|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

|  |
| --- |
| Your name: Preet Manandhar  4/18/2023  Version: 8 |

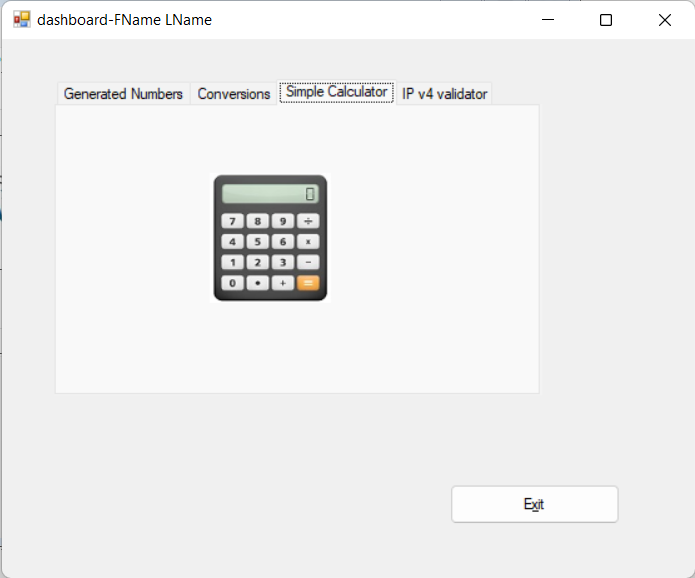
1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language: C#
3. Tools (IDE): Visual Studio 2022

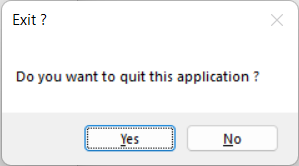
Multiform (.Net Framework) project, covers 6 applications

1. **Present the print screens of yours forms, and have a detailed description of the functionalities (step by step).**

Graphical user interface, application, Word

Description automatically generatedGraphical user interface, application

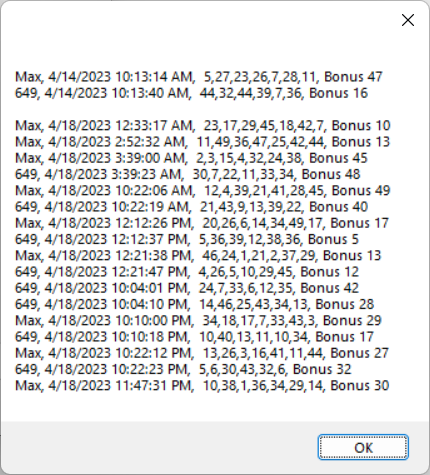
Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

This is my main dashboard form. It has group boxes and 6 forms inside. It displays message of confirmation when we press button exit.

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Form Max

This form generates 8 random unique numbers out of 1 to 50 numbers. This form then writes and read text file in LottoNumbers.txt. The last number is bonus number.

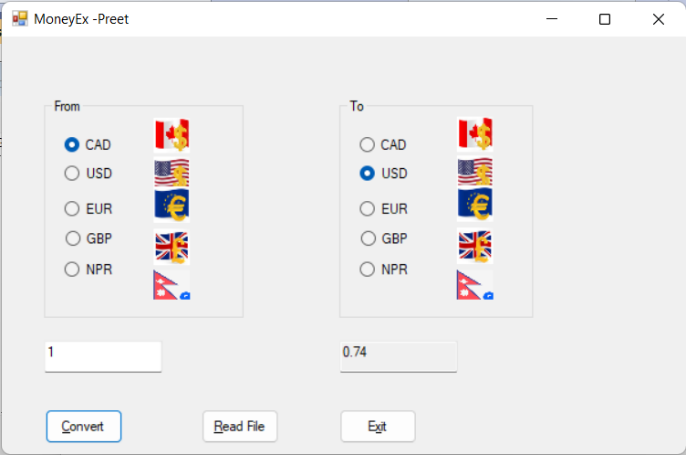
Graphical user interface, application

Description automatically generatedText

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Form lotto 649

This form has button to generate number(which also write text file name LottoNumber.txt),botton to read file and exit botton that ask permission before closing of the form.Text, chat or text message

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Form Money exchange

This form perform function to calculate currency. It did conversions from one country currency to another. In this form, I convert each country currency into another with calculations using checked radiobuttons. The botton convert has all the calculations of checking radio buttons from and to and then I stored currencies display in string. Then it writes in the file named MoneyExchange.txt.

The botton read file reads the text file and displays information with today date

The botton exit closes the form with this.Close asking permission from users before closing it.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

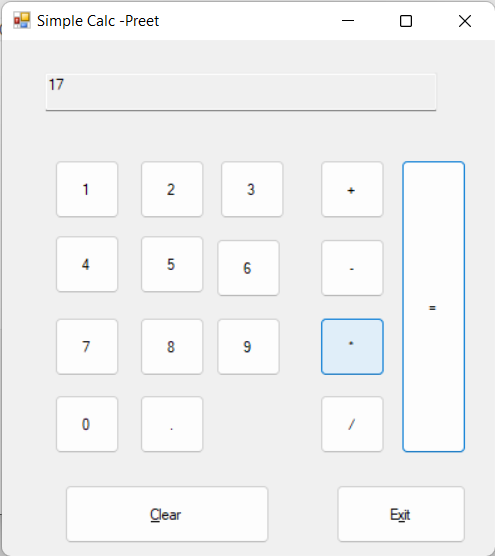
Description automatically generated

Form Temperature conversion

This form has 2 radio buttons to convert temperatures from Celsius to Fahrenheit and vice versa.

Among 2 text boxes, one is read only that displays result. I use radio click check method to change label of radio buttons. The form has message box that displays message in different font colors .3 buttons are used which are to convert(write text as well), read file TempConversions.txt and exit botton.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Form Calculator

I have created class Class to perform basic operations with private and public field, constructir, property and methods. Values are stored in operands and op is operator. when the equal button is pressed, the result of the operation will be displayed. Clear button will clear all the fields’ members of the Calculator class object, and the text box

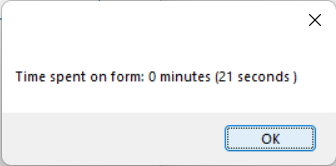
Graphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

Form IP4 Validator

This form lets the user perform operations i.e validation of ip, reset and exit form using appropriate String and DateTime object methods (ToLongDateString(), Trim(),Split()).

I have used regex to have Ip of 4 bytes and between 0 to 255.

1. **Present the code of your application (forms).**

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 frmDashboard

{

public partial class Dashboard : Form

{

public Dashboard()

{

InitializeComponent();

}

private void Dashboard\_Load(object sender, EventArgs e)

{

startTime = DateTime.Now;

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

frmMax obj = new frmMax();

//obj.Show();

obj.ShowDialog();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

Lotto649 obj = new Lotto649();

obj.Show();

}

private DateTime startTime;

private DateTime endTime;

private void button7\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit this application ?", "Exit ?", MessageBoxButtons.YesNo).ToString() == "Yes")

{

// Set the end time to the current time

endTime = DateTime.Now;

// Calculate the time difference in seconds and minutes

TimeSpan timeDiff = endTime - startTime;

int seconds = (int)timeDiff.TotalSeconds;

int minutes = (int)timeDiff.TotalMinutes;

// Display the time on the form using a MessageBox

MessageBox.Show($"Time spent on form: {minutes} minutes ({seconds} seconds )");

Application.Exit();

}

Application.Exit();

}

//temperature

private void pictureBox6\_Click(object sender, EventArgs e)

{

Ip4Validator obj = new Ip4Validator();

obj.Show();

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

Calculator obj = new Calculator();

obj.Show();

}

private void pictureBox4\_Click\_1(object sender, EventArgs e)

{

MoneyExchange obj = new MoneyExchange();

obj.Show();

}

//temperature

private void pictureBox3\_Click\_1(object sender, EventArgs e)

{

TemperatureConversion obj = new TemperatureConversion();

obj.Show();

}

}

}

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 frmDashboard

{

public partial class Dashboard : Form

{

public Dashboard()

{

InitializeComponent();

}

private void Dashboard\_Load(object sender, EventArgs e)

{

startTime = DateTime.Now;

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

frmMax obj = new frmMax();

//obj.Show();

obj.ShowDialog();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

Lotto649 obj = new Lotto649();

obj.Show();

}

private DateTime startTime;

private DateTime endTime;

private void button7\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit this application ?", "Exit ?", MessageBoxButtons.YesNo).ToString() == "Yes")

{

// Set the end time to the current time

endTime = DateTime.Now;

// Calculate the time difference in seconds and minutes

TimeSpan timeDiff = endTime - startTime;

int seconds = (int)timeDiff.TotalSeconds;

int minutes = (int)timeDiff.TotalMinutes;

// Display the time on the form using a MessageBox

MessageBox.Show($"Time spent on form: {minutes} minutes ({seconds} seconds )");

Application.Exit();

}

Application.Exit();

}

//temperature

private void pictureBox6\_Click(object sender, EventArgs e)

{

Ip4Validator obj = new Ip4Validator();

obj.Show();

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

Calculator obj = new Calculator();

obj.Show();

}

private void pictureBox4\_Click\_1(object sender, EventArgs e)

{

MoneyExchange obj = new MoneyExchange();

obj.Show();

}

//temperature

private void pictureBox3\_Click\_1(object sender, EventArgs e)

{

TemperatureConversion obj = new TemperatureConversion();

obj.Show();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

/\*name : preet manandhar

this form Lotto649 generates 7 random numbers between 1 to 49 and then write and read in text file

botton 1 is for generating and writing text file

botton 2 is for readinf text file

\*/

namespace frmDashboard

{

public partial class Lotto649 : Form

{

public Lotto649()

{

InitializeComponent();

}

string path = @".\LottoNbrs.txt";

private void button1\_Click(object sender, EventArgs e)

{

int lastNumber = 0;

string numbers = "";

textBox1.Text = "";

string currentDate = DateTime.Now.ToString();

Random random = new Random();

//loop to create random numbers

for (int i = 0; i < 7; i++)

{

int randomNumber = random.Next(1, 49);

textBox1.Text += randomNumber + "\t";

if (i < 6) numbers += randomNumber.ToString() + ",";

else lastNumber = randomNumber;

}

//for writing

FileStream fs = null;

try

{

fs = new FileStream(path, FileMode.Append, FileAccess.Write);

// create the output stream for a text file that exists

StreamWriter textOut = new StreamWriter(fs);

// write the fields into text file

textOut.WriteLine($"649, {currentDate}, {numbers} Bonus {lastNumber}");

// close the output stream for the text file

textOut.Close();

}

catch (IOException ex)

{ MessageBox.Show(ex.Message, "IOException"); }

finally {

if (fs != null) fs.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

FileStream fs2 = null;

fs2 = new FileStream(path, FileMode.OpenOrCreate,

FileAccess.Read);

// create the object for the input stream for a text file

StreamReader textIn = new StreamReader(fs2);

string textToPrint = "";

while (textIn.Peek() != -1) {

textToPrint += textIn.ReadLine()+"\n";

// read the data from the file and store it in the list

}

MessageBox.Show(textToPrint);

// close the input stream for the text file

textIn.Close();

}

private void button3\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit this application?", "Exit?", MessageBoxButtons.YesNo).ToString() == "Yes") ;

{

this.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Security.Policy;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

/\*name= Preet Manandhar

this form is for money exchange . it has radio buttons for choices and is used to convert currency from one to another.

textbox on left is to enter the currency and text box on right is readonly to get desired currency

this form also has read file .i used switch cases for radio button options

regular expression is used for currency

while closing form it displays time of duration that this form was loaded for.\*/

namespace frmDashboard

{

public partial class MoneyExchange : Form

{

private string from;

private string to;

private double money;

public string From

{

get { return from; }

set { from = value; }

}

public string To

{

get { return to; }

set { to = value; }

}

public MoneyExchange()

{

InitializeComponent();

}

public MoneyExchange(double money)

{

this.money = money;

}

public double MoneyGetSet

{

get { return money; }

set { money = value; }

}

public double CADToCAD(double money)

{

return money;

}

public double CADToUSD(double money)

{

return money \* 0.74;

}

public double CADToEUR(double money)

{

return money \* 0.68;

}

public double CADToGBP(double money)

{

return money \* 0.60;

}

public double CADToNPR(double money)

{

return money \* 98.96;

}

public double USDToCAD(double money)

{

return money \* 1.35;

}

public double USDToUSD(double money)

{

return money;

}

public double USDToEUR(double money)

{

return money \* 0.92;

}

public double USDToGBP(double money)

{

return money \* 0.80;

}

public double USDToNPR(double money)

{

return money \* 100;

}

public double EURToCAD(double money)

{

return money \* 1.47;

}

public double EURToUSD(double money)

{

return money \* 1.09;

}

public double EURToEUR(double money)

{

return money;

}

public double EURToGBP(double money)

{

return money \* 0.88;

}

public double EURToNPR(double money)

{

return money \* 144.10;

}

public double GBPToCAD(double money)

{

return money \* 1.68;

}

public double GBPToUSD(double money)

{

return money \* 1.24;

}

public double GBPToEUR(double money)

{

return money \* 1.14;

}

public double GBPToGBP(double money)

{

return money;

}

public double GBPToNPR(double money)

{

return money \* 163.17;

}

public double NPRToCAD(double money)

{

return money \* 0.010;

}

public double NPRToUSD(double money)

{

return money \* 0.0076;

}

public double NPRToEUR(double money)

{

return money \* 0.0069;

}

public double NPRToGBP(double money)

{

return money \* 0.0061;

}

public double NPRToNPR(double money)

{

return money;

}

string path = @".\MoneyExchange.txt";

private void button1\_Click(object sender, EventArgs e)

{

double input;

try

{

input = Convert.ToDouble(textBox1.Text);

}

catch (FormatException)

{

MessageBox.Show("Please enter a valid number.");

return;

}

double fromVal, toVal;

if (cadfrom.Checked == true && cadto.Checked == true)

{

//fromVal = input;

toVal = CADToCAD(input);

textBox2.Text = Convert.ToString(toVal);

from = "CAD";

to = "CAD";

//mx.Write(fromVal, toVal, from, to);

}

else if (cadfrom.Checked == true && usdto.Checked == true)

{

fromVal = input;

toVal = CADToUSD(input);

textBox2.Text = Convert.ToString(toVal);

from = "CAD";

to = "USD";

// mx.Write(fromVal, toVal, from, to);

}

else if (cadfrom.Checked == true && eurto.Checked == true)

{

fromVal = input;

toVal = CADToEUR(input);

textBox2.Text = Convert.ToString(toVal);

from = "CAD";

to = "EUR";

//mx.Write(fromVal, toVal, from, to);

}

else if (cadfrom.Checked == true && gbpto.Checked == true)

{

fromVal = input;

toVal = CADToGBP(input);

textBox2.Text = Convert.ToString(toVal);

from = "CAD";

to = "GBP";

//mx.Write(fromVal, toVal, from, to);

}

else if (cadfrom.Checked == true && nprto.Checked == true)

{

fromVal = input;

toVal = CADToNPR(input);

textBox2.Text = Convert.ToString(toVal);

from = "CAD";

to = "NPR";

//mx.Write(fromVal, toVal, from, to);

}

else if (usdfrom.Checked == true && cadto.Checked == true)

{

fromVal = input;

toVal = USDToCAD(input);

textBox2.Text = Convert.ToString(toVal);

from = "USD";

to = "CAD";

//mx.Write(fromVal, toVal, from, to);

}

else if (usdfrom.Checked == true && usdto.Checked == true)

{

fromVal = input;

toVal = USDToUSD(input);

textBox2.Text = Convert.ToString(toVal);

from = "USD";

to = "USD";

//mx.Write(fromVal, toVal, from, to);

}

else if (usdfrom.Checked == true && eurto.Checked == true)

{

fromVal = input;

toVal = USDToEUR(input);

textBox2.Text = Convert.ToString(toVal);

from = "USD";

to = "EUR";

//mx.Write(fromVal, toVal, from, to);

}

else if (usdfrom.Checked == true && gbpto.Checked == true)

{

fromVal = input;

toVal = USDToGBP(input);

textBox2.Text = Convert.ToString(toVal);

from = "USD";

to = "GBP";

// mx.Write(fromVal, toVal, from, to);

}

else if (usdfrom.Checked == true && nprto.Checked == true)

{

fromVal = input;

toVal = USDToNPR(input);

textBox2.Text = Convert.ToString(toVal);

from = "USD";

to = "NPR";

//mx.Write(fromVal, toVal, from, to);

}

else if (eurfrom.Checked == true && cadto.Checked == true)

{

fromVal = input;

toVal = EURToCAD(input);

textBox2.Text = Convert.ToString(toVal);

from = "EUR";

to = "CAD";

// mx.Write(fromVal, toVal, from, to);

}

else if (eurfrom.Checked == true && usdto.Checked == true)

{

fromVal = input;

toVal = EURToUSD(input);

textBox2.Text = Convert.ToString(toVal);

from = "EUR";

to = "USD";

//mx.Write(fromVal, toVal, from, to);

}

else if (eurfrom.Checked == true && eurto.Checked == true)

{

fromVal = input;

toVal = EURToEUR(input);

textBox2.Text = Convert.ToString(toVal);

from = "EUR";

to = "EUR";

//mx.Write(fromVal, toVal, from, to);

}

else if (eurfrom.Checked == true && gbpto.Checked == true)

{

fromVal = input;

toVal = EURToGBP(input);

textBox2.Text = Convert.ToString(toVal);

from = "EUR";

to = "GBP";

//mx.Write(fromVal, toVal, from, to);

}

else if (eurfrom.Checked == true && nprto.Checked == true)

{

fromVal = input;

toVal = EURToNPR(input);

textBox2.Text = Convert.ToString(toVal);

from = "EUR";

to = "IND";

//mx.Write(fromVal, toVal, from, to);

}

else if (gbpfrom.Checked == true && cadto.Checked == true)

{

fromVal = input;

toVal = GBPToCAD(input);

textBox2.Text = Convert.ToString(toVal);

from = "GBP";

to = "CAD";

// mx.Write(fromVal, toVal, from, to);

}

else if (gbpfrom.Checked == true && usdto.Checked == true)

{

fromVal = input;

toVal = GBPToUSD(input);

textBox2.Text = Convert.ToString(toVal);

from = "GBP";

to = "USD";

// mx.Write(fromVal, toVal, from, to);

}

else if (gbpfrom.Checked == true && eurto.Checked == true)

{

fromVal = input;

toVal = GBPToEUR(input);

textBox2.Text = Convert.ToString(toVal);

from = "GBP";

to = "EUR";

// mx.Write(fromVal, toVal, from, to);

}

else if (gbpfrom.Checked == true && gbpto.Checked == true)

{

fromVal = input;

toVal = GBPToGBP(input);

textBox2.Text = Convert.ToString(toVal);

from = "GBP";

to = "GBP";

// mx.Write(fromVal, toVal, from, to);

}

else if (gbpfrom.Checked == true && nprto.Checked == true)

{

fromVal = input;

toVal = GBPToNPR(input);

textBox2.Text = Convert.ToString(toVal);

from = "GBP";

to = "NPR";

// mx.Write(fromVal, toVal, from, to);

}

else if (nprfrom.Checked == true && cadto.Checked == true)

{

fromVal = input;

toVal = NPRToCAD(input);

textBox2.Text = Convert.ToString(toVal);

from = "NPR";

to = "CAD";

// mx.Write(fromVal, toVal, from, to);

}

else if (nprfrom.Checked == true && usdto.Checked == true)

{

fromVal = input;

toVal = NPRToUSD(input);

textBox2.Text = Convert.ToString(toVal);

from = "NPR";

to = "USD";

//mx.Write(fromVal, toVal, from, to);

}

else if (nprfrom.Checked == true && eurto.Checked == true)

{

fromVal = input;

toVal = NPRToCAD(input);

textBox2.Text = Convert.ToString(toVal);

from = "NPR";

to = "EUR";

// mx.Write(fromVal, toVal, from, to);

}

else if (nprfrom.Checked == true && gbpto.Checked == true)

{

fromVal = input;

toVal = NPRToGBP(input);

textBox2.Text = Convert.ToString(toVal);

from = "NPR";

to = "GBP";

//mx.Write(fromVal, toVal, from, to);

}

else if (nprfrom.Checked == true && nprto.Checked == true)

{

fromVal = input;

toVal = NPRToNPR(input);

textBox2.Text = Convert.ToString(toVal);

from = "NPR";

to = "NPR";

}

FileStream fs1 = null;

try

{

fs1 = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs1);

//textOut.Write(textBox1.Text + " = " + textBox2.Text + ", " + DateTime.Now );

// textOut.Write($"{textBox1.Text} , {from} ," = " + textBox2.Text + ", " + DateTime.Now);

//textOut.WriteLine(textBox1.Text + from + " = " + textBox2.Text + to + ", " + DateTime.Now);

textOut.WriteLine(textBox1.Text + " " + From + " = " + textBox2.Text + " " + To + ", " + DateTime.Now);

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

{ MessageBox.Show(ex.Message, "IOException"); }

finally { if (fs1 != null) fs1.Close(); }

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

using (StreamReader f2 = new StreamReader(@".\MoneyExchange.txt"))

{

string contents = f2.ReadToEnd();

if (contents != null && contents.Trim() != "")

{

MessageBox.Show(contents, "Money Exchange - Preet", MessageBoxButtons.OK);

}

else

{

MessageBox.Show("The Textfile is empty.");

}

}

}

catch (Exception ex)

{

MessageBox.Show("Error reading file: " + ex.Message);

}

textBox1.Clear();

textBox2.Clear();

}

private DateTime startTime;

private DateTime endTime;

private void button3\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit this application IP Validator?", "Exit ?", MessageBoxButtons.YesNo).ToString() == "Yes")

{

// Set the end time to the current time

endTime = DateTime.Now;

// Calculate the time difference in seconds and minutes

TimeSpan timeDiff = endTime - startTime;

int seconds = (int)timeDiff.TotalSeconds;

int minutes = (int)timeDiff.TotalMinutes;

// Display the time on the form using a MessageBox

MessageBox.Show($"Time spent on form: {minutes} minutes ({seconds} seconds )");

this.Close();

}

}

private void MoneyExchange\_Load(object sender, EventArgs e)

{

startTime = DateTime.Now;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

/\*name= Preet Manandhar

this form is about temperature conversion form celsius to farenheit and farenheit to celsius

i have used 3 bottons that are to convert read and right respectively

right side text box is read inly and i have used colors to display message \*/

namespace frmDashboard

{

public partial class TemperatureConversion : Form

{

public TemperatureConversion()

{

InitializeComponent();

}

// string path = @"\.TempConversions.txt"; //path of text file

private void button2\_Click(object sender, EventArgs e)

{

try

{

using (StreamReader f2 = new StreamReader(@".\TempConversions.txt"))

{

string contents = f2.ReadToEnd();

if (contents != null && contents.Trim() != "")

{

MessageBox.Show(contents, "Temperature Conversion - Preet", MessageBoxButtons.OK);

}

else

{

MessageBox.Show("The Textfile is empty.");

}

}

}

catch (Exception ex)

{

MessageBox.Show("Error reading file: " + ex.Message);

}

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

}

private DateTime startTime;

private DateTime endTime;

private void button3\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit this application IP Validator?", "Exit ?", MessageBoxButtons.YesNo).ToString() == "Yes")

{

// Set the end time to the current time

endTime = DateTime.Now;

// Calculate the time difference in seconds and minutes

TimeSpan timeDiff = endTime - startTime;

int seconds = (int)timeDiff.TotalSeconds;

int minutes = (int)timeDiff.TotalMinutes;

// Display the time on the form using a MessageBox

MessageBox.Show($"Time spent on form: {minutes} minutes ({seconds} seconds )");

this.Close();

}

this.Close();

}

string path = @".\TempConversions.txt";

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

{

try

{

decimal TempC;

decimal TempF;

Regex codeRegex = new Regex(@"^-?\d\*\.?\d\*$");

string temp = textBox1.Text.Trim();

if (!codeRegex.IsMatch(temp))

{

MessageBox.Show("Please enter a decimal number.");

textBox1.Clear();

textBox2.Clear();

}

else if (codeRegex.IsMatch(temp))

{

decimal input = decimal.Parse(textBox1.Text.Trim());

if (radioButton1.Checked) // celsius to farenheit

{

TempC = input;

TempF = (TempC \* 9 / 5) + 32;

TempF = Decimal.Round(TempF, 1);

textBox2.Text = TempF.ToString();

switch (textBox2.Text)

{

case "-40":

textBox2.ForeColor = Color.Purple;

textBox3.ForeColor = Color.Purple;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Extremely Cold Day\r\n (and the same number!)";

break;

case "0":

textBox2.ForeColor = Color.Blue;

textBox3.ForeColor = Color.Blue;

textBox3.Font = new Font(textBox3.Font, FontStyle.Regular);

textBox3.Text = "Very Cold Day";

break;

case "32":

textBox2.ForeColor = Color.Blue;

textBox3.ForeColor = Color.Blue;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Freezing point of water";

break;

case "50":

textBox2.ForeColor = Color.LightBlue;

textBox3.ForeColor = Color.LightBlue;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Cool Day";

break;

case "69.8":

textBox2.ForeColor = Color.Green;

textBox3.ForeColor = Color.Green;

textBox3.Font = new Font(textBox3.Font, FontStyle.Regular);

textBox3.Text = "Room temperature";

break;

case "70":

textBox2.ForeColor = Color.Green;

textBox3.ForeColor = Color.Green;

textBox3.Font = new Font(textBox3.Font, FontStyle.Regular);

textBox3.Text = "Room temperature";

break;

case "86":

textBox2.ForeColor = Color.Orange;

textBox3.ForeColor = Color.Orange;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Beach weather";

break;

case "98.6":

textBox2.ForeColor = Color.Orange;

textBox3.ForeColor = Color.Orange;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = " Body temperature";

break;

case "104":

textBox2.ForeColor = Color.Red;

textBox3.ForeColor = Color.Red;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Hot Bath";

break;

case "212":

textBox2.ForeColor = Color.DarkRed;

textBox3.ForeColor = Color.DarkRed;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Water Boils";

break;

default:

textBox3.Clear();

break;

}

FileStream fs1 = null;

try

{

fs1 = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs1);

textOut.Write(textBox1.Text.Trim() + "°C = ");

textOut.Write(textBox2.Text.Trim() + "°F, ");

textOut.Write(DateTime.Now.ToString("yyyy/MM/dd h:mm:ss tt "));

textOut.WriteLine(textBox3.Text.Trim() + " ");

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

{ MessageBox.Show(ex.Message, "IOException"); }

finally { if (fs1 != null) fs1.Close(); }

}

else if (radioButton2.Checked) // farenheit to celcius

{

TempF = decimal.Parse(textBox1.Text);

TempC = (TempF - 32) \* 5 / 9;

TempC = decimal.Round(TempC);

textBox2.Text = TempC.ToString();

switch (textBox2.Text)

{

case "-40":

textBox2.ForeColor = Color.Purple;

textBox3.ForeColor = Color.Purple;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Extremely Cold Day\r\n (and the same number!)";

break;

case "-18":

textBox2.ForeColor = Color.Blue;

textBox3.ForeColor = Color.Blue;

textBox3.Font = new Font(textBox3.Font, FontStyle.Regular);

textBox3.Text = "Very Cold Day";

break;

case "0":

textBox2.ForeColor = Color.Blue;

textBox3.ForeColor = Color.Blue;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Freezing point of water";

break;

case "10":

textBox2.ForeColor = Color.LightBlue;

textBox3.ForeColor = Color.LightBlue;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Cool Day";

break;

case "21":

textBox2.ForeColor = Color.Green;

textBox3.ForeColor = Color.Green;

textBox3.Font = new Font(textBox3.Font, FontStyle.Regular);

textBox3.Text = "Room temperature";

break;

case "30":

textBox2.ForeColor = Color.Orange;

textBox3.ForeColor = Color.Orange;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Beach weather";

break;

case "37":

textBox2.ForeColor = Color.Orange;

textBox3.ForeColor = Color.Orange;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Body temperature";

break;

case "40":

textBox2.ForeColor = Color.Red;

textBox3.ForeColor = Color.Red;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Hot Bath";

break;

case "100":

textBox2.ForeColor = Color.DarkRed;

textBox3.ForeColor = Color.DarkRed;

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

textBox3.Text = "Water Boils";

break;

default:

textBox3.Clear();

break;

}

FileStream fs3 = null;

try

{

fs3 = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs3);

textOut.Write(textBox1.Text.Trim() + "°F = ");

textOut.Write(textBox2.Text.Trim() + "°C, ");

textOut.Write(DateTime.Now.ToString("yyyy/MM/dd h:mm:ss tt "));

textOut.WriteLine(textBox3.Text.Trim() + " ");

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

{ MessageBox.Show(ex.Message, "IOException"); }

finally { if (fs3 != null) fs3.Close(); }

}

}

}

catch (FormatException)

{

MessageBox.Show("Please enter a valid temperature.");

}

}

private void radioButton1\_CheckedChanged\_1(object sender, EventArgs e)

{

label2.Text = "°C";

label4.Text = "°F";

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

label2.Text = "°F";

label4.Text = "°C";

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void TemperatureConversion\_Load(object sender, EventArgs e)

{

startTime = DateTime.Now;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace frmDashboard

{

internal class Class

{

//properties

private decimal currentValue; //A decimal that stores the result currently displayed by the calculator.

private decimal operand1; //A decimal that stores the value of the first operand

private decimal operand2; //A decimal that stores the value of the second operand

private string op; //A string type that stores the value of the operator

public decimal CurrentValue

{

get { return currentValue; }

set { currentValue=value; }

}

public decimal Operand1

{

get { return operand1; }

set { operand1 = value; }

}

public decimal Operand2

{

get { return operand2; }

set { operand2 = value; }

}

public string Op

{

get { return op; }

set { op = value; }

}

public void Add(string displayValue)

{

operand1 = Convert.ToDecimal(displayValue);

currentValue =operand1;

op = "+";

}

public void Subtract(string displayValue)

{

operand1 = Convert.ToDecimal(displayValue);

currentValue = operand1;

op = "-";

}

public void Multiply(string displayValue)

{

operand1 = Convert.ToDecimal(displayValue);

currentValue = operand1;

op = "\*";

}

public void Divide(string displayValue)

{

operand1 = Convert.ToDecimal(displayValue);

currentValue = operand1;

op = "/";

}

public string Equals(string displayValue)

{

//operand2 = Convert.ToDecimal(displayValue);

//currentValue = DoOperation();

//return displayValue;

operand2 = Convert.ToDecimal(displayValue);

switch (op)

{

case "+":

operand1 += operand2;

break;

case "-":

operand1 -= operand2;

break;

case "\*":

operand1 \*= operand2;

break;

case "/":

if (operand2 == 0)

{

MessageBox.Show("Cannot divide by zero.");

}

operand1 /= operand2;

break;

default:

break;

}

return operand1.ToString();

}

public void Clear()

{

currentValue = 0;

operand1 = 0;

operand2 = 0;

op = null;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Text;

using System.IO;

using System.Linq;

using System.Reflection.Emit;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

/\*name=Preet Manandhar

This from is for calculations

i have used class and methods for calculations with private fields, constructor, properties and methods

i tried my best to display result but i couldnot get the result as expected.

\*/

namespace frmDashboard

{

public partial class Calculator : Form

{

Class calc = new Class();

public Calculator()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "1";

}

else

{

textBox1.Text = textBox1.Text + "1";

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "2";

}

else

{

textBox1.Text = textBox1.Text + "2";

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "3";

}

else

{

textBox1.Text = textBox1.Text + "3";

}

}

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

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "4";

}

else

{

textBox1.Text = textBox1.Text + "4";

}

}

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

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "5";

}

else

{

textBox1.Text = textBox1.Text + "5";

}

}

private void button6\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "6";

}

else

{

textBox1.Text = textBox1.Text + "6";

}

}

private void button7\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "7";

}

else

{

textBox1.Text = textBox1.Text + "7";

}

}

private void button8\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "8";

}

else

{

textBox1.Text = textBox1.Text + "8";

}

}

private void button9\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "9";

}

else

{

textBox1.Text = textBox1.Text + "9";

}

}

private void button10\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = "0";

}

else

{

textBox1.Text = textBox1.Text + "0";

}

}

private void button11\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "0" && textBox1.Text != null)

{

textBox1.Text = ".";

}

else

{

textBox1.Text = textBox1.Text + ".";

}

}

private void Calculator\_Load(object sender, EventArgs e)

{

// Set the start time to the current time

startTime = DateTime.Now;

}

private void button14\_Click(object sender, EventArgs e)

{

calc.Subtract(textBox1.Text);

textBox1.Text = "";

}

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

{

calc.Multiply(textBox1.Text);

textBox1.Text = "";

}

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

{

calc.Divide(textBox1.Text);

textBox1.Text = "";

}

string path= @".\Calculator.txt";

private void button17\_Click(object sender, EventArgs e)

{

textBox1.Text = calc.Equals(textBox1.Text);

FileStream fs1 = null;

try

{

string textin = textBox1.Text.Trim();

fs1 = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs1);

textOut.WriteLine( calc.CurrentValue +" " + calc.Op + " " + calc.Operand2 + " = "+ calc.Operand1 );

textOut.Close();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "IO Exception"); }

finally { if (fs1 != null) fs1.Close(); }

}

private DateTime startTime;

private DateTime endTime;

private void button18\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit this calculator?", "Exit ?", MessageBoxButtons.YesNo).ToString() == "Yes")

{

// Set the end time to the current time

endTime = DateTime.Now;

// Calculate the time difference in seconds and minutes

TimeSpan timeDiff = endTime - startTime;

int seconds = (int)timeDiff.TotalSeconds;

int minutes = (int)timeDiff.TotalMinutes;

// Display the time on the form using a MessageBox

MessageBox.Show($"Time spent on form: {minutes} minutes ({seconds} seconds )");

this.Close();

}

}

//clear botton

private void button12\_Click(object sender, EventArgs e)

{

calc.Clear();

textBox1.Text = "";

}

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

{

calc.Add(textBox1.Text);

textBox1.Text = "";

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

/\*NAME = Preet Manandhar

This form is about validtation of ip addresss that should be between 0 to 255

i have used regex for ip address

the button reset resets the io address and close form with display of load of time this form took\*/

namespace frmDashboard

{

public partial class Ip4Validator : Form

{

public Ip4Validator()

{

InitializeComponent();

}

string pathBinary = @".\Ip4Binary.txt";

private DateTime startTime;

private DateTime endTime;

private void validate\_botton\_Click(object sender, EventArgs e)

{

string ip = textbox.Text.Trim();

Regex address = new Regex(@"^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$"); //regex for integer between 0 to 255

if(address.IsMatch(ip)== true)

{

try

{

FileStream fs = new FileStream(pathBinary, FileMode.Append, FileAccess.Write);

// create the output stream for a binary file that exists

BinaryWriter binaryOut = new BinaryWriter(fs);

// write the fields into text file

binaryOut.Write(ip);

binaryOut.Write(DateTime.Now.ToLongDateString());

// close the output stream for the text file

binaryOut.Close();

MessageBox.Show(textbox.Text + "\nThe IP address is correct", "Valid IP");

}

catch(Exception ex) { MessageBox.Show(ex.Message); }

}

else

{

MessageBox.Show(textbox.Text + "\nThe IP must have 4 bytes \n Integer number between 0 to 255 \n Separated by a dot (255.255.255.255)","Error" );

}

}

private void reset\_button\_Click(object sender, EventArgs e)

{

//this resets the entered data in textbox with empty string

textbox.Text = "";

}

private void exit\_botton\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit this application IP Validator?", "Exit ?", MessageBoxButtons.YesNo).ToString() == "Yes")

{

// Set the end time to the current time

endTime = DateTime.Now;

// Calculate the time difference in seconds and minutes

TimeSpan timeDiff = endTime - startTime;

int seconds = (int)timeDiff.TotalSeconds;

int minutes = (int)timeDiff.TotalMinutes;

// Display the time on the form using a MessageBox

MessageBox.Show($"Time spent on form: {minutes} minutes ({seconds} seconds )");

this.Close();

}

}

private void Ip4Validator\_Load(object sender, EventArgs e)

{

// Set the start time to the current time

startTime = DateTime.Now;

label2.Text = " " + DateTime.Now.ToString("dd,-MM-yy");

}

}

}

1. **Present the classes and/or methods that you create or you did use in the project.**

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| Class in calculator  currentValue  operand1 operand2 | it has private fields, constructor, properties and methods…….  A decimal that stores the result currently displayed by the calculator.  A decimal that stores the value of the first operand.  A decimal that stores the value of the second operand. op A string type that stores the value of the operator () |
| Constructor | Creates a Calculator object with default values. The default value for the op field is Null. |
| Property CurrentValue | Gets the value of the currentValue field. |
| Add(displayValue)  Subtract(displayValue)  Multiply(displayValue)  Divide(displayValue)  Equals(displayValue)  Clear() | Sets the operand1 and currentValue fields to the value that’s passed to it and sets the op field to "+".  Sets the operand1 and currentValue fields to the value that’s passed to it and sets the op field to "-".  Sets the operand1 and currentValue fields to the value that’s passed to it and sets the op field to "\*".  Sets the operand1 and currentValue fields to the value that’s passed to it and sets the op field to "/".  Sets the operand2 field to the value that’s passed to it. Then, performs the operation specified by the op field on the operand1 and operand2 fields, and stores the result in the operand1 field.  Sets the private fields to their default values. |

1. **Present the difficulties that you have, what was the hardest and the easiest part of your project.**

… calculator was hardest form me and first 2 forms were easy .

It was difficult t o connect with class