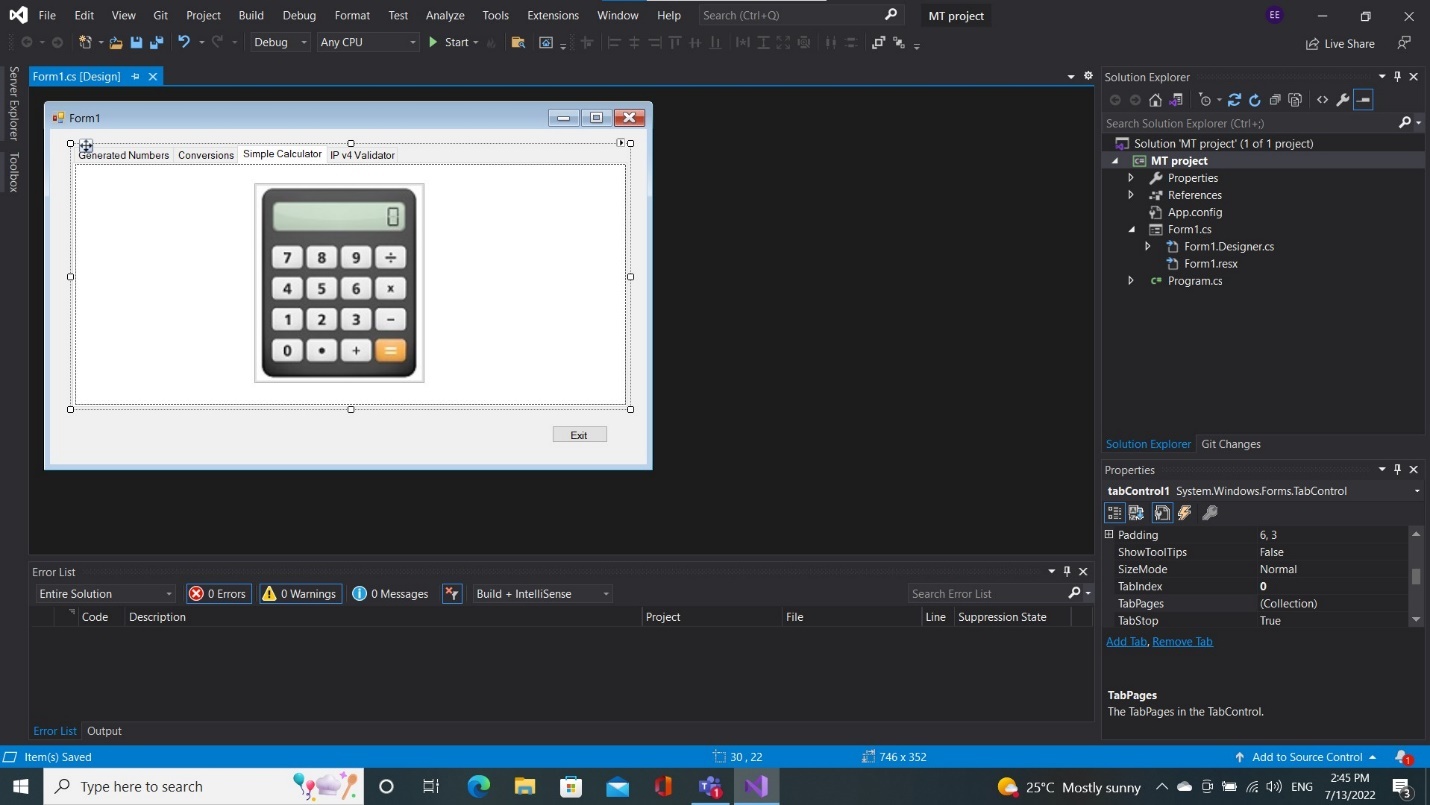
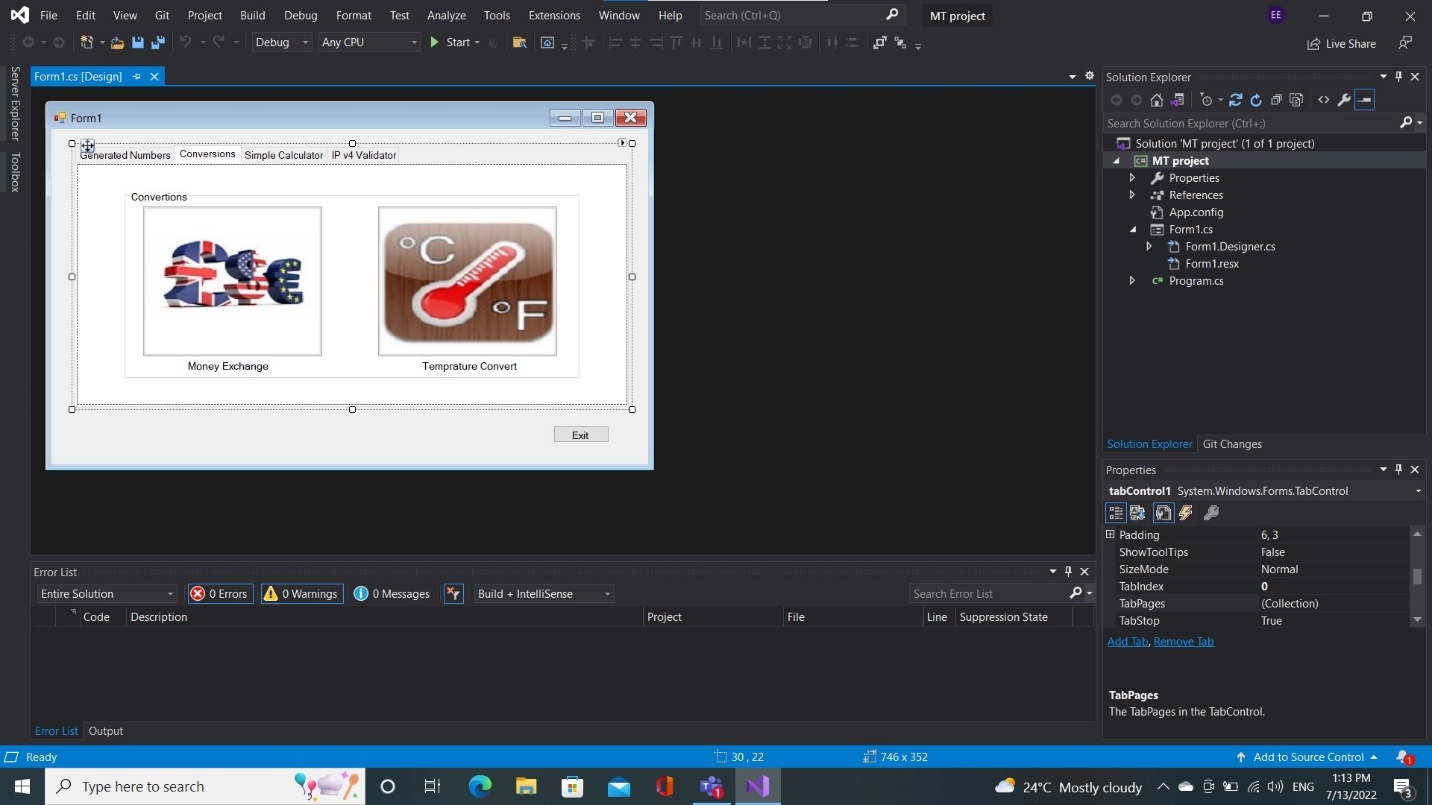
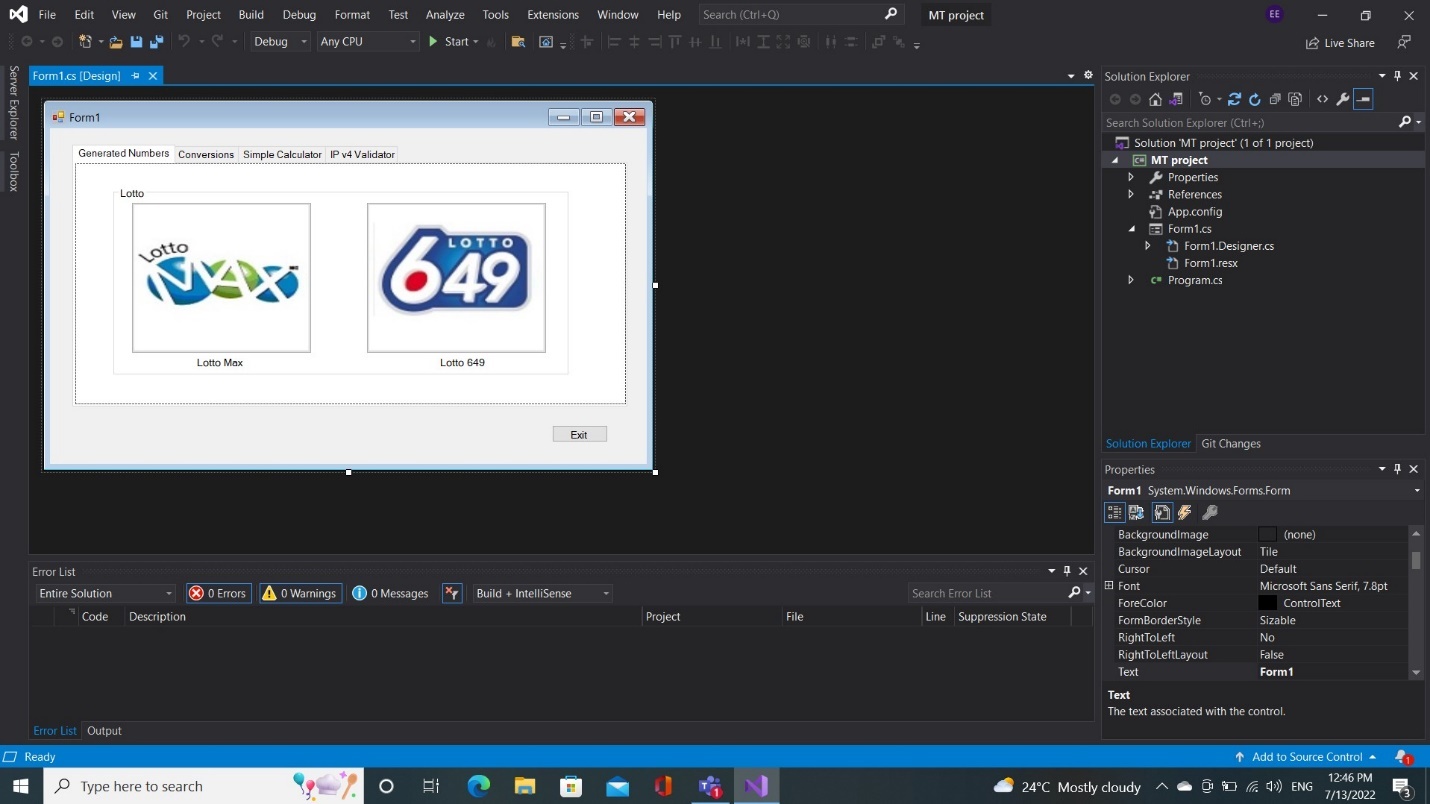
|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

|  |
| --- |
| Your name: Essam elfakharany  7/25/2022  Version 1.0 |

1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language(s) C#
3. tool Visual Studio version ( 2019 )
4. **Present the print screens of yours forms, and have a detailed description of the functionalities (step by step).**
5. Graphical user interface

   Description automatically generatedGraphical user interface, application

   Description automatically generatedGraphical user interface, application

   Description automatically generatedGraphical user interface

   Description automatically generatedGraphical user interface

   Description automatically generatedGraphical user interface

   Description automatically generatedGraphical user interface

   Description automatically generatedGraphical user interface

   Description automatically generatedGraphical user interface

   Description automatically generatedGraphical user interface

   Description automatically generatedGraphical user interface, website

   Description automatically generated
6. If you click on tab Generated Numbers will go to (Lotto Max- Lotto649)
7. If you click on tab Conversions will go to (MoneyEx- Temperature Convertor)
8. If you click on tab Simple Calculator will go to Calculator App.
9. If you click on tab IP V4 Validator will go to IP V4 Validator App
10. If you click on the Exit button, Will Exit the program.
11. If you click on lottoMax button it will direct you to lottomax form.
12. If you click on lotto649 button it will direct you to lotto649 form.
13. If you click on Money Ex button it will direct you to Money exchange form.
14. If you click on Temp temperature button it will direct you to temperature form.
15. If you click on Calculator button it will direct you to The calculator form.
16. If you click on IP4 validator button it will direct you to IP4 Validator form.
17. **Present the code of your application (forms).**

**frmLottoMax**

public partial class frmMax : Form

{

static string dirPath = @".\Files\";

static string filePath = dirPath + "frmx.txt";

public frmMax()

{

InitializeComponent();

}

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

{

if (MessageBox.Show("Do you want to quit the App? ",

"Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

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

{

string displayNbrs = "";

Random random = new Random();

int randomNumber = 0;

for (int i = 0; i < 8; i++)

{

randomNumber = random.Next(1, 50); //random.Next(1, 50); for MAX

displayNbrs += randomNumber.ToString() + " ";

}

textBox1.Text = displayNbrs;

FileStream fs = null;

try

{

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

StreamWriter textOut = new StreamWriter(fs);

string textTofile = textBox1.Text + " " + "\t" + DateTime.Now + " ";

textOut.WriteLine(textTofile);// write in new line

textOut.Close();

// code that uses the file stream to read and write data file

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

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

{

FileStream fs = null;

try

{

fs = new FileStream(filePath, FileMode.Open, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToDisplay = "";

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

{

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

}

MessageBox.Show(textToDisplay, "frmMax- Essam");

textIn.Close();

}

catch (IOException ex)

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

finally

{

if (fs != null)

{

fs.Close();

}

}

}

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

{

if (!Directory.Exists(dirPath))

Directory.CreateDirectory(dirPath);

}

// catch (Exception ex)

//{

// MessageBox.Show(ex.Message);//Ex

// textBox1.Clear();//delete the txtbox

// textBox1.Focus();//back to the clear txt

}

}

**frmLotto649**

public partial class frmMax : Form

{

static string dirPath = @".\Files\";

static string filePath = dirPath + "frmx.txt";

public frmMax()

{

InitializeComponent();

}

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

{

if (MessageBox.Show("Do you want to quit the App? ",

"Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

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

{

string displayNbrs = "";

Random random = new Random();

int randomNumber = 0;

for (int i = 0; i < 8; i++)

{

randomNumber = random.Next(1, 50); //random.Next(1, 50); for MAX

displayNbrs += randomNumber.ToString() + " ";

}

textBox1.Text = displayNbrs;

FileStream fs = null;

try

{

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

StreamWriter textOut = new StreamWriter(fs);

string textTofile = textBox1.Text + " " + "\t" + DateTime.Now + " ";

textOut.WriteLine(textTofile);// write in new line

textOut.Close();

// code that uses the file stream to read and write data file

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

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

{

FileStream fs = null;

try

{

fs = new FileStream(filePath, FileMode.Open, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToDisplay = "";

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

{

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

}

MessageBox.Show(textToDisplay, "frmMax- Essam");

textIn.Close();

}

catch (IOException ex)

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

finally

{

if (fs != null)

{

fs.Close();

}

}

}

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

{

if (!Directory.Exists(dirPath))

Directory.CreateDirectory(dirPath);

}

// catch (Exception ex)

//{

// MessageBox.Show(ex.Message);//Ex

// textBox1.Clear();//delete the txtbox

// textBox1.Focus();//back to the clear txt

}

**frmMoneyEx**

{

public partial class MoneyEX : Form

{

public MoneyEX()

{

InitializeComponent();

}

string fromName = "";

string toName = ""; // holding the name of the currency

double fromCurrency = 0;

double toCurrency = 0;

static string dirPath = @".\Files\";

static string filePath = dirPath + "moneyEX.txt";

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

{

if (!Directory.Exists(dirPath))

Directory.CreateDirectory(dirPath);

}

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

{

Checkbtn();// call the function

if (txtFrom.Text == null || txtFrom.Text.Trim() == "")

{

MessageBox.Show("please Enter a valid amount");

}

// MessageBox.Show(fromCurrency + "\t" + toCurrency);

try

{

double amount = Convert.ToDouble(txtFrom.Text.Trim());//

txtto.Text = (fromCurrency \* amount / toCurrency).ToString("0.00");// 2 decemel points

FileStream fs = null;

try

{

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

StreamWriter textOut = new StreamWriter(fs);

string textTofile = txtFrom.Text + " " + " = " + txtto.Text + " " + "\t" + DateTime.Now ;

textOut.WriteLine(textTofile);// write in new line

textOut.Close();

// code that uses the file stream to read and write data file

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

txtFrom.Clear();

txtFrom.Focus();

}

}

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

{

FileStream fs = null;

try

{

fs = new FileStream(filePath, FileMode.Open, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToDisplay = "";

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

{

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

}

MessageBox.Show(textToDisplay, "MoneyEX- Essam");

textIn.Close();

}

catch (IOException ex)

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

finally

{

if (fs != null)

{

fs.Close();

}

}

}

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

{

if (MessageBox.Show("Do you want to quit the App? ",

"Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

private void Checkbtn()// function to check RB

{

double CAD = 1;

double USD = 1.28807;

double EUR = 1.30486;

double GBP = 1.54959;

double EGP = 0.0679411;//Egyption pound

if (rbcadf.Checked)

{

fromName = rbcadf.Text;

fromCurrency = CAD;

}

else if (rbusdf.Checked)

{

fromName = rbusdf.Text;

fromCurrency = USD;

}

else if (rbeurf.Checked)

{

fromName = rbeurf.Text;

fromCurrency = EUR;

}

else if (rbgbpf.Checked)

{

fromName = rbgbpf.Text;

fromCurrency = GBP;

}

else if (rbegpf.Checked)

{

fromName = rbegpf.Text;

fromCurrency = EGP;

}

if (rbcadt.Checked)

{

toName = rbcadt.Text;

toCurrency = CAD;

}

else if (rbusdt.Checked)

{

toName = rbusdt.Text;

toCurrency = USD;

}

else if (rbeurt.Checked)

{

toName = rbeurt.Text;

toCurrency = EUR;

}

else if (rbgbpt.Checked)

{

toName = rbgbpt.Text;

toCurrency = GBP;

}

else if (rbegpt.Checked)

{

toName = rbegpt.Text;

toCurrency = EGP;

}

}

}

**frmTempConversion**

namespace MT\_project

{

public partial class tempConversion : Form

{

public tempConversion()

{

InitializeComponent();

}

double celsius;

double fahrenhit;

static string dirPath = @".\Files\";

static string filePath = dirPath + "temp.txt";

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

{

if (!Directory.Exists(dirPath))

Directory.CreateDirectory(dirPath);

}

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

{

txtMessage.Clear();

if (txtFromTemp.Text== null || txtFromTemp.Text.Trim()=="")

{

MessageBox.Show("please Enter a valid temperature");

}

try

{

if (rbCtoF.Checked)

{

celsius = Convert.ToInt32(txtFromTemp.Text);

fahrenhit = ((celsius \* 1.8) + 32);

txtToTemp.Text = fahrenhit.ToString("0.##");

Messages();

}

else

{

fahrenhit = Convert.ToInt32(txtFromTemp.Text);

celsius = ((fahrenhit - 32) \* 5 / 9);

txtToTemp.Text = celsius.ToString("0.##");

Messages();

}

FileStream fs = null;

try

{

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

StreamWriter textOut = new StreamWriter(fs);

string textTofile = txtFromTemp.Text + " " + lblFromTemp.Text + " = " + txtToTemp.Text + " " + lblToTemp.Text + "\t" + DateTime.Now + " " + txtMessage.Text;

textOut.WriteLine(textTofile);// write in new line

textOut.Close();

// code that uses the file stream to read and write data file

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);//Ex

txtFromTemp.Clear();//delete the txtbox

txtFromTemp.Focus();//back to the clear txt

}

}

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

{

FileStream fs = null;

try

{

fs = new FileStream(filePath, FileMode.Open, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToDisplay = "";

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

{

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

}

MessageBox.Show(textToDisplay, "tempconversion- Essam");

textIn.Close();

}

catch (IOException ex)

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

finally

{

if (fs != null)

{

fs.Close();

}

}

}

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

{

if (MessageBox.Show("Do you want to quit the App? ",

"Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

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

{

// switch C to F and reverse with radoibuttons check

if (rbCtoF.Checked)

{

lblFromTemp.Text = "C";

lblToTemp.Text = "F";

}

else

{

lblFromTemp.Text = "F";

lblToTemp.Text = "C";

}

}

private void Messages()//method for messages

{

//Messages

if (celsius == 100 )

{

txtMessage.Text = " Water boils";

}

else if (celsius == 40)

{

txtMessage.Text = " Hot bath";

}

else if ( celsius==37)

{

txtMessage.Text = " Body temperature";

}

else if (celsius == 30)

{

txtMessage.Text = "Beach weather";

}

else if (celsius >= 21&& celsius<22)

{

txtMessage.Text = "Room temperutre";

}

else if (celsius ==10)

{

txtMessage.Text = " Cool Day";

}

else if (celsius == 0)

{

txtMessage.Text = " Freezing point";

}

else if (celsius >=-18 && celsius<-17)

{

txtMessage.Text = " Very Cold Day";

}

else if (celsius == -40)

{

txtMessage.Text = "Extremely cold Day";

}

if (celsius >= 40)

{

txtFromTemp.ForeColor = Color.Red;

txtToTemp.ForeColor = Color.Red;

}

else if (celsius < 40 && celsius >= 37)

{

txtFromTemp.ForeColor = Color.Orange;

txtToTemp.ForeColor = Color.Orange;

}

else if (celsius < 37 && celsius > 21)

{

txtFromTemp.ForeColor = Color.Green;

txtToTemp.ForeColor = Color.Green;

}

else if (celsius <= 10)

{

txtFromTemp.ForeColor = Color.Blue;

txtToTemp.ForeColor = Color.Blue;

}

}

**frmCalculator**

namespace MT\_project

{

public partial class simpleCalc : Form

{

public simpleCalc()

{

InitializeComponent();

}

CalcLog objcalc = new CalcLog();

static string dirPath = @".\Files\";

static string filePath = dirPath + "Calc.txt";

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

{

if (!Directory.Exists(dirPath))

Directory.CreateDirectory(dirPath);

}

private void btnN1\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != “”)

{

txtbScr.Text = "1";//Replace 0 with 1 if the screen is 0

}

else

{

txtbScr.Text += "1";//add the 1 to the screen

}

}

private void btnN2\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "2";

}

else

{

txtbScr.Text += "2";

}

}

private void btnN3\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "3";

}

else

{

txtbScr.Text += "3";

}

}

private void btnN4\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "4";

}

else

{

txtbScr.Text += "4";

}

}

private void btnN5\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "5";

}

else

{

txtbScr.Text += "5";

}

}

private void btnN6\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "6";

}

else

{

txtbScr.Text += "6";

}

}

private void btnN7\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "7";

}

else

{

txtbScr.Text += "7";

}

}

private void btnN8\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "8";

}

else

{

txtbScr.Text += "8";

}

}

private void btnN9\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "9";

}

else

{

txtbScr.Text += "9";

}

}

private void btnN0\_Click(object sender, EventArgs e)

{

if (txtbScr.Text == "0")//&& txtbScr.Text != null)

{

txtbScr.Text = "0";

}

else

{

txtbScr.Text += "0";

}

}

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

{

if (txtbScr.Text.Contains('.'))// if there is Decimel point don't add any

{

return;

}

else

{

txtbScr.Text += ".";

}

}

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

{

if (objcalc.Op != null)

{

return;

}

else

{

objcalc.Operand1 = Convert.ToDouble(txtbScr.Text);

objcalc.Op = "+";

txtbScr.Text = txtbScr.Tag.ToString();//the defult text (0)

}

}

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

{

if (objcalc.Op != null)

{

return;

}

else

{

objcalc.Operand1 = Convert.ToDouble(txtbScr.Text);

objcalc.Op = "-";

txtbScr.Text = txtbScr.Tag.ToString();//the defult text (0)

}

}

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

{

if (objcalc.Op != null)

{

return;

}

else

{

objcalc.Operand1 = Convert.ToDouble(txtbScr.Text);

objcalc.Op = "\*";

txtbScr.Text = txtbScr.Tag.ToString();//the defult text (0)

}

}

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

{

if (objcalc.Op != null)

{

return;

}

else

{

objcalc.Operand1 = Convert.ToDouble(txtbScr.Text);

objcalc.Op = "/";

txtbScr.Text = txtbScr.Tag.ToString();//the defult text (0)

}

}

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

{

if (objcalc.Op != null)

{

objcalc.Operand2 = Convert.ToDouble(txtbScr.Text);

txtbScr.Text = objcalc.Eql().ToString();

Write();

objcalc.Op = null;// back to "0" or reset after finish the operation

}

else

{

return;//go back don't excute.

}

}

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

{

txtbScr.Text = txtbScr.Tag.ToString();//

}

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

{

if (MessageBox.Show("Do you want to quit the App? ",

"Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

private void Write()

{

FileStream fs = null;

try

{

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

StreamWriter textOut = new StreamWriter(fs);

string textTofile = objcalc.Operand1.ToString() + " " + objcalc.Op + " " + objcalc.Operand2.ToString() + " = " + objcalc.CurrentValue.ToString() + " " + DateTime.Now;

textOut.WriteLine(textTofile);// write in new line

textOut.Close();

// code that uses the file stream to read and write data file

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

}

**frmIP4Validator**

namespace MT\_project

{

public partial class IP4\_Valiidator : Form

{

public IP4\_Valiidator()

{

InitializeComponent();

}

static string dirPath = @".\Files\";

static string filePath = dirPath + "IP4.dat";

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

{

txtIP.Clear();//rest//method

}

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

{

if (MessageBox.Show("Do you want to quit the App? ",

"Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

private void IP4\_Valiidator\_Load(object sender, EventArgs e)

{

lblDate.Text += DateTime.Today.ToLongDateString();

if (!Directory.Exists(dirPath))

Directory.CreateDirectory(dirPath);

}

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

{

string IP4 = txtIP.Text.ToString().Trim();

string IP = @"^([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(\.([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3}$";

Regex check = new Regex(IP);

if (string.IsNullOrEmpty(IP4))

{

MessageBox.Show(IP4 + "Please enter IP");

}

else if (check.IsMatch(IP4, 0))

{

MessageBox.Show(IP4 + "IP is Valid");

Binary();

}

else if (!check.IsMatch(IP, 0))

{

MessageBox.Show(IP4 + "\nThe IP must have 4 bytes" + "\ninteger number between 0 to 255" + "\nseparated by a dot(255.255.255.255)", "Error");

}

else

{

MessageBox.Show(IP4 + "IP is not Valid");

}

}

private void Binary()

{

FileStream fs = null;

try

{

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

if (txtIP.Text != "")

{

BinaryWriter binaryOut = new BinaryWriter(fs);

binaryOut.Write(txtIP.Text);

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

binaryOut.Close();

}

}

catch (IOException ex)

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

finally

{

if (fs != null)

{

fs.Close();

}

}

}//metod

}

}

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

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Class calcLog | Class where the obj of the calculator and where the logical operation methods are created. |
| 1. Double Eql() | For Equal |
| 1. Double Div() | For Division |
| 1. Double mlt() | For Multiply |
| 1. Double sub() | For Subtract |
| 1. Double Add () | For addition |
| 1. Void Write() | To use it for writing the txt files |
| 1. Void Binary() | To use it for writing the .dat files |
| 1. void Messages() | For temperature app massages. Show Messages |

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

All wasn’t easy and I had to review a lot before finish it.