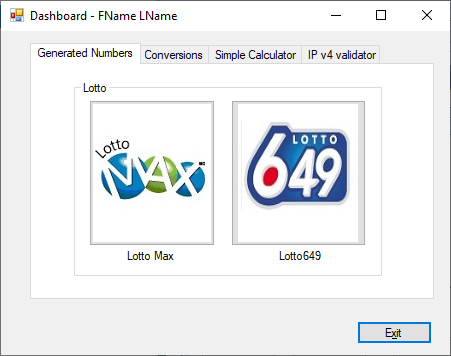
|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

|  |
| --- |
| Your name:  0/0/2019 |

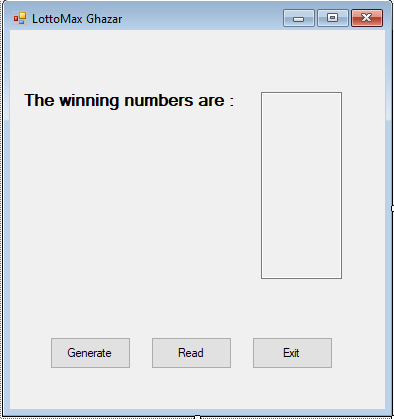
1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language C# Windows form

Microsoft Visual Studio version 2019

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



1. LOTTO MAX image:



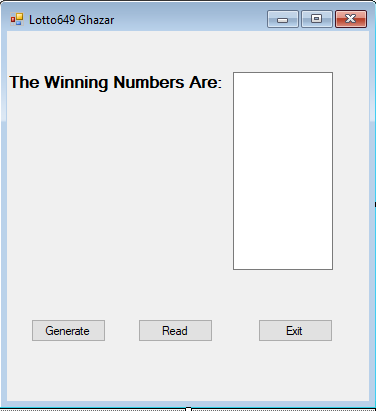
1. IT will display a text box with three buttons:

Button generate: it will generate 7+1 random numbers and display them on the textbox

Button Read : it will show a history of all the numbers generated with the date

Button Exit: it will exit from lotto max form

B – 649 LOTTO



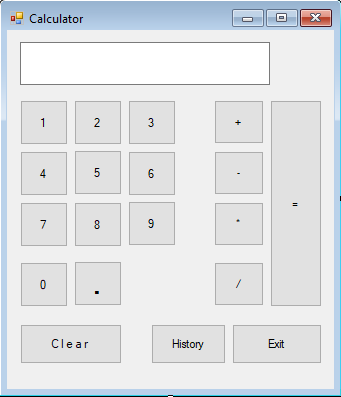
1. IT will display a text box with three buttons:

Button generate: it will generate 6+1 random numbers and display them on the textbox

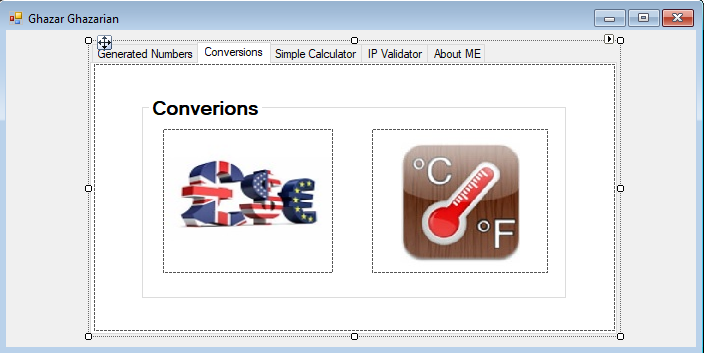
Button Read : it will show a history of all the numbers generated with the date

Button Exit: it will exit from lotto max form

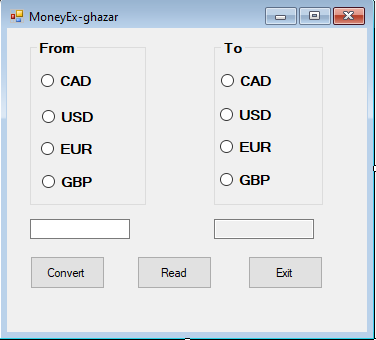
-Form Calculator:



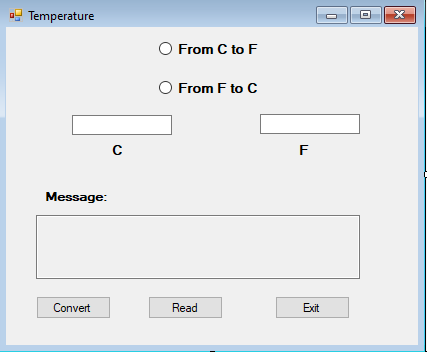
1. It will display buttons from 1-9 that will capture the numbers entered by the user to start the calculations
2. Operation buttons(+,-,\*,/)
3. Button clear that will clear the textbox and be ready for another calculation
4. Button history : will display all the calculations done and the date .
5. Button Exit: it will exit from the form



-FORM CONVERIONS:

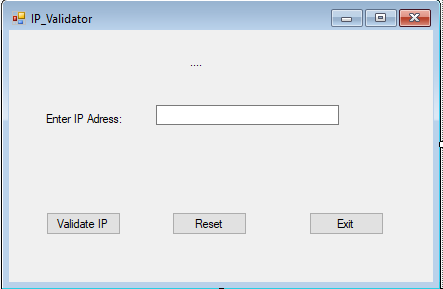


1. MONEY EXCHANGE APP.
2. From Radio buttons with the currency of corresponding country
3. To Radio button with the currency of corresponding county
4. Convert button that will convert one currency chosen from radio buttons to another currency from the To radio buttons
5. Read: a history of money exchanged
6. Exit: Exit the moneyex app



2-temerature app:

1. Two radio buttons that a user should choose one to convert the temperature from C to F .
2. Convert : it will display the conversion in respond to the chosen operation
3. Read: a history of the conversions
4. Exit: exit the temperature app.
5. Message box : it will display a brief message related to that current temperature shown in the textbox



IP validator app:

1. Validate button: it will check if the IP address is in the right format (255-255-255-255)
2. Reset : it will reset the textbox and start another operation
3. Exit: it will exit from the ip validator app.
4. **Present the code of your application (forms).**

* LOTTO MAX:
* 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;
* using System.IO;
* namespace WindowsFormsProject420\_CT2
* {
* public partial class LottoMax : Form
* {
* DateTime currentDateTime = DateTime.Now;
* FileStream fs = null;
* string path = @".\lotto.txt";
* public LottoMax()
* {
* InitializeComponent();
* }
* private void button1\_Click(object sender, EventArgs e)
* {
* fs = new FileStream(path, FileMode.Append, FileAccess.Write);
* StreamWriter textOut = new StreamWriter(fs);
* Random random = new Random();
* int value;
* string[] g = new string[8];
* //int count = 0;
* int i;
* for (i = 0; i < 8; i++)
* {
* value = random.Next(1, 49);
* MaxBox.Text += value.ToString() + "\r\n";
* g[i] = value.ToString();
* }
* textOut.Write("Max"+"\t" + "|");
* for (i = 0; i < 7; i++)
* {
* textOut.Write(g[i] + ",");
* }
* textOut.Write(" Extra (" + g[7] + ")" + "|");
* textOut.WriteLine(" " + currentDateTime);
* textOut.Close();
* }
* private void button3\_Click(object sender, EventArgs e)
* {
* byte btnVal = 0;
* btnVal = Convert.ToByte(MessageBox.Show("Do you want to Exit?", "lotto649", MessageBoxButtons.OKCancel));
* if (btnVal == 1)
* {
* this.Close();
* }
* }
* private void button2\_Click(object sender, EventArgs e)
* {
* string textToPrint = "";
* fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);
* StreamReader textIn = new StreamReader(fs);
* // read the data from the file and store it in the list
* while (textIn.Peek() != -1)
* {
* string row = textIn.ReadLine();
* string[] columns = row.Split('|');
* textToPrint += columns[0] +columns[1] + columns[2] + "\n";
* }
* MessageBox.Show(textToPrint,"All Winning Numbers");
* // close the input stream for the text file
* textIn.Close();
* }
* }
* }
* 649 lotto:
* 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;
* using System.IO;
* namespace WindowsFormsProject420\_CT2
* {
* public partial class Lotto649 : Form
* {
* DateTime currentDateTime = DateTime.Now;
* FileStream fs = null;
* string path = @".\lotto.txt";
* public Lotto649()
* {
* InitializeComponent();
* }
* private void button1\_Click(object sender, EventArgs e)
* {
* fs = new FileStream(path, FileMode.Append, FileAccess.Write);
* StreamWriter textOut = new StreamWriter(fs);
* Random random = new Random();
* int value;
* string[] g = new string[8];
* //int count = 0;
* int i;
* for (i = 0; i < 7; i++)
* {
* value = random.Next(1, 49);
* showbox.Text += value.ToString() + "\r\n";
* g[i] = value.ToString();
* }
* textOut.Write("649" + "\t" + "|");
* for (i = 0; i < 6; i++)
* {
* textOut.Write(g[i] + ",");
* }
* textOut.Write(" Extra (" + g[6] + ")" + "|");
* textOut.WriteLine(" " + currentDateTime);
* textOut.Close();
* }
* private void button2\_Click(object sender, EventArgs e)
* {
* string textToPrint = "";
* fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);
* StreamReader textIn = new StreamReader(fs);
* // read the data from the file and store it in the list
* while (textIn.Peek() != -1)
* {
* string row = textIn.ReadLine();
* string[] columns = row.Split('|');
* textToPrint += columns[0] + columns[1] + columns[2] + "\n";
* }
* MessageBox.Show(textToPrint, "All Winning Numbers");
* // close the input stream for the text file
* textIn.Close();
* }
* private void button3\_Click(object sender, EventArgs e)
* {
* byte btnVal = 0;
* btnVal = Convert.ToByte(MessageBox.Show("Do you want to Exit?", "lottoMax", MessageBoxButtons.OKCancel));
* if (btnVal == 1)
* {
* this.Close();
* }
* }
* }
* }
* Calculator:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Linq.Expressions;

namespace WindowsFormsProject420\_CT2

{

public partial class Calculator : Form

{

Cal obj= new Cal();

public Calculator()

{

InitializeComponent();

}

private void button13\_Click(object sender, EventArgs e)

{

byte btnVal = 0;

btnVal = Convert.ToByte(MessageBox.Show("Do you want to Exit?", "Calculator", MessageBoxButtons.OKCancel));

if (btnVal == 1)

{

this.Close();

}

}

private void button\_click(object sender, EventArgs e)

{

Button b = (Button)sender;

result.Text =result.Text + b.Text;

}

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

{

result.Clear();

obj.CurrentValue = 0;

label.Text = "";

}

private void op\_click(object sender, EventArgs e)

{

Button b = (Button)sender;

obj.Op = b.Text;

try

{

obj.Operand1 = double.Parse(result.Text);

}

catch(Exception ex1)

{

MessageBox.Show(ex1.Message);

result.Text = "";

}

label.Text =result.Text+ obj.Op;

result.Text = "";

}

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

{

try { obj.Operand2 = double.Parse(result.Text); }

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

result.Text = "";

}

switch (obj.Op)

{

case "+":

result.Text = obj.Add();

break;

case"-":

result.Text = obj.Sub();

break;

case "\*":

result.Text = obj.Mult();

break;

case "/":

result.Text = obj.Div();

break;

default:

break;

}

label.Text = "";

result.Text =obj.CurrentValue.ToString();

obj.Display();

}

private void button19\_Click(object sender, EventArgs e)

{

MessageBox.Show(obj.History(),"History");

}

}

}

* Temperature:
* 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;
* using System.IO;
* namespace WindowsFormsProject420\_CT2
* {
* public partial class Temperature : Form
* {
* string path = @".\Temperature.txt";
* DateTime currentDateTime = DateTime.Now;
* FileStream fs = null;
* double c;
* double f;
* double currentvalue;
* public Temperature()
* {
* InitializeComponent();
* }
* private void button3\_Click(object sender, EventArgs e)
* {
* byte btnVal = 0;
* btnVal = System.Convert.ToByte(MessageBox.Show("Do you want to Exit?", "Temperatue", MessageBoxButtons.OKCancel));
* if (btnVal == 1)
* {
* this.Close();
* }
* }
* private void CtoF\_CheckedChanged(object sender, EventArgs e)
* {
* if (CtoF.Checked)
* {
* Ftextbox.ReadOnly = true;
* Ctextbox.ReadOnly = false;
* Ctextbox.Text = "";
* Ftextbox.Text = "";
* Message.Text = "";
* }
* }
* private void FtoC\_CheckedChanged(object sender, EventArgs e)
* {
* if (FtoC.Checked)
* {
* Ctextbox.ReadOnly = true;
* Ftextbox.ReadOnly = false;
* Ftextbox.Text = "";
* Ctextbox.Text = "";
* Message.Text = "";
* }
* }
* private void Convert\_Click(object sender, EventArgs e)
* {
* fs = new FileStream(path, FileMode.Append, FileAccess.Write);
* StreamWriter textOut = new StreamWriter(fs);
* try
* {
* if (CtoF.Checked)
* {
* c = double.Parse(Ctextbox.Text);
* currentvalue = (c \* 9 / 5) + 32;
* Ftextbox.Text = currentvalue.ToString();
* textOut.Write(c + " C " + "|");
* textOut.Write(" = " + Ftextbox.Text + " F " + "," + "|");
* textOut.WriteLine(currentDateTime);
* }
* else if (FtoC.Checked)
* {
* f = double.Parse(Ftextbox.Text);
* currentvalue = (f - 32) \* 5 / 9;
* Ctextbox.Text = currentvalue.ToString();
* textOut.Write(f + " F " + "|");
* textOut.Write(" = " + Ctextbox.Text + " C " + "," + "|");
* textOut.WriteLine(currentDateTime);
* }
* if (currentvalue == 100 || currentvalue == 212)
* {
* Message.Text = "Water boils";
* }
* else if (currentvalue == 37 || currentvalue == 98.6)
* {
* Message.Text = "Body temperature";
* }
* else if (currentvalue == 40 || currentvalue == 104)
* {
* Message.Text = "Hot Bath";
* }
* else if (currentvalue == 30 || currentvalue == 86)
* {
* Message.Text = "Beach weather";
* }
* else if (currentvalue == 21 || currentvalue == 70)
* {
* Message.Text = "Room temperature";
* }
* else if (currentvalue == 10 || currentvalue == 50)
* {
* Message.Text = "Cool Day";
* }
* else if (currentvalue == 0 || currentvalue == 32)
* {
* Message.Text = "Freezing point of water";
* }
* else if (currentvalue == -18 || currentvalue == 0)
* {
* Message.Text = "Very Cold Day";
* }
* else if (currentvalue == -40 || currentvalue == -40)
* {
* Message.Text = "Extremely Cold Day (and the same number!)";
* }
* else
* {
* Message.Text = "Nothing Special";
* }
* }
* catch (Exception)
* {
* MessageBox.Show("No Value Entered", "error");
* Ftextbox.Text = "";
* Ctextbox.Text = "";
* Message.Text = "";
* }
* finally
* {
* textOut.Close();
* }
* }
* private void button2\_Click(object sender, EventArgs e)
* {
* string textToPrint = "";
* fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);
* StreamReader textIn = new StreamReader(fs);
* // read the data from the file and store it in the list
* while (textIn.Peek() != -1)
* {
* string row = textIn.ReadLine();
* string[] columns = row.Split('|');
* textToPrint += columns[0] + columns[1] + columns[2] + "\n";
* }
* MessageBox.Show(textToPrint, "Temperature Ghazar");
* // close the input stream for the text file
* textIn.Close();
* }
* private void Message\_TextChanged(object sender, EventArgs e)
* {
* }
* }
* }
* Money Exchange:

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;

using System.IO;

namespace WindowsFormsProject420\_CT2

{

public partial class MoneyEx : Form

{

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

DateTime currentDateTime = DateTime.Now;

FileStream fs = null;

string fromchoice ="";

string tochoice ="";

public MoneyEx()

{

InitializeComponent();

}

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

{

MessageBox.Show("Do you want to Exit?", "Calculator", MessageBoxButtons.OKCancel);

}

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

{

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

StreamWriter textOut = new StreamWriter(fs);

//IF THE USER WANTS TO EXCHANGE FROM CANADIAN DOLLOR

if (fromchoice == "CAD")

{

switch (tochoice)

{

case "USD":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 0.75).ToString() + "$";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "EUR":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 0.64).ToString() + " €";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "GBP":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 0.58).ToString() + " £";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "CAD":

MessageBox.Show("Can not convert to the same currency", "Error");

break;

default:

break;

}

}

//IF THE USER WANTS TO EXCHANGE FROM US DOLLAR

else if (fromchoice == "USD")

{

switch (tochoice)

{

case "USD":

MessageBox.Show("Can not convert to the same currency", "Error");

break;

case "EUR":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 0.86).ToString() + " €";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "GBP":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 0.78).ToString() + " £";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "CAD":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 1.34).ToString() + " $";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

default:

break;

}

}

//IF THE USER WANTS TO EXCHANGE FROM EURO

else if (fromchoice == "EUR")

{

switch (tochoice)

{

case "USD":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 1.17).ToString() + "$";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "EUR":

MessageBox.Show("Can not convert to the same currency", "Error");

break;

case "GBP":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 0.91).ToString() + " £";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "CAD":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 1.56).ToString() + " £";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

default:

break;

}

}

//IF THE USER WANTS TO EXCHANGE FROM POUND STERLING

else if (fromchoice == "GBP")

{

switch (tochoice)

{

case "USD":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 1.28).ToString() + "$";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "EUR":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 1.10).ToString() + " €";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

case "GBP":

MessageBox.Show("Can not convert to the same currency", "Error");

textboxfrom.Text = "";

break;

case "CAD":

try

{

textboxto.Text = (double.Parse(textboxfrom.Text) \* 1.72).ToString() + " £";

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

textboxfrom.Text = "";

}

break;

default:

break;

}

}

else if (fromchoice == "DD")

{

}

textOut.Write(textboxfrom.Text + "|");

textOut.Write(fromchoice +" = "+ "|");

textOut.Write(textboxto.Text + "|");

textOut.Write(tochoice + "|");

textOut.WriteLine(" " + currentDateTime);

textOut.Close();

}

private void from\_click(object sender, EventArgs e)

{

if(btnFromCAD.Checked)

{

fromchoice = btnFromCAD.Text;

}

else if(btnFromEUR.Checked)

{

fromchoice = btnFromEUR.Text;

}

else if (btnFromGBP.Checked)

{

fromchoice = btnFromGBP.Text;

}

else

{

fromchoice = btnFromUSD.Text;

}

}

private void to\_check(object sender, EventArgs e)

{

if (btnToCAD.Checked)

{

tochoice = btnToCAD.Text;

}

else if (btnToEUR.Checked)

{

tochoice = btnToEUR.Text;

}

else if (btnToGBP.Checked)

{

tochoice = btnToGBP.Text;

}

else

{

tochoice = btnToUSD.Text;

}

}

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

{

string textToPrint = "";

fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

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

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

{

string row = textIn.ReadLine();

string[] columns = row.Split('|');

textToPrint += columns[0] + columns[1] + columns[2] + columns[3]+ columns[4]+ "\n";

}

MessageBox.Show(textToPrint,"Exchanges made");

// close the input stream for the text file

textIn.Close();

}

private void button1\_Click\_1(object sender, EventArgs e)

{

string btn;

btn = MessageBox.Show("Do you want to Exit?", "lotto649", MessageBoxButtons.OKCancel).ToString();

if (btn == "OK")

{

this.Close();

}

}

}

}

* IP Validator:
* 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;
* using System.Text.RegularExpressions;
* using System.IO;
* namespace WindowsFormsProject420\_CT2
* {
* public partial class IP\_Validator : Form
* {
* DateTime currentDate = DateTime.Now;
* public IP\_Validator()
* {
* InitializeComponent();
* }
* private void button1\_Click(object sender, EventArgs e)
* {
* Regex myObj = new Regex(@"^([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})$");
* if (myObj.IsMatch(textBox1.Text.Trim()) == true)
* {
* MessageBox.Show(textBox1.Text.Trim()+"\n This IP Adress is corrct");
* }
* else
* {
* MessageBox.Show(textBox1.Text.Trim() + "\nthis IP adress is wrong\nthe IP should be 4 byts\nInteger between 0 to 255\nseperated by a dot\n(255.255.255.255)");
* }
* }
* private void button2\_Click(object sender, EventArgs e)
* {
* textBox1.Clear();
* }
* private void IP\_Validator\_Load(object sender, EventArgs e)
* {
* date.Text = ("Today : "+String.Format("{0:M/d/yy}", currentDate));
* }
* }
* }

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

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. ClassCalcul() | Contains three private field with three objects and several functions that will do the calculations for the user. |
| public string Add() | It will take two values and add them together and return the answer in a string variable |
| 1. public string Sub() | It will take two values and subtruct them together and return the answer in a string variable |
| 1. public string Div() | It will take two values and divide them together and return the answer in a string variable |
| public string Mul() | It will take two values and multiply them together and return the answer in a string variable |
| public string Display() | It will first open the text file to write and then save each input from the user in a string variable and finally write the equation into the file |
| public string History() | It will open the text file for reading and the display the content which in this case is the equations saved |
|  |  |
|  |  |

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

The hardest part was in the temperature app. I found a difficulty in changing the value of the variables of type int from (0) to an empty space” “ whenever the user enters a wrong entry the app kept on assigning the variable to 0 and a freezing weather was displayed in the message box.

The easiest part was the IP address validator not much work had to be done there.