

Practical no. 1

1. **Design a window application to demonstrate basic and advanced controls. Create Registration form with following fields first name, middle name, surname, photo, dob, address, mobile no, username, password, gender, hobbies, city. Use appropriate controls to take inputs. And show the entered data on another form.**

**Design:**

**Code:**

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 Pract1\_RegistrationForm

{

publicpartialclassForm1 : Form

{

private String first = ""; publicForm1()

{

InitializeComponent();

} privatevoid Form1\_Load(object sender, EventArgs e)

{

}

privatevoid btnPhoto\_Click(object sender, EventArgs e)

{

OpenFileDialog opnfd = new OpenFileDialog(); opnfd.Filter = "Image Files (\*.jpg;\*.jpeg;.\*.gif;)|\*.jpg;\*.jpeg;.\*.gif"; if (opnfd.ShowDialog() == DialogResult.OK)

{

pictureBox1.Image = new Bitmap(opnfd.FileName);

}

}

privatevoid btnSubmit\_Click(object sender, EventArgs e)

{

string firstName = txtFirstName.Text; string middleName = txtMiddleName.Text; string lastName = txtLastName.Text;

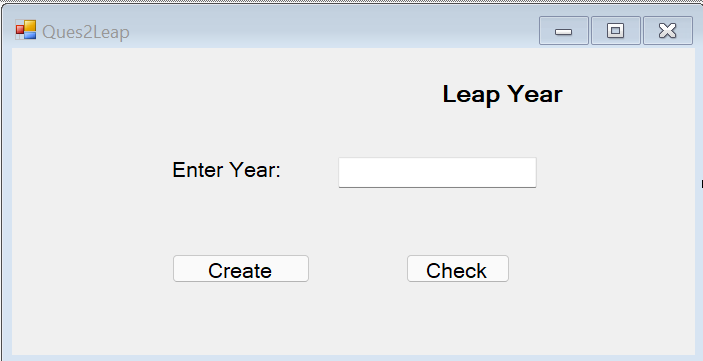
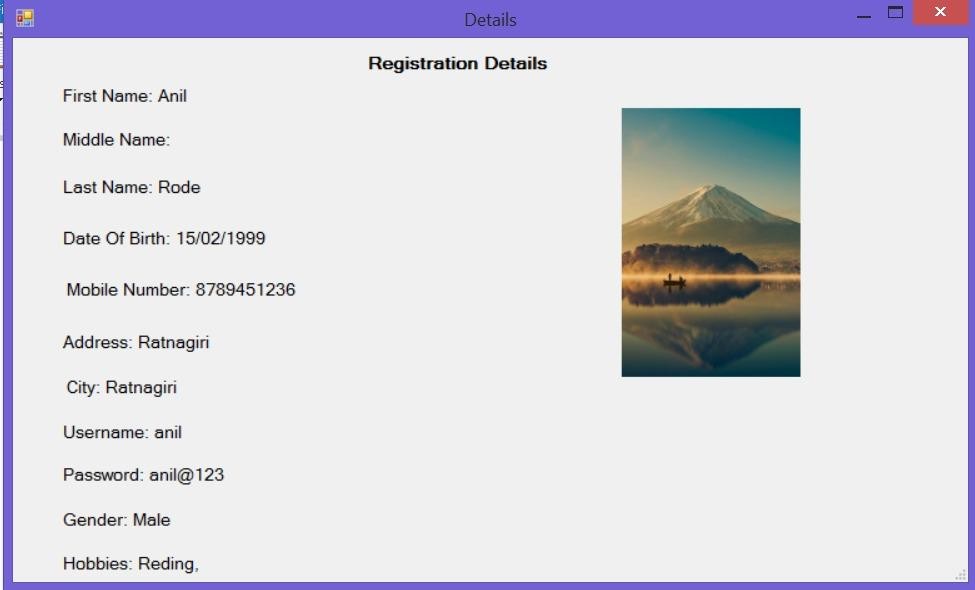
string dob = txtDOB.Text; string mobile=txtMobile.Text; string username=txtUsername.Text;

Details frm = new Details(pictureBox1.Image,firstName, middleName,lastName,dob,mobile,username); frm.Show();

}

} }

**Output:**



1. **Design a window application in c# using objects and classes to find whether an entered year is leap year or not.**

**Design:**

**Code:**

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 Practical1

{

publicpartialclassQues2Leap : Form

{

YearLeap leap =null;

publicQues2Leap()

{

InitializeComponent();

}

privatevoid btncreate\_Click(object sender, EventArgs e)

{

int year;

year = Convert.ToInt32(txtleap.Text); leap = new YearLeap(year);

MessageBox.Show("Object Created");

}

privatevoid btncheck\_Click(object sender, EventArgs e)

{

if (leap.checkLeapYear(year))

{

}

else

{

}

}

}

lblmsg.Text = year + " is leap year";

lblmsg.Text = year + " is not leap year";

publicclassYearLeap

{

int year;

YearLeap(int y)

{

Year=y;

}

public Boolean checkLeapYear(int y)

{

if ((y % 4 == 0 && y % 100 != 0) || (y % 400 == 0))

{

returntrue;

}

else

{

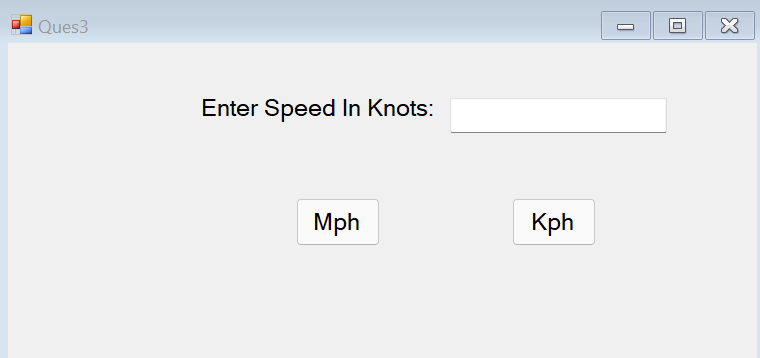
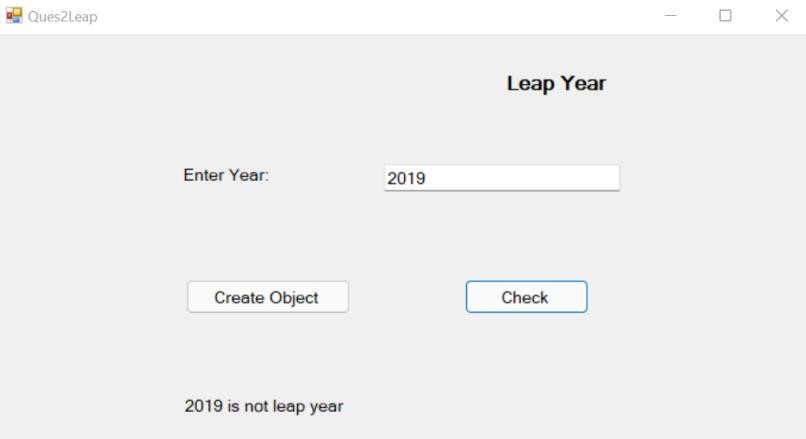
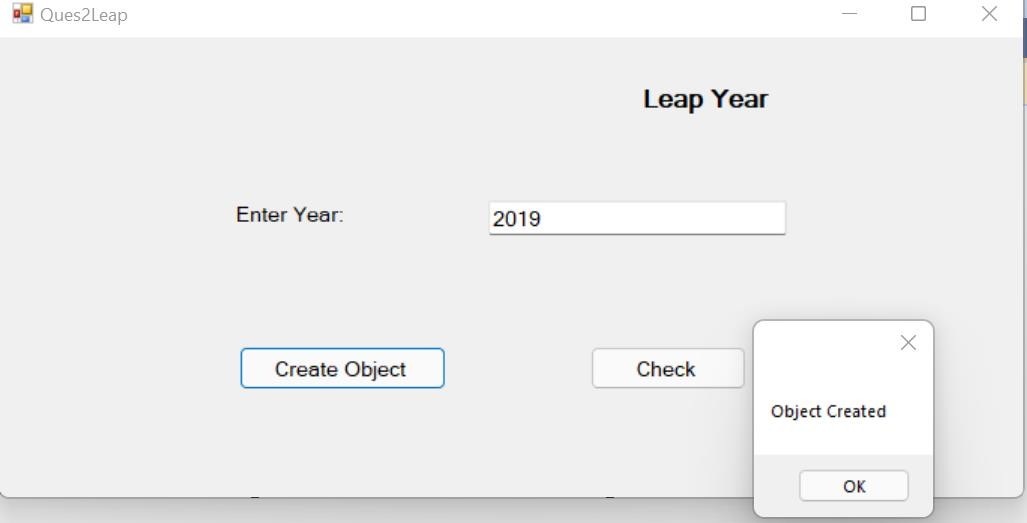
returnfalse;

}

}

}

}



**Output:**

1. **Design a Window application in c# using objects and classes for wind conversion from knots to mph, kph.**

**Design:**

**Code:**

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 Practical1

{

publicpartialclassQues3 : Form

{

int knots = Convert.ToInt32(txtknots.Text); SpeedConvert c = new SpeedConvert(knots);

publicQues3()

{

InitializeComponent();

}

privatevoid Ques3\_Load(object sender, EventArgs e)

{

}

privatevoid btnmph\_Click(object sender, EventArgs e)

{

lblmsg.Text ="Speed In Miles Per Hour: "+Convert.ToString(c.convertMph());

}

privatevoid btnkph\_Click(object sender, EventArgs e)

{

lblmsg.Text = "Speed In Kilometer Per Hour: " + Convert.ToString(c.convertKph());

}

}

publicclassSpeedConvert

{

int knots;

SpeedConvert(int kn)

{

Knots=kn;

}

publicdouble convertMph()

{

return knots \* 1.1508;

}

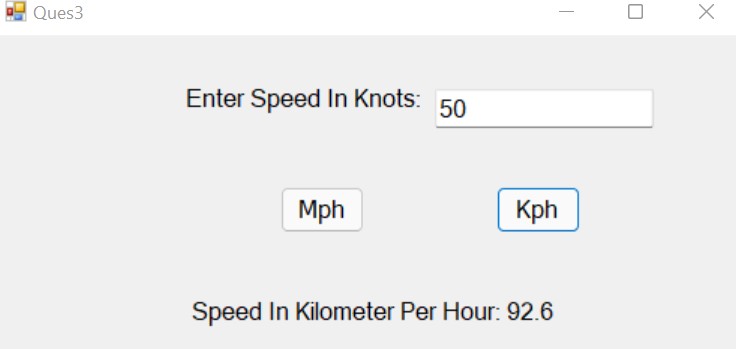
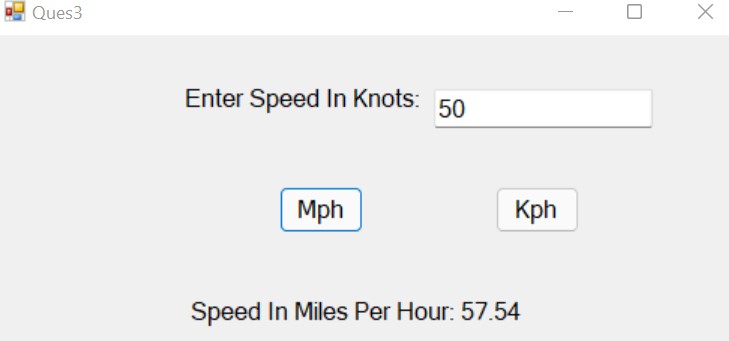
publicdouble convertKph()

{

return knots \* 1.852;

}

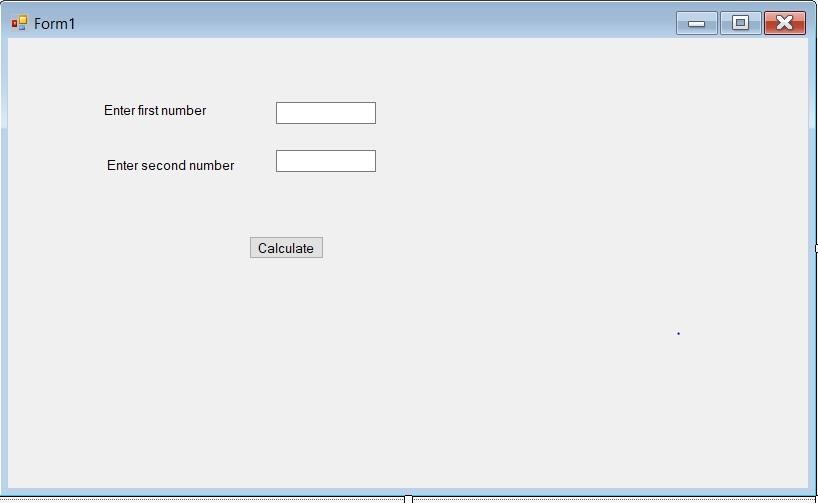
}



}

1. **Design a Window application to demonstrate multiple inheritance.**

**Design:**



**Code:**

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 Pract1\_multipleInheritence

{ interfaceaddition

{

int add(int a, int b);

}

interfacesubstraction

{

int sub(int a, int b);

}

publicpartialclassForm1 : Form

{

publicForm1()

{

InitializeComponent();

}

privatevoid button1\_Click(object sender, EventArgs e)

{ try

{

int num1 = Convert.ToInt32(txt1.Text); int num2 = Convert.ToInt32(txt2.Text);

Calculation obj = new Calculation(); int rs1=obj.add(num1, num2); int rs2=obj.sub(num1, num2);

l1.Text ="Addition of two numbers: " +rs1.ToString(); l2.Text = "Substraction of two numbers: "+rs2.ToString();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

classCalculation : addition, substraction

{

publicint add(int a, int b)

{

return a + b;

}

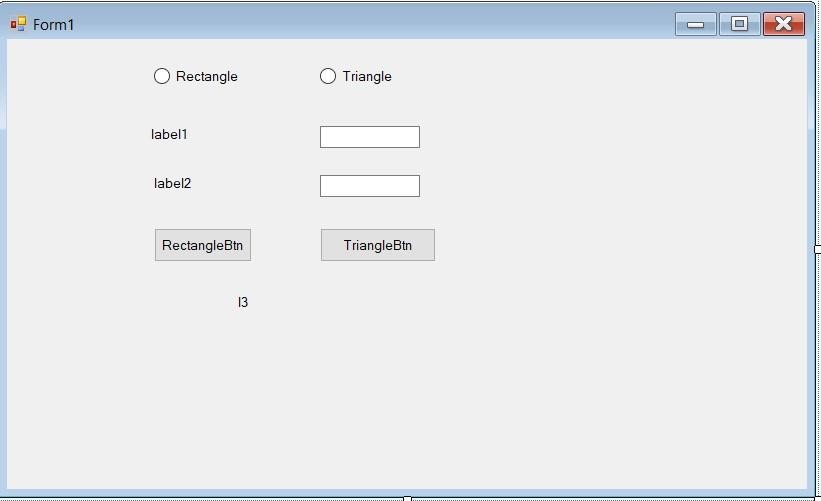
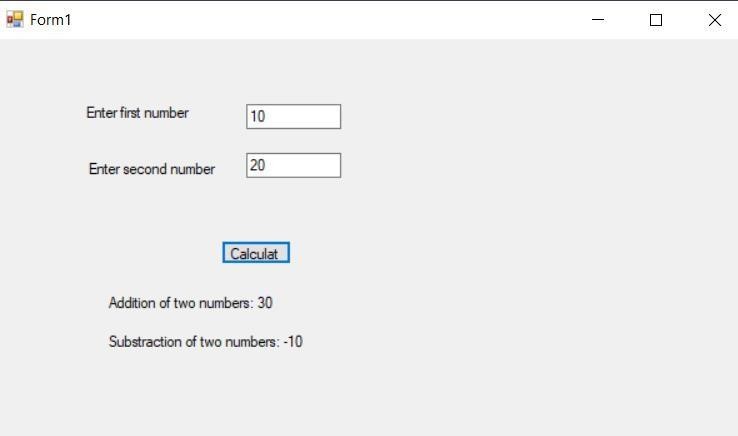
publicint sub(int a, int b)

{

return a - b;

}

} }



**Output:**

1. **Design a Window application to demonstrate abstract class. Design:**

**Code:**

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 pract1\_abstractClas

{

publicpartialclassForm1 : Form

{

RectangleArea obj1 = null; TriangleArea obj2 = null; publicForm1()

{

InitializeComponent();

}

privatevoid Form1\_Load(object sender, EventArgs e)

{

l1.Text = ""; l2.Text = ""; l3.Text = "";

RectangleBtn.Enabled = false; TriangleBtn.Enabled = false;

}

privatevoid rectangle\_CheckedChanged(object sender, EventArgs e)

{

l1.Text = "Enter Lenght"; l2.Text = "Enter Breadth";

RectangleBtn.Enabled = true; TriangleBtn.Enabled = false;

}

privatevoid triangle\_CheckedChanged(object sender, EventArgs e)

{

l1.Text = "Enter Base"; l2.Text = "Enter Height"; TriangleBtn.Enabled = true;

RectangleBtn.Enabled = false;

}

privatevoid RectangleBtn\_Click(object sender, EventArgs e)

{

double val1 = Convert.ToDouble(txt1.Text); double val2 = Convert.ToDouble(txt2.Text);

obj1 = new RectangleArea();

l3.Text="Area of Rectangle: "+Convert.ToString(obj1.Area(val1, val2));

}

privatevoid TriangleBtn\_Click(object sender, EventArgs e)

{

double val1 = Convert.ToDouble(txt1.Text); double val2 = Convert.ToDouble(txt2.Text);

obj2 = new TriangleArea();

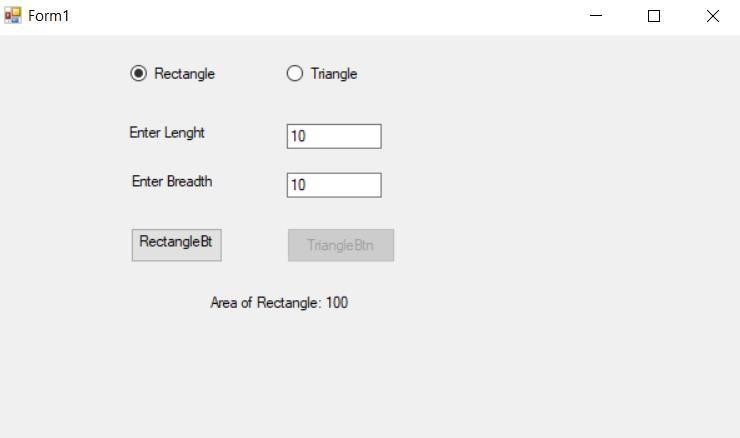
l3.Text = "Area of Triangle: " + Convert.ToString(obj2.Area(val1, val2)); } } abstractclassShape

{

publicabstractdouble Area(double val1, double val2);

}

classRectangleArea : Shape



{

publicoverridedouble Area(double val1, double val2) { return val1 \* val2;

}

}

classTriangleArea : Shape

{

publicoverridedouble Area(double val1, double val2)

{

return (0.5 \* val1 \* val2);

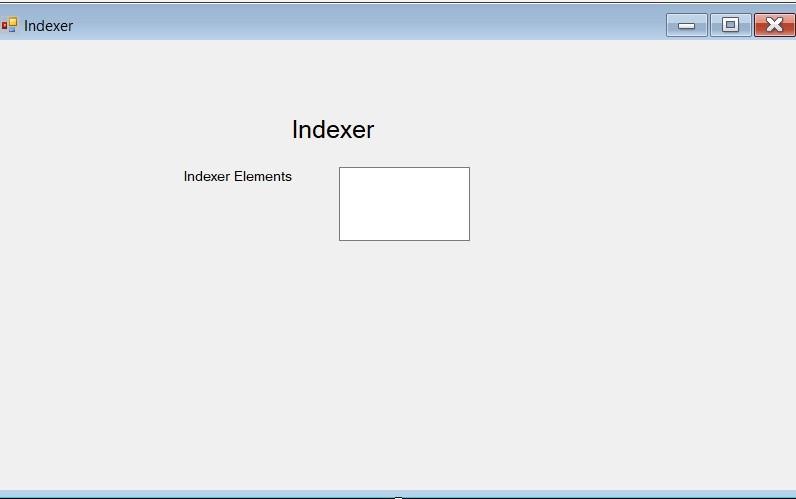
}

} }

**Output:**

1. **Design a Window application to demonstrate Indexer.**

**Design:**



**Code:**

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 Practical\_1.\_5

{

publicpartialclassForm1 : Form

{

publicForm1()

{

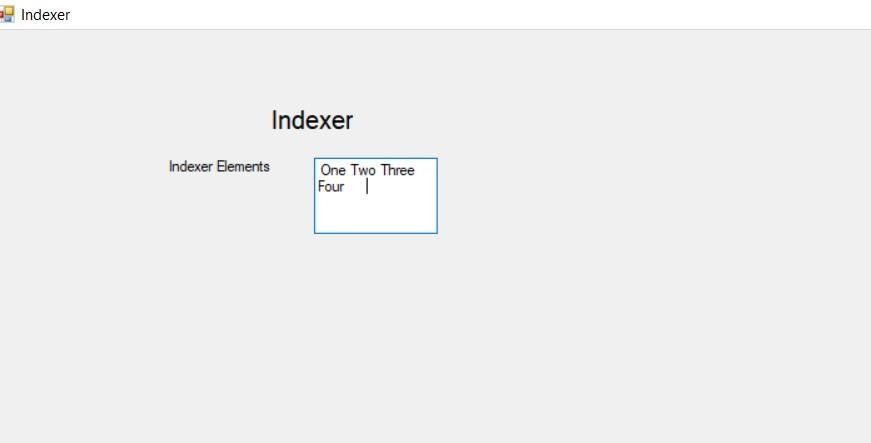
InitializeComponent();

}

privatevoid Form1\_Load(object sender, EventArgs e)

{

StringDataStore strStore = new StringDataStore(); strStore[0] = "One"; strStore[1] = "Two"; strStore[2] = "Three"; strStore[3] = "Four"; for (int i = 0; i < 10; i++)



txtElement.Text += " " + strStore[i];

} }

classStringDataStore

{

privatestring[] strArr = newstring[10]; publicstringthis[int index]

{ get

{

if (index < 0 || index >= strArr.Length)

thrownew IndexOutOfRangeException("Index out of range"); return strArr[index];

} set

{

if (index < 0 || index >= strArr.Length)

thrownew IndexOutOfRangeException("Index out of range"); strArr[index] = value;

}

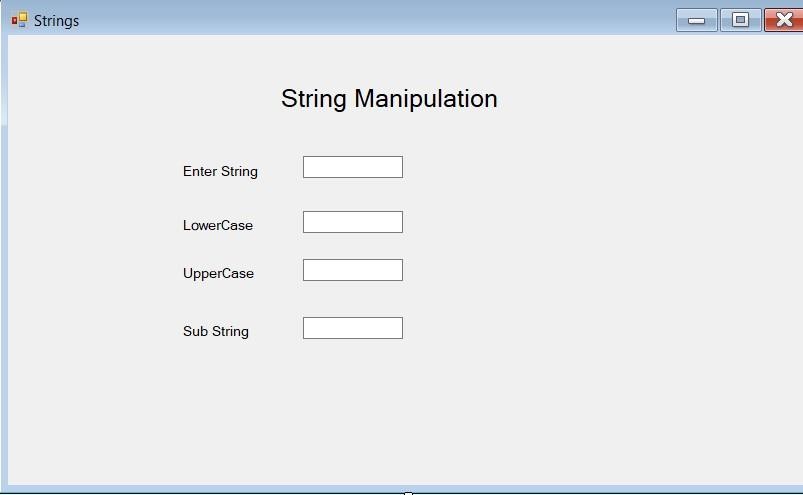
}

} }

**Output:**

1. **Design a Window application to demonstrate string manipulations.**

**Design:**



**Code:**

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 Practical\_1.\_6

{ publicpartialclassForm1 : Form

{

publicForm1()

{

InitializeComponent();

} privatevoid Form1\_Load(object sender, EventArgs e)

{

}

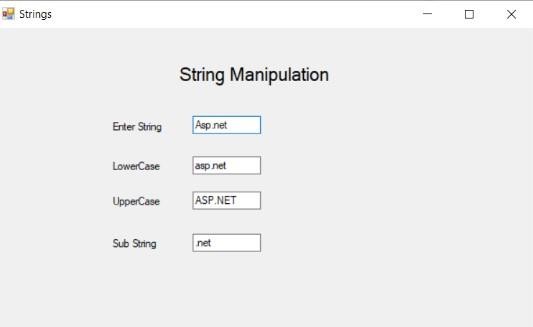
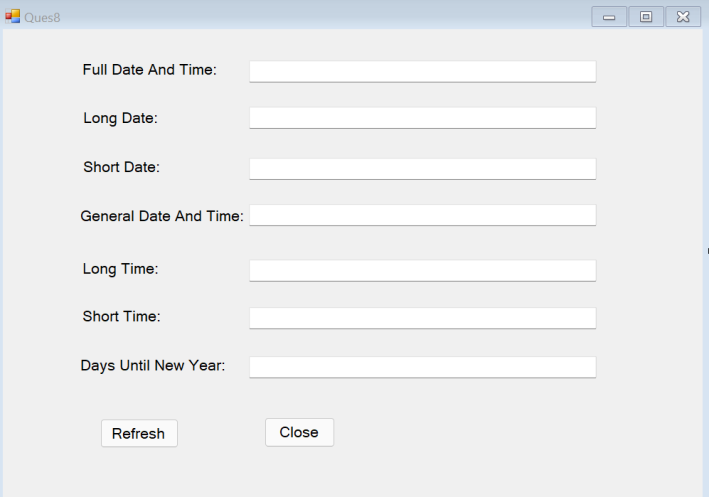
privatevoid textBox1\_TextChanged(object sender, EventArgs e)

{

lbllower.Text = txtString.Text.ToLower(); lblupper.Text = txtString.Text.ToUpper(); if (txtString.Text.Length >= 3)

lblsubstring.Text = txtString.Text.Substring(3);

}



} }

**Output:**

1. **Design a window application to show following output. Design:**

**Code:**

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 Practical1

{

publicpartialclassQues8 : Form

{

DateTime d = new DateTime();

publicQues8()

{

InitializeComponent();

}

privatevoid btnrefresh\_Click(object sender, EventArgs e)

{

d = DateTime.Now;

txtfullname.Text = d.ToString("dddd, MMMM dd yyyy HH:mm:ss tt"); txtlongdate.Text = d.ToLongDateString();

txtshortdate.Text = d.ToString("dd-MMM-yy"); txtgeneral.Text = d.ToString(" dd-MMM-yy HH:mm tt"); txtlongtime.Text = d.ToLongTimeString(); txtshorttime.Text = d.ToShortTimeString();

int daysInYear = DateTime.IsLeapYear(d.Year) ? 366 : 365; int daysLeftInYear = daysInYear - d.DayOfYear;

txtdays.Text = daysLeftInYear.ToString();

txtfullname.Enabled = false; txtlongdate.Enabled = false; txtshortdate.Enabled = false; txtgeneral.Enabled = false; txtlongtime.Enabled = false; txtshorttime.Enabled = false; txtdays.Enabled = false;

}

privatevoid btnclose\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

**Output:**

