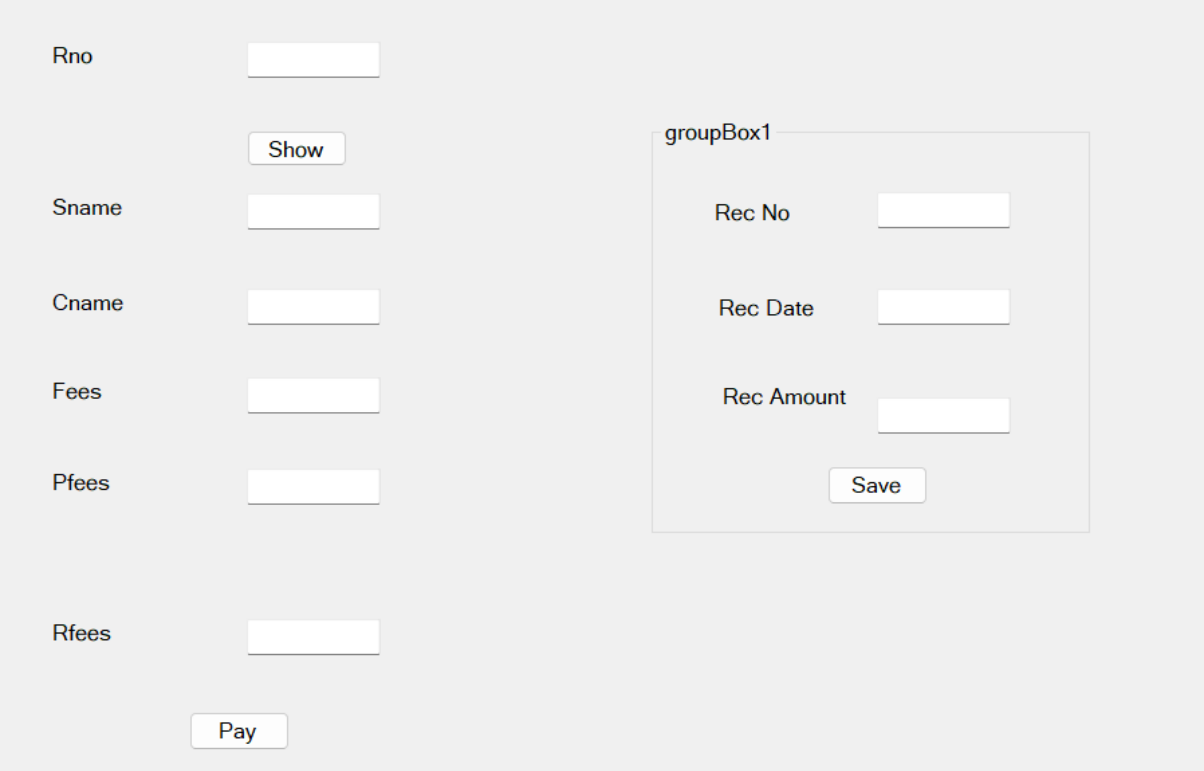
Output:-



Name : Bilal Rafik Mirje **Roll No :** 64 **Batch :** A3

Experiment No : 7

Q. Write a program considering “MS-ACCESS” as a backend.




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
namespace WindowsFormExp71
{
   public partial class Form1 : Form
   {
      OleDbConnection cn;
      OleDbCommand co;
```

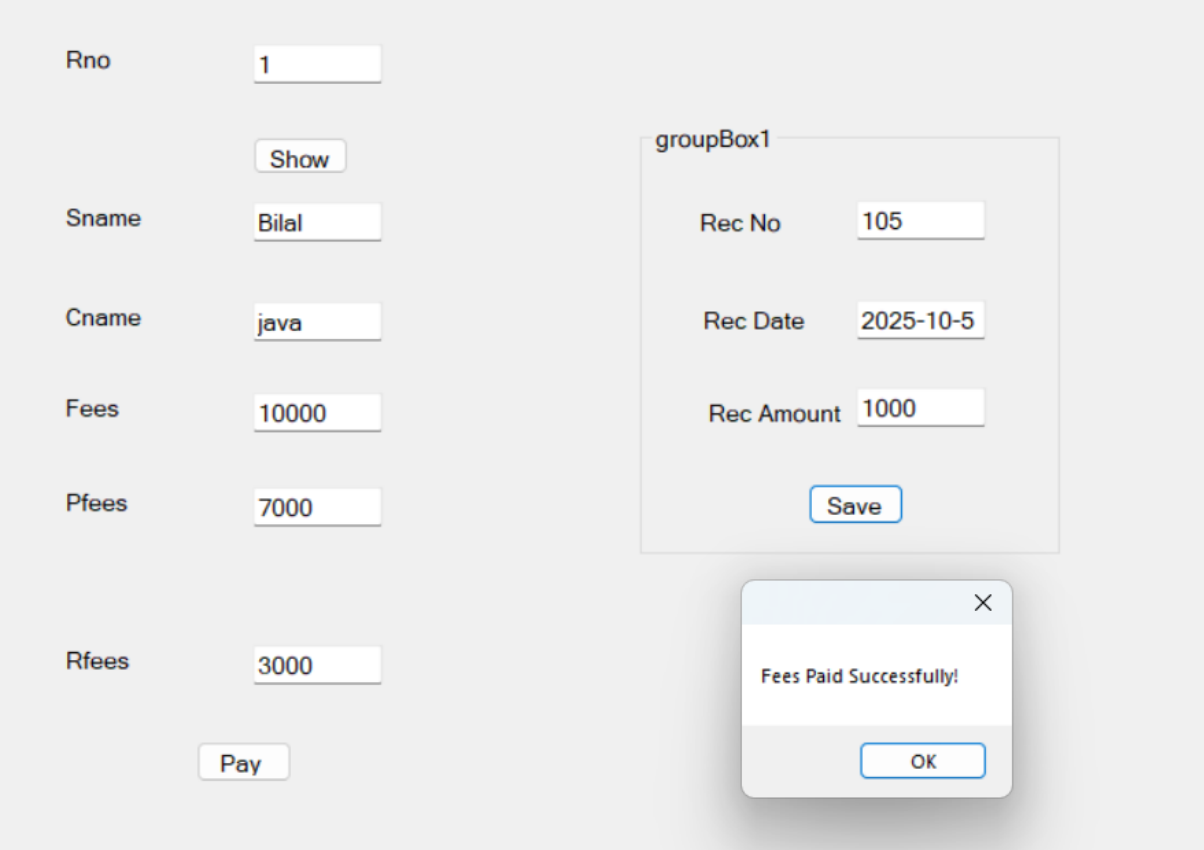
```
OleDbDataReader dr;


public Form1()
{
   InitializeComponent();
}
private void button1_Click(object sender, EventArgs e)
{
   try
   {
      cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;
                  Data Source=C:\Users\mirje\Documents\Database71.accdb");
      cn.Open();
      co = cn.CreateCommand();
     co.CommandText = "SELECT student.rno, student.sname, course.cname, course.fees,
     studentfees.pfees, studentfees.rfees FROM (course INNER JOIN student ON course.cno =
     student.cno) INNER JOIN studentfees ON student.rno = studentfees.rno WHERE
     student.rno=@rno";


      co.Parameters.AddWithValue("@rno", textBox1.Text);
      dr = co.ExecuteReader();
      if (dr.HasRows)
      {
         dr.Read();
         textBox2.Text = dr[1].ToString();
         textBox3.Text = dr[2].ToString();
         textBox4.Text = dr[3].ToString();
         textBox5.Text = dr[4].ToString();
         textBox6.Text = dr[5].ToString();
         dr.Close();


      }
```

```
            else
            {
                MessageBox.Show("No data found", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                textBox1.Clear();
                textBox2.Clear();
                textBox3.Clear();
                textBox4.Clear();
                textBox5.Clear();
                textBox6.Clear();
                textBox1.Focus();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("An error occured : " + ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
        finally
        {
            dr.Close();
            cn.Close();
        }
    }
    private void button2_Click(object sender, EventArgs e)
    {
        groupBox1.Visible = true;
    }
    private void button3_Click(object sender, EventArgs e)
    {
        try
        {
```

```
using (OleDbConnection connection = new OleDbConnection())
{
   connection.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;
                              Data Source=C:\Users\mirje\Documents\Database71.accdb";
   connection.Open();
   int recno = int.Parse(textBox7.Text);
   DateTime recdate = DateTime.Parse(textBox8.Text);
   int recamt = int.Parse(textBox9.Text);
   int rno = int.Parse(textBox1.Text);

   int totalFees = int.Parse(textBox4.Text);
   int oldPaid = int.Parse(textBox5.Text);
   if (oldPaid >= totalFees)
   {
      MessageBox.Show("Fees already fully paid.");
      return;
   }
   int newPaid = oldPaid + recamt;
   int remFees = totalFees - newPaid;

   string insertQuery = "INSERT INTO studentfees (recno, recdate, rno, recamt, pfees, rfees)
            VALUES(?, ?, ?, ?, ?, ?)";
   using (OleDbCommand command = new OleDbCommand(insertQuery, connection))
   {
      command.Parameters.AddWithValue("?", recno);
      command.Parameters.AddWithValue("?", recdate);
      command.Parameters.AddWithValue("?", rno);
      command.Parameters.AddWithValue("?", recamt);
      command.Parameters.AddWithValue("?", newPaid);
      command.Parameters.AddWithValue("?", remFees);
```

```
                int result = command.ExecuteNonQuery();
                if (result > 0)
                {
                    MessageBox.Show("Fees Paid Successfully!");
                    textBox5.Text = newPaid.ToString();
                    textBox6.Text = remFees.ToString();
                    textBox7.Clear();
                    textBox8.Clear();
                    textBox9.Clear();
                }
                else
                {
                    MessageBox.Show("Insertion failed");
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("An error occurred: " + ex.Message);
        }
    }
  }
}
```

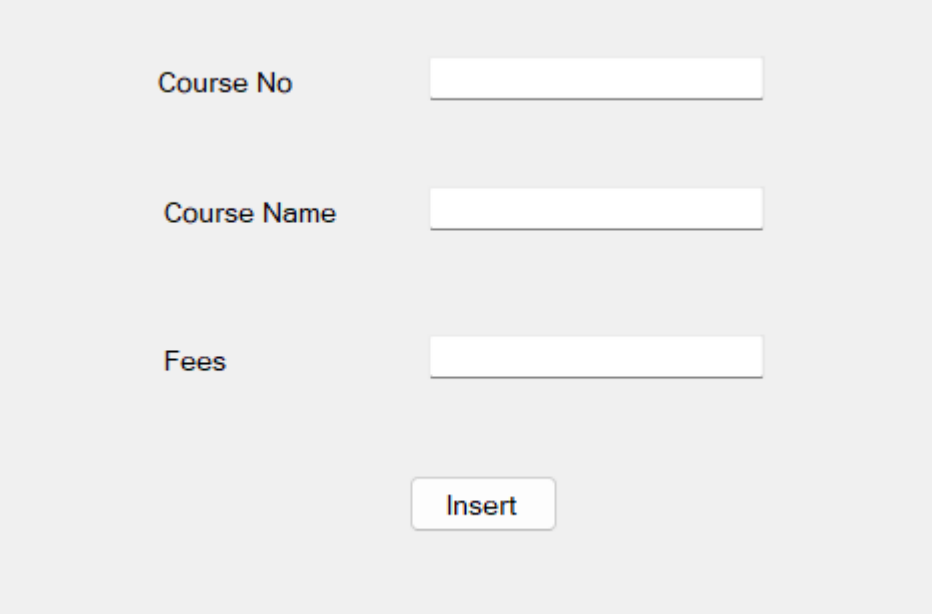
Output :


Rno
1
Show
Sname
Bilal
Cname
java
Fees
10000
Pfees
7000
Rfees
3000
Pay
groupBox1
Rec No
105
Rec Date
2025-10-5
Rec Amount
1000
Save
Fees Paid Successfully!
OK

Name : Bilal Rafik Mirje **Roll No :** 64 **Batch :** A3

Experiment No : 8

Q. Write a program to call stored procedure using “MS-SQL SERVER” as a backend to insert record into course table.



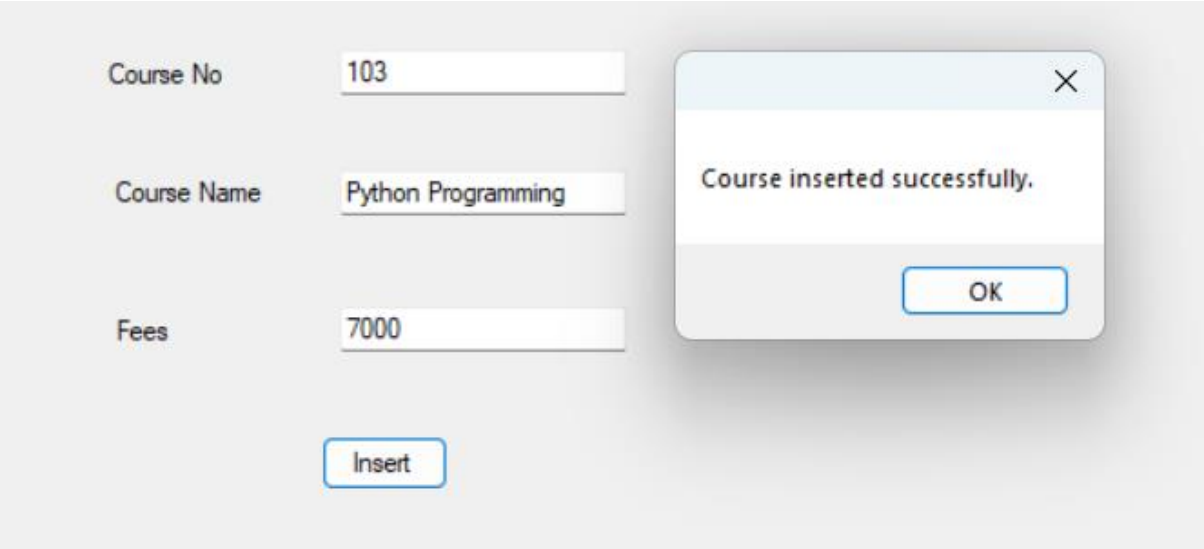

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsFormExp81
{
   public partial class Form1 : Form
   {
      string connectionString = @"Data Source=BILAL\SQLEXPRESS01;Initial Catalog=Exp81;Integrated Security=True";
      public Form1()
      {
```

```
        InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            using (SqlConnection cn = new SqlConnection(connectionString))
            {
                cn.Open();
                using (SqlCommand cmd = new SqlCommand("InsertCourse", cn))
                {
                    cmd.CommandType = CommandType.StoredProcedure;

                    cmd.Parameters.AddWithValue("@cno", int.Parse(textBox1.Text));
                    cmd.Parameters.AddWithValue("@cname", textBox2.Text);
                    cmd.Parameters.AddWithValue("@fees", int.Parse(textBox3.Text));
                    int rows = cmd.ExecuteNonQuery();
                    if (rows > 0)
                        MessageBox.Show("Course inserted successfully.");
                    else
                        MessageBox.Show("Insert failed.");
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error: " + ex.Message);
        }
    }
  }
}
```

Procedure :

```
CREATE PROCEDURE InsertCourse
    @cno INT,
    @cname NVARCHAR(100),
    @fees INT
AS
BEGIN
    INSERT INTO course (cno, cname, fees)
    VALUES (@cno, @cname, @fees);
END
```

Output :



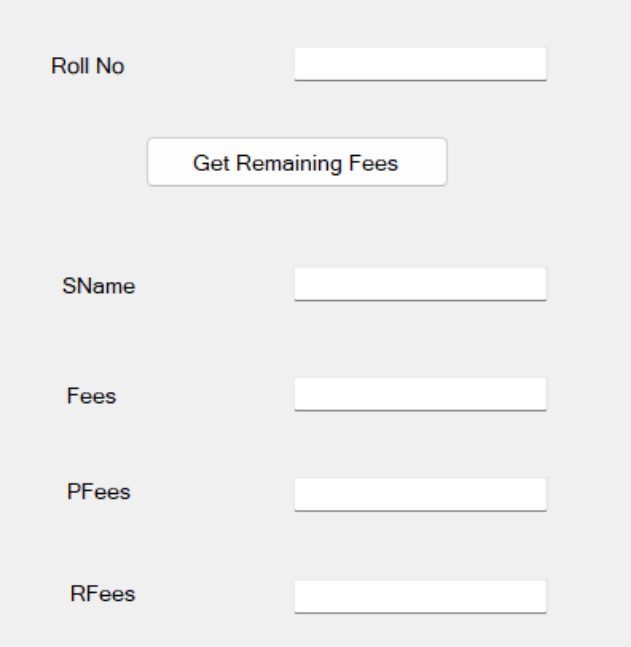
Results Messages

	cno	cname	fees
1	101	Java Development	15000
2	102	cpp	4000
3	103	Python Programming	7000

Name : Bilal Rafik Mirje **Roll No :** 64 **Batch :** A3

Experiment No : 9

Q. Write a program to call stored procedure using “MS-SQL SERVER” as a backend to get remaining fees.




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

namespace WindowsFormsExp9
{
   public partial class Form1 : Form
   {
      string connectionString = @"Data Source=BILAL\SQLEXPRESS01;Initial Catalog=Exp9;Integrated Security=True";
```

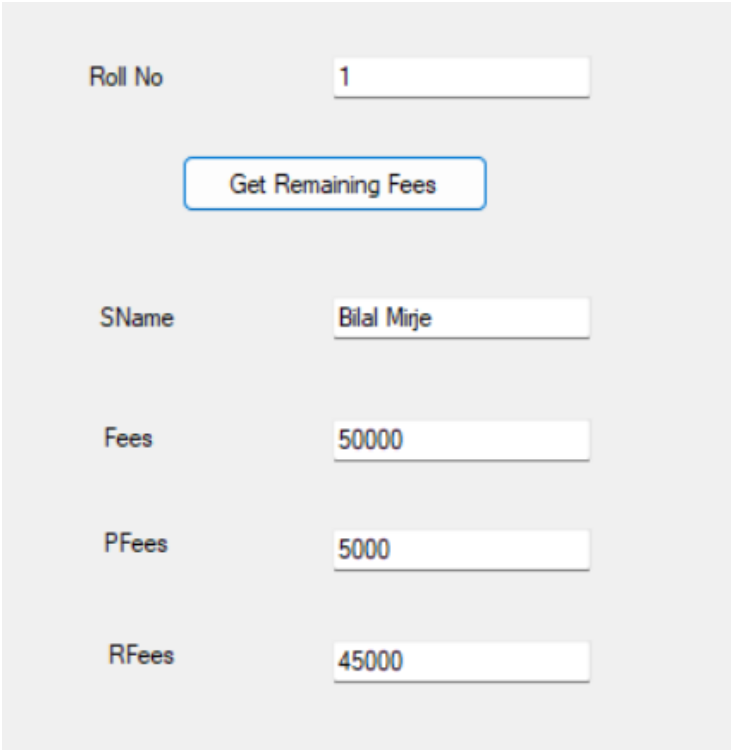
```
public Form1()
{
   InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
   try
   {
      using (SqlConnection cn = new SqlConnection(connectionString))
      {
         cn.Open();
         SqlCommand cmd = new SqlCommand("GetRemainingFees", cn);
         cmd.CommandType = CommandType.StoredProcedure;

         cmd.Parameters.AddWithValue("@rno", int.Parse(textBox1.Text));

         SqlDataReader dr = cmd.ExecuteReader();
         if (dr.Read())
         {
            textBox2.Text = dr["sname"].ToString();
            textBox3.Text = dr["cfees"].ToString();
            textBox4.Text = dr["pfees"].ToString();
            textBox5.Text = dr["rfees"].ToString();
         }
         else
         {
            MessageBox.Show("Record not found.");
         }
         dr.Close();
      }
```

```
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error: " + ex.Message);
        }
      }
   }
}
```

PROCEDURE :

```
CREATE PROCEDURE GetRemainingFees
   @rno INT
AS
BEGIN
   SELECT rno, sname, cfees, pfees, (cfees - pfees) AS rfees
   FROM studentfees
   WHERE rno = @rno;
END;
```

Output :



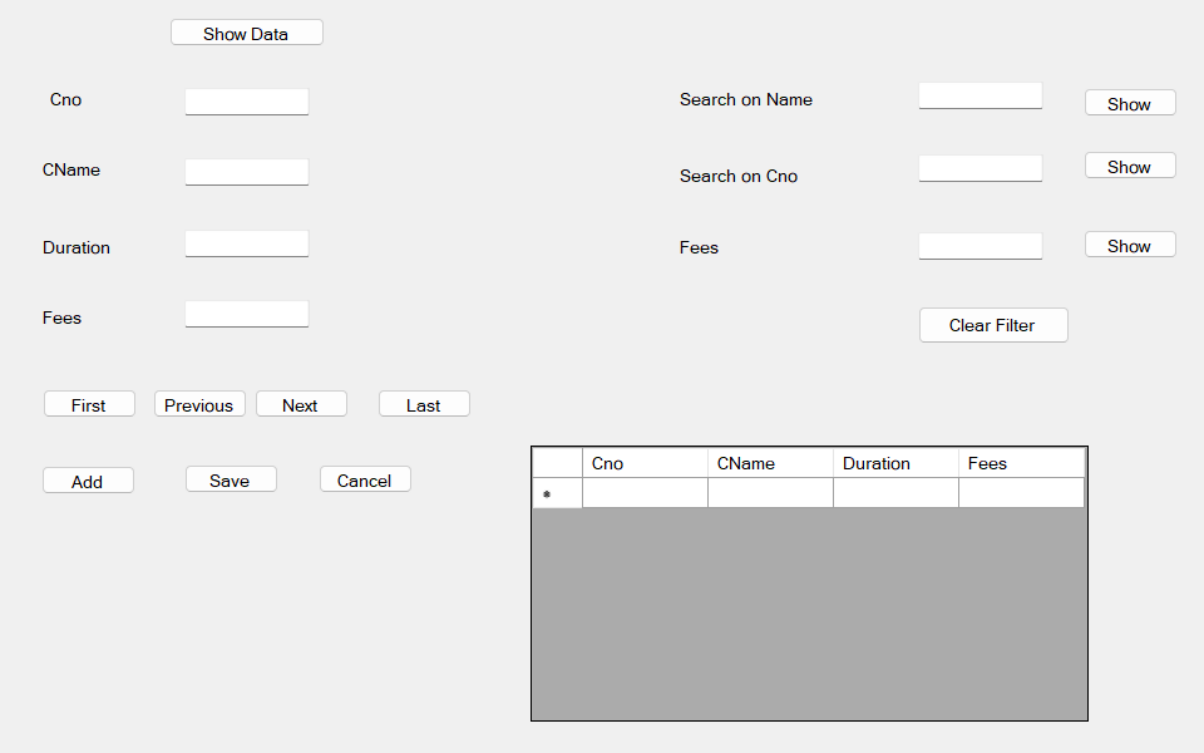
Results Messages

	rno	sname	cfees	pfees
1	1	Bilal Mirje	50000	5000
2	2	Kaif Mulla	60000	60000
3	3	Maaj	45000	30000

Name : Bilal Rafik Mirje **Roll No :** 64 **Batch :** A3

Experiment No : 10

Q. Write a program for




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormExp10
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
```

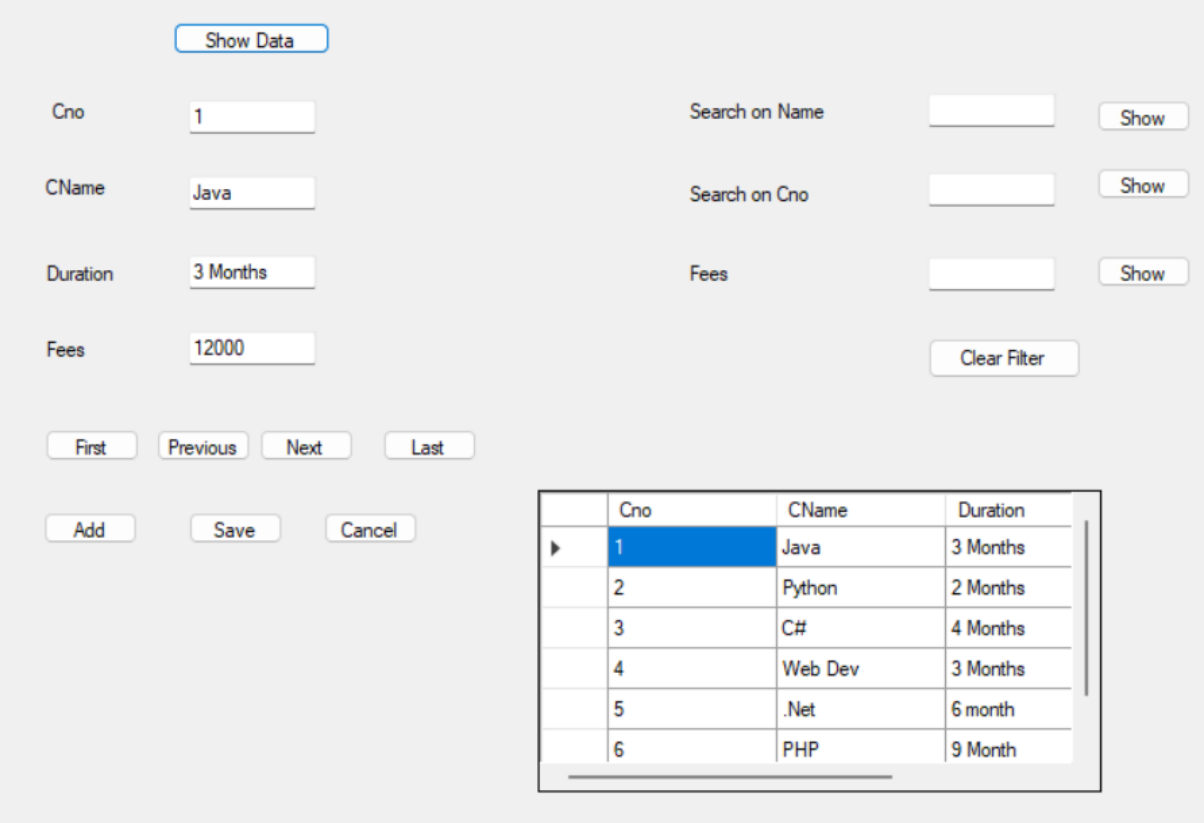
```
private void Form1_Load(object sender, EventArgs e)
{
   textBox1.DataBindings.Clear();
   textBox2.DataBindings.Clear();
   textBox3.DataBindings.Clear();
   textBox4.DataBindings.Clear();

   textBox1.DataBindings.Add("Text", courseBindingSource, "Cno");
   textBox2.DataBindings.Add("Text", courseBindingSource, "CName");
   textBox3.DataBindings.Add("Text", courseBindingSource, "Duration");
   textBox4.DataBindings.Add("Text", courseBindingSource, "Fees");

}
private void button6_Click(object sender, EventArgs e)
{
   courseBindingSource.MoveLast();
   int cno = Convert.ToInt16(textBox1.Text) + 1;
   courseBindingSource.AddNew();
   textBox1.Text = cno.ToString();
   textBox2.Focus();
}
private void button7_Click(object sender, EventArgs e)
{
   try
   {
      this.Validate();
      courseBindingSource.EndEdit();
      courseTableAdapter.Update(dataSet1.Course);
      MessageBox.Show("Record Saved Successfully");
   }
```

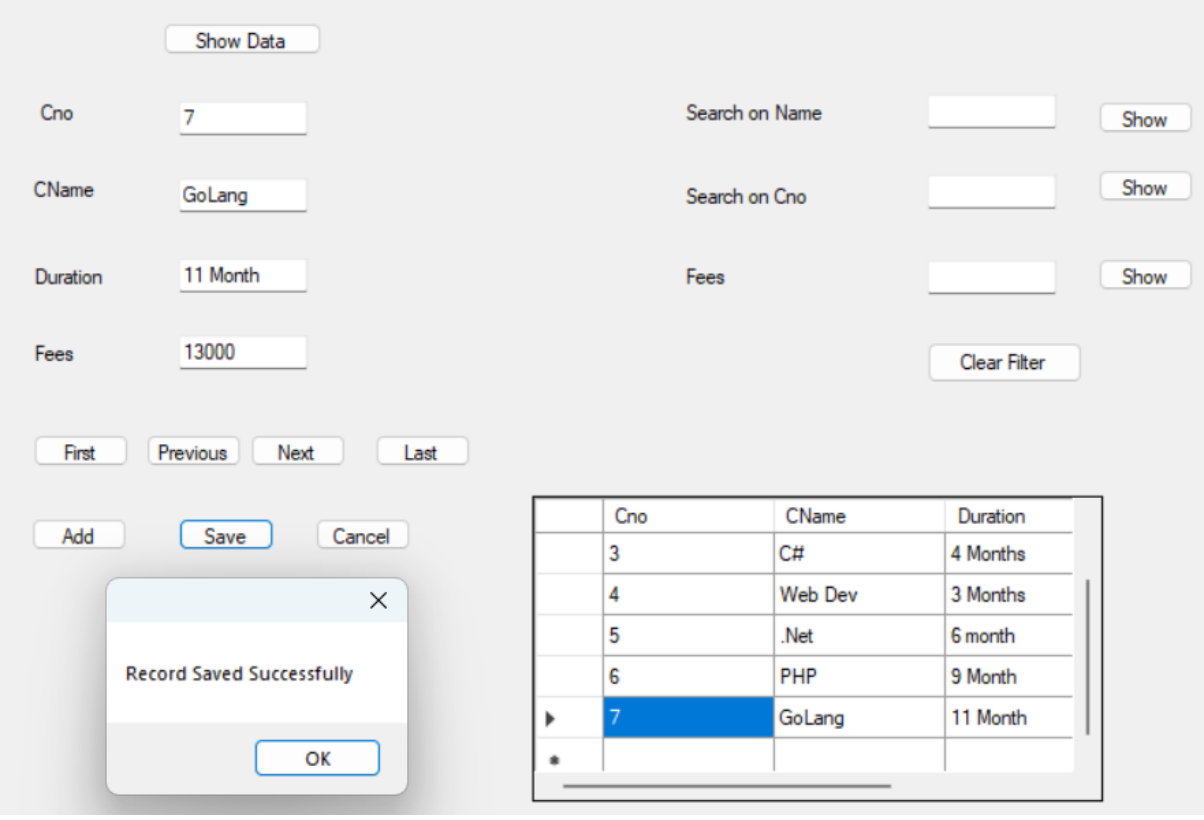
```
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
}
private void button8_Click(object sender, EventArgs e)
{
    courseBindingSource.CancelEdit();
}
private void button2_Click(object sender, EventArgs e)
{
    courseBindingSource.MoveFirst();
}
private void button3_Click(object sender, EventArgs e)
{
    courseBindingSource.MovePrevious();
}
private void button4_Click(object sender, EventArgs e)
{
    courseBindingSource.MoveNext();
}
private void button5_Click(object sender, EventArgs e)
{
    courseBindingSource.MoveLast();
}
private void button1_Click(object sender, EventArgs e)
{
    courseTableAdapter.Fill(dataSet1.Course);
}
private void button9_Click(object sender, EventArgs e)
{
```

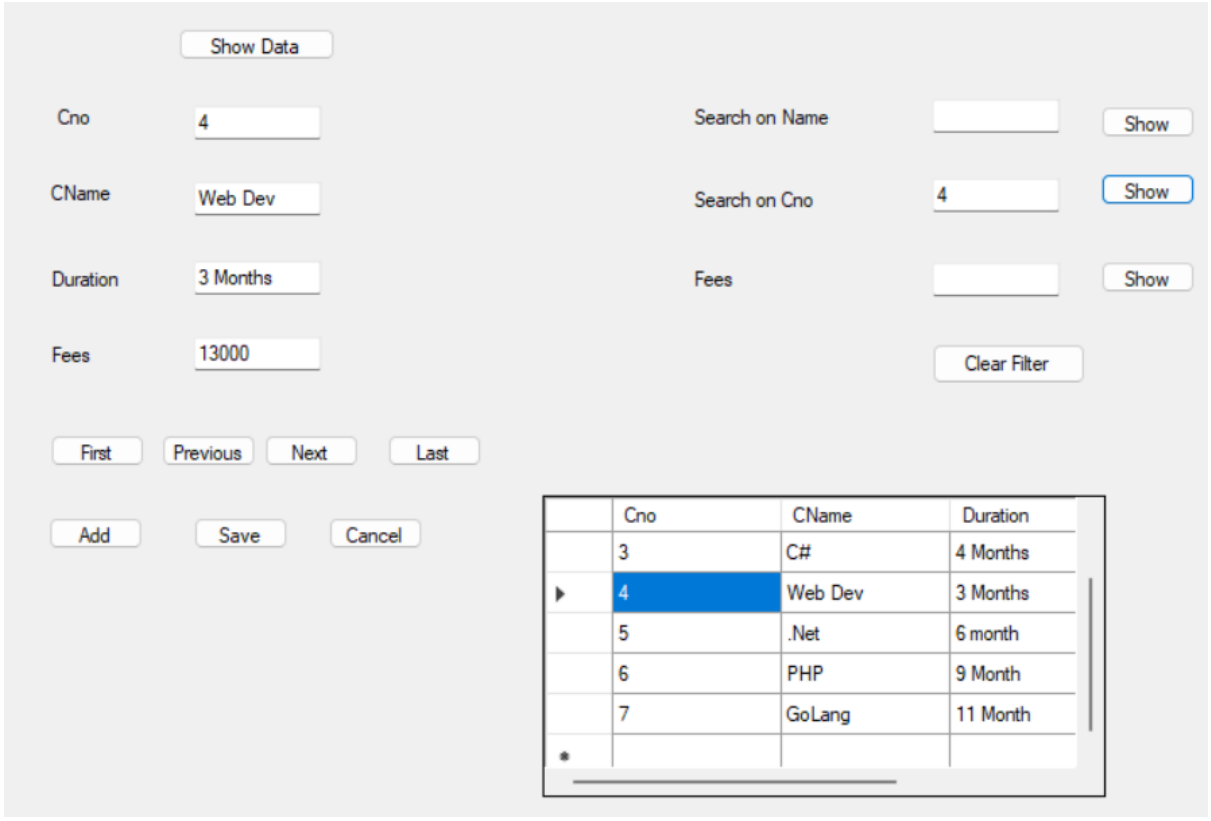
```
    int pos = courseBindingSource.Find("Cno",textBox6.Text);
    if (pos == -1)
    {
        MessageBox.Show("No such record");
        textBox6.Clear();
    }
    else
    {
        courseBindingSource.Position = pos;
    }
}
private void button11_Click(object sender, EventArgs e)
{
    int pos = courseBindingSource.Find("CName", textBox5.Text);
    if (pos == -1)
    {
        MessageBox.Show("No such record");
        textBox5.Clear();
    }
    else
    {
        courseBindingSource.Position = pos;
    }
}

private void button12_Click(object sender, EventArgs e)
{
    int pos = courseBindingSource.Find("Fees", textBox7.Text);
    if (pos == -1)
    {
        MessageBox.Show("No such record");
```

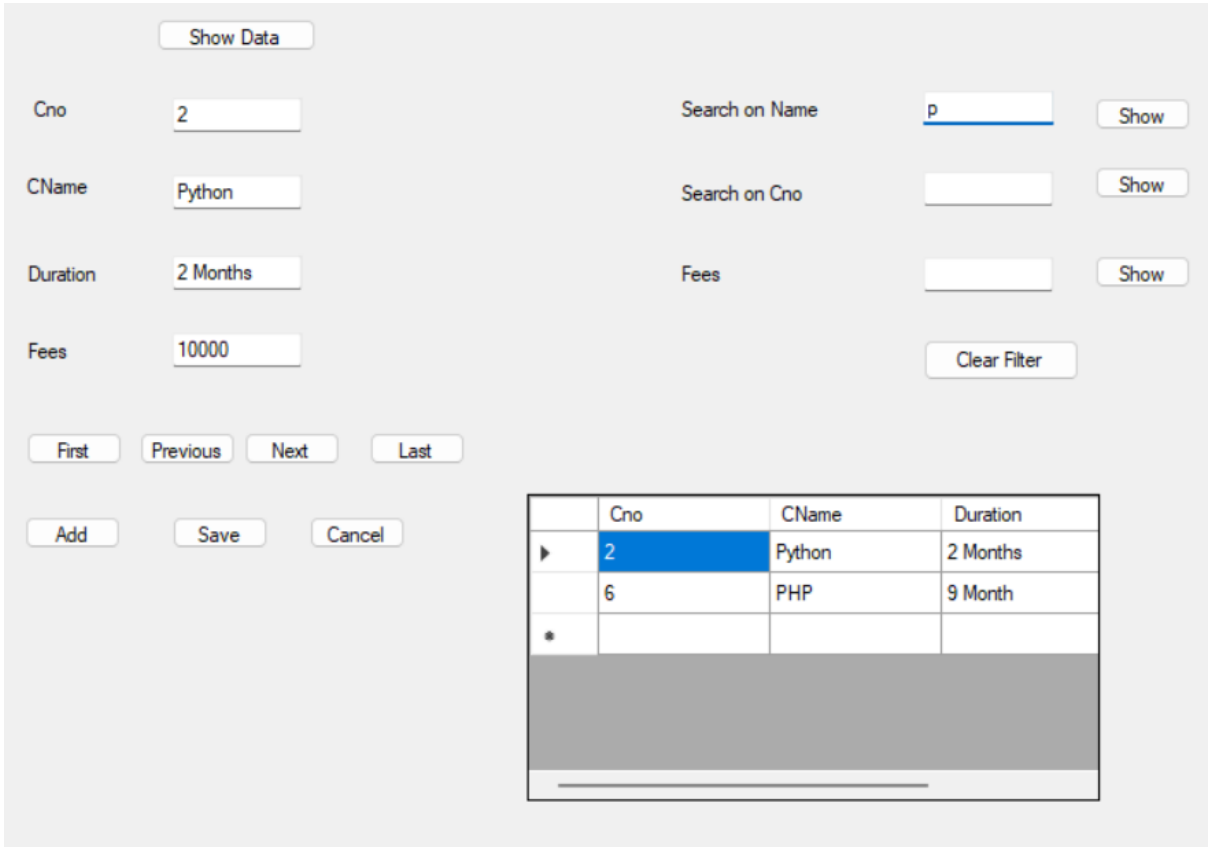
```
            textBox7.Clear();
        }
        else
        {
            courseBindingSource.Position = pos;
        }
    }
    private void textBox5_TextChanged(object sender, EventArgs e)
    {
        courseBindingSource.Filter = "CName like '" + textBox5.Text + "%'";
    }
    private void textBox7_TextChanged(object sender, EventArgs e)
    {
        courseBindingSource.Filter = "Fees >= " + textBox7.Text;
    }
    private void button10_Click(object sender, EventArgs e)
    {
        courseBindingSource.RemoveFilter();
    }

  }
}
```

Output :


Show Data
Cno
1
CName
Java
Duration
3 Months
Fees
12000
First
Previous
Next
Last
Add
Save
Cancel
Search on Name
Show
Search on Cno
Show
Fees
Show
Clear Filter
Cno
CName
Duration
1
Java
3 Months
2
Python
2 Months
3
C#
4 Months
4
Web Dev
3 Months
5
.Net
6 month
6
PHP
9 Month


Show Data
Cno
7
CName
GoLang
Duration
11 Month
Fees
13000
First
Previous
Next
Last
Add
Save
Cancel
Record Saved Successfully
OK
Search on Name
Show
Search on Cno
Show
Fees
Show
Clear Filter
Cno
CName
Duration
3
C#
4 Months
4
Web Dev
3 Months
5
.Net
6 month
6
PHP
9 Month
7
GoLang
11 Month

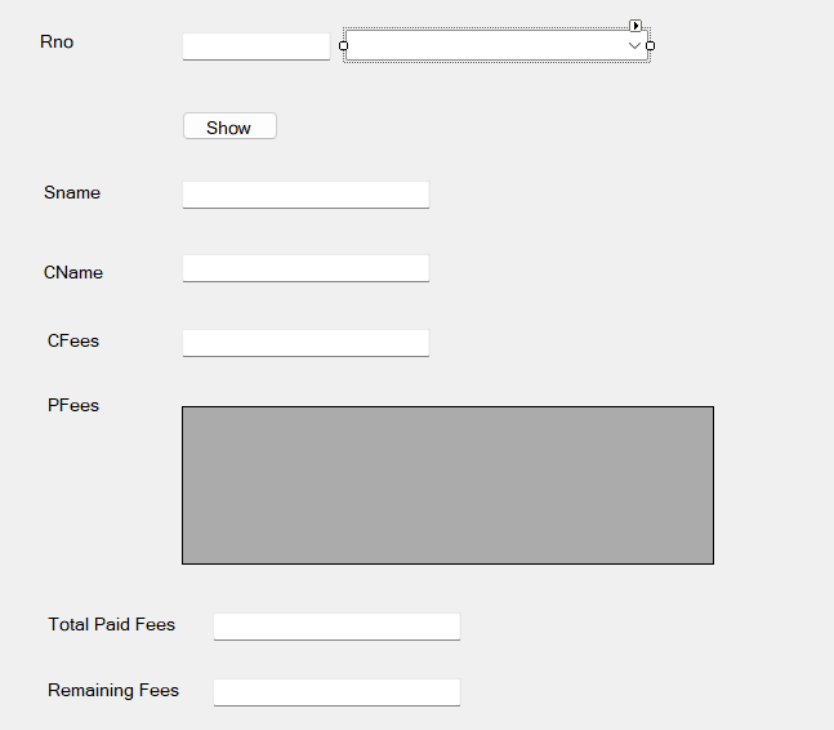

Show Data
Cno
4
CName
Web Dev
Duration
3 Months
Fees
13000
First
Previous
Next
Last
Add
Save
Cancel
Search on Name
Show
Search on Cno
4
Show
Fees
Show
Clear Filter
Cno
CName
Duration
3
C#
4 Months
4
Web Dev
3 Months
5
.Net
6 month
6
PHP
9 Month
7
GoLang
11 Month


Show Data
Cno
2
CName
Python
Duration
2 Months
Fees
10000
First
Previous
Next
Last
Add
Save
Cancel
Search on Name
p
Show
Search on Cno
Show
Fees
Show
Clear Filter
Cno
CName
Duration
2
Python
2 Months
6
PHP
9 Month

Name : Bilal Rafik Mirje **Roll No :** 64 **Batch :** A3

Experiment No : 11

Q. Write a code for given form use MySql as a Backend




```
using System;
using System.Data;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace WindowsFormExp11
{
   public partial class Form1 : Form
   {
      string connStr = "server=localhost;user=root;database=studentdb;";

      public Form1()
      {
         InitializeComponent();
      }
```

```
private void Form1_Load(object sender, EventArgs e)
{
   try
   {
      using (MySqlConnection conn = new MySqlConnection(connStr))
      {
         conn.Open();
         MySqlCommand cmd = new MySqlCommand("SELECT Rno FROM student_fees", conn);
         MySqlDataReader dr = cmd.ExecuteReader();
         while (dr.Read())
         {
            comboBox1.Items.Add(dr["Rno"].ToString());
         }
      }
   }
   catch (Exception ex)
   {
      MessageBox.Show("Error loading Rno values: " + ex.Message);
   }
}

private void button1_Click(object sender, EventArgs e)
{
   int rno = 0;
   if (!string.IsNullOrWhiteSpace(comboBox1.Text))
      int.TryParse(comboBox1.Text, out rno);
   else if (!string.IsNullOrWhiteSpace(textBox1.Text))
      int.TryParse(textBox1.Text, out rno);

   if (rno == 0)
   {
```

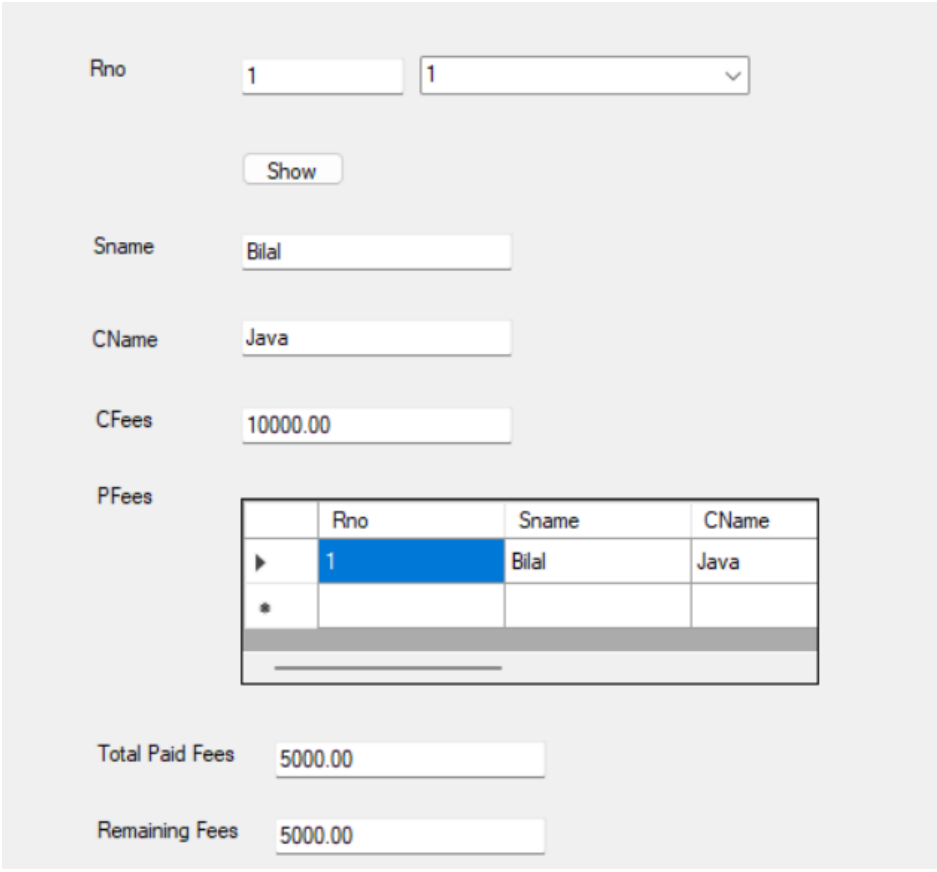
```
    MessageBox.Show("Please enter or select a valid Roll Number.");
    return;
}
try
{
    using (MySqlConnection conn = new MySqlConnection(connStr))
    {
        conn.Open();
        MySqlCommand cmd = new MySqlCommand("GetStudentFeeDetails", conn);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@input_rno", rno);

        MySqlDataAdapter da = new MySqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        da.Fill(dt);

        if (dt.Rows.Count > 0)
        {
            DataRow row = dt.Rows[0];

            textBox1.Text = row["Rno"].ToString();
            textBox2.Text = row["Sname"].ToString();
            textBox3.Text = row["CName"].ToString();
            textBox4.Text = row["CFees"].ToString();
            textBox5.Text = row["PFees"].ToString();
            textBox6.Text = row["RemainingFees"].ToString();

            dataGridView1.DataSource = dt;
        }
        else
        {
```

```
                MessageBox.Show("Record not found.");
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error fetching data: " + ex.Message);
    }
  }
 }
}
```

PROCEDURE :

```
DELIMITER //

CREATE PROCEDURE GetStudentFeeDetails(IN input_rno INT)
BEGIN
  SELECT
    Rno, Sname, CName, CFees, PFees,
    (CFees - PFees) AS RemainingFees
  FROM student_fees
  WHERE Rno = input_rno;
END //

DELIMITER ;
```

Output :



	Rno	Sname	CName	CFees	PFees
Edit Copy Delete	1	Bilal	Java	10000.00	5000.00
Edit Copy Delete	2	Maaj	Python	12000.00	4000.00
Edit Copy Delete	3	Mohseen	C#	7000.00	2000.00
Edit Copy Delete	4	Kaif	php	8000.00	3000.00