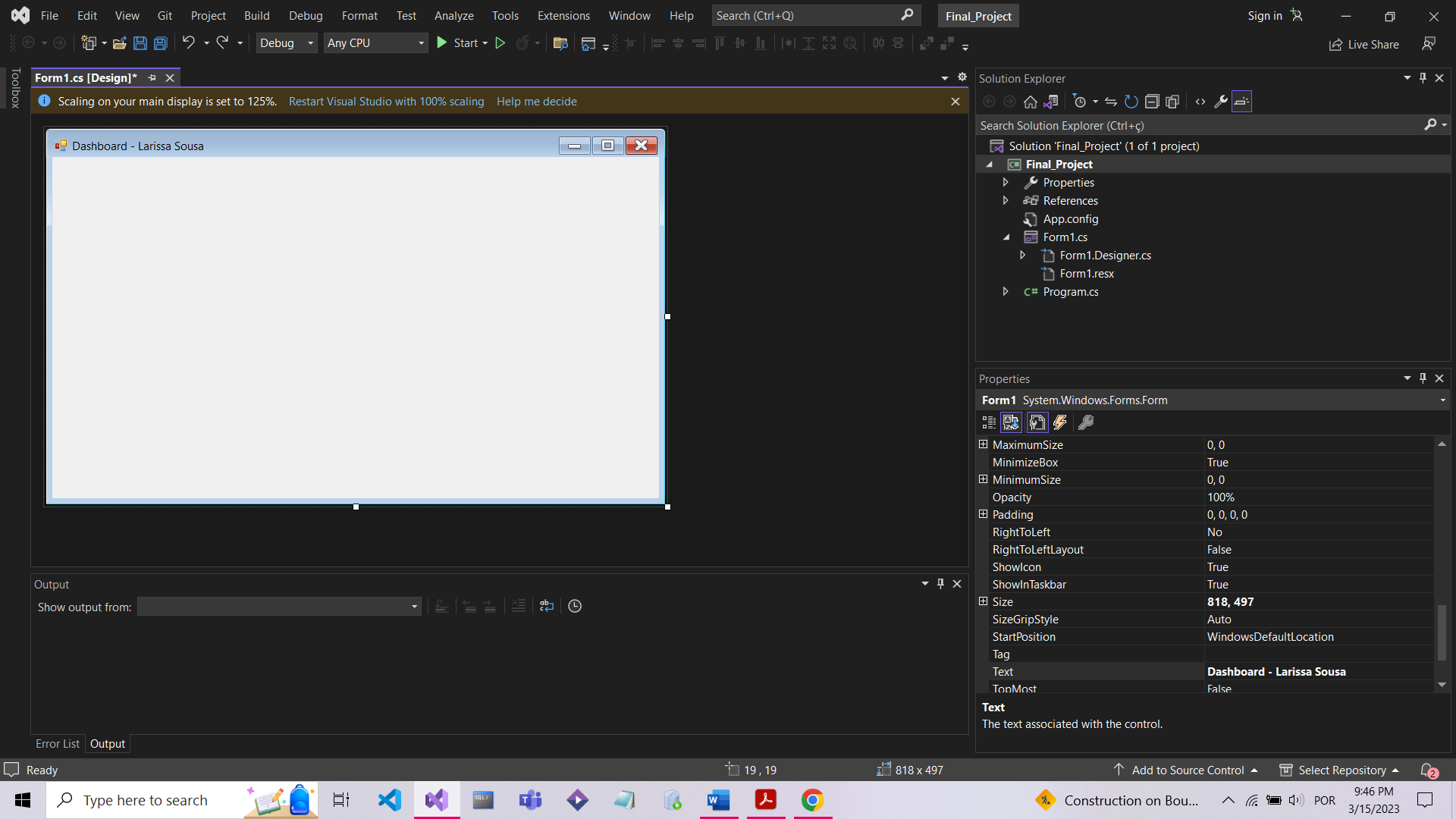
|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

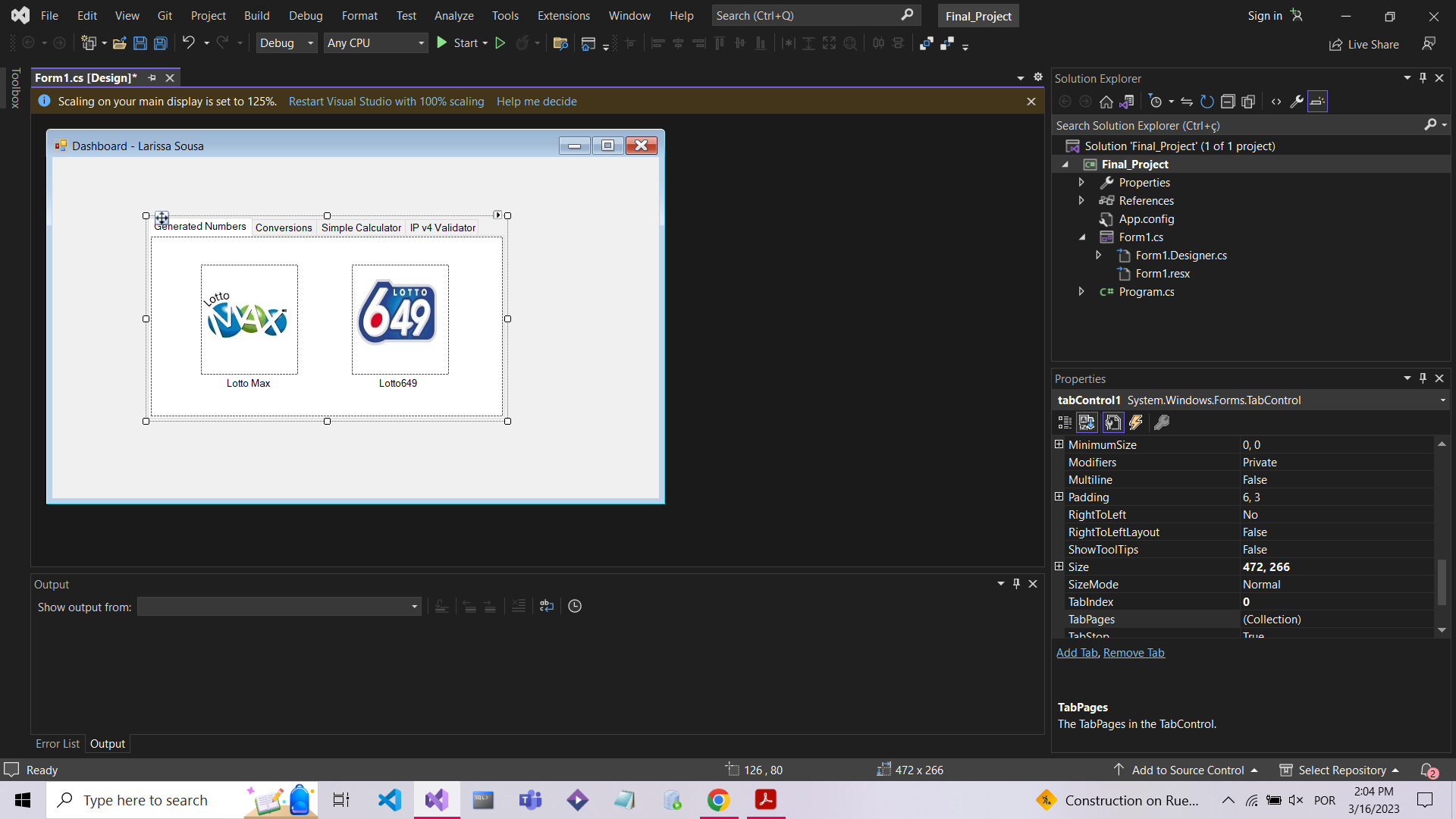
|  |
| --- |
| Your name: Larissa Maria Araujo Sousa  4/15/2023  Version: 8?, FINAL |

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

This application is to create a Windows Form application with several functions that are linked with each other to the user access better such as a money exchange, temperature conversion, if an IP is valid or not, 2 lottery numbers generators and a simple calculator tool.

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





**Actual – 03-21-2023**

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

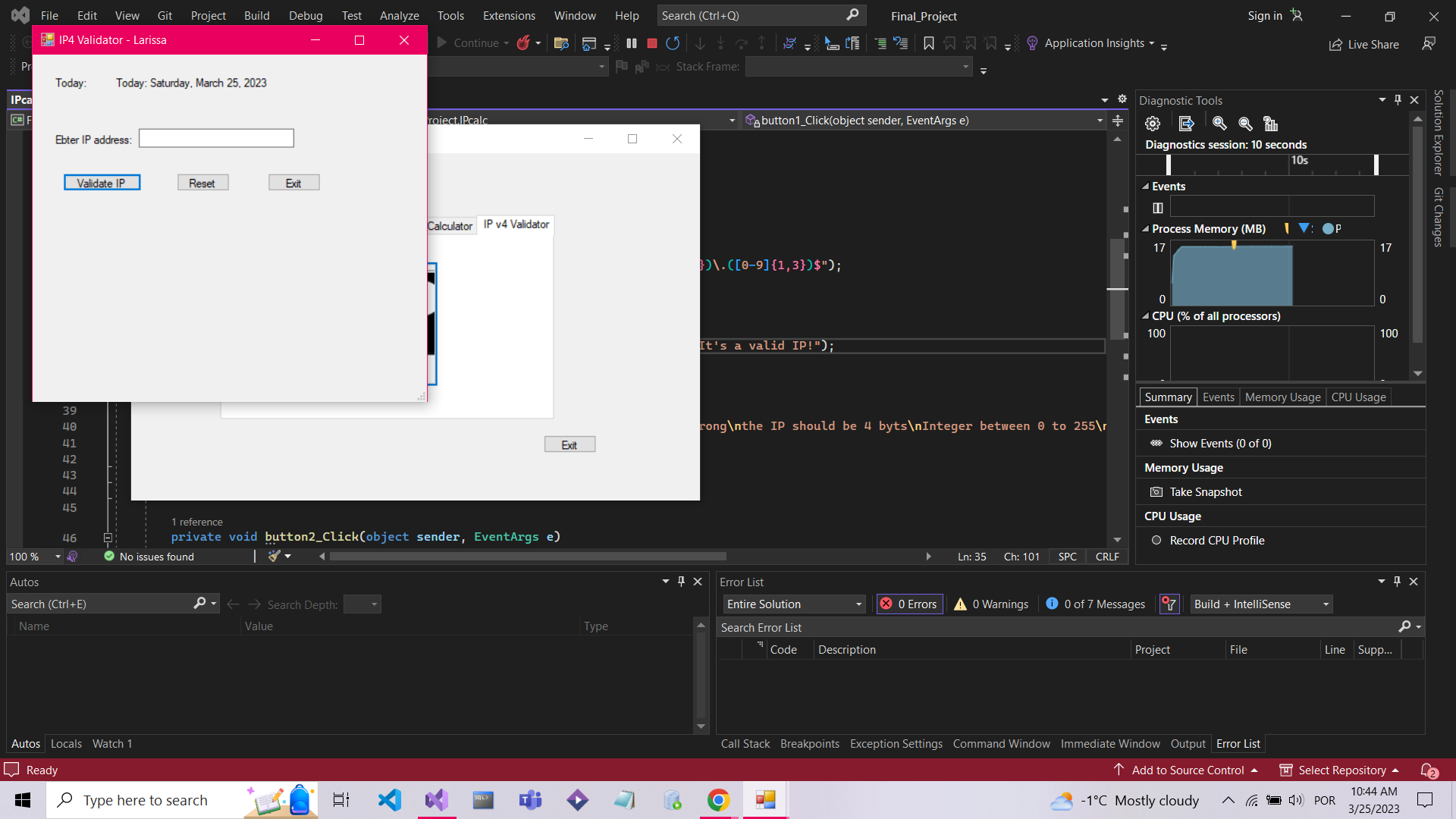
Description automatically generated

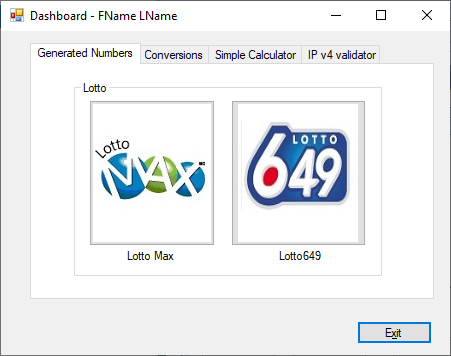
**03-24-2023**

A screenshot of a computer

Description automatically generated with medium confidence

**03-25-2023**





1. If you click on tab…
2. If you click on tab…
3. …
4. If you click on the Exit button, …

**Explanation step by step:**

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated Graphical user interface, application

Description automatically generated

DASHBOARD

* The dashboard is divided in 4 areas allowing the user to transit between them by clicking on the tab or the image;
* After selecting one of the tools, the system creates an instance and open up another tab with the selected functionality;

MONEY EXCHANGE

* After selecting the Money Exchange button, a new instance of Money Exchange will open.
* If the user clicks on Read before writing anything before, a message warning about it will pop up with the past conversions;
* The user should select one money currency “From” and “To” from the radio buttons;
* By default, one country of the “To” is selected when the form load;
* If the user tries to convert anything before selecting a country for “To”, then a message warning about the issue is pop up;
* If the user clicks “Convert” and all the data is valid then the program will perform the conversion by comparing the strings of the checked radio buttons, a dictionary and register it in a file with the current date and time;

Graphical user interface

Description automatically generated Text

Description automatically generated

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

CALCULATOR

* After selecting the calculator image button, a new instance of the calculator will open;
* The user has an option to select the decimal separator as a point to act like the comma and with this it will solve any problems with operations done in different formats;
* To use the calculator the user should insert a number either by typing the value or pressing the buttons, choose an operation, select a second value and then press equal to see the result;
* For each number inserted, the calculator will clean the display and if the user wants to clear all values and operations stored inside the program, the user should press “Clear” or the “CE” button;

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a cell phone

Description automatically generated with medium confidence

IP VALIDATOR

* After selecting the IP image button, a new instance of the IP Validator will open;
* The program will get the current date and write inside the label when the forms load;
* The system will break the string inserted in 4 parts and verifies if the content is in the correct form and then it will display the result when clicked on the “Validate IP” button;
* If the user tries to convert an invalid data, the program will display a message notifying about the issue and showing the correct form to put the data;
* If the user wants to clean the textbox the user press the “Reset” button.
* If the user clicks on the Exit button, a new message box will pop up asking if the instance should be closed. Only the current instance is closed if the user elects “yes”, otherwise nothing happens.

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface, text, chat or text message

Description automatically generated Graphical user interface

Description automatically generated

LOTTO MAX & 649

After selecting the Lotto that you want on the Dashboard button the user has the 3 options:

* If the user clicks at “Generate” button, the program will generate 8 numbers between them and will record it in a file with the name of the lottery, date and time;
* If the user clicks at “Read File” button, the program will read the file created with everything that it was sorted;
* If the user clicks on the Exit button, a new message box will pop up asking if the instance should be closed. Only the current instance is closed if the user elects “yes”, otherwise nothing happens.

Graphical user interface, application, Word

Description automatically generated Graphical user interface, application, Word

Description automatically generated Text

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated Graphical user interface, application

Description automatically generated

Text

Description automatically generatedGraphical user interface, application

Description automatically generated

TEMPERATURE APP

* After selecting the Temperature Convert button, a new instance of Temp App will open;
* By default, the first radio button is checked. If the user changes it, then the labels under the text boxes will also change their places;
* If the user tries to convert an invalid data, the program will display a message notifying about the issue;
* If the user tries to convert a valid data, then the program will convert the number and check if the result matches with one of the specific conditions. If it matches, then the program will display the message at the read only text box under the label “Message”, otherwise it will not display anything. In addition, the operation is registered in a specific file with the current date and time.
* By clicking “Read File”, the program will display the information inside the file mentioned before.
* If the user clicks on the Exit button, a new message box will pop up asking if the instance should be closed. Only the current instance is closed if the user elects “yes”, otherwise nothing happens.

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Text

Description automatically generated Graphical user interface, application

Description automatically generated

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

**DASHBOARD**

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 Final\_Project

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label3\_Click(object sender, EventArgs e)

{

}

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

{// it'll be activated where a new window will open when you click the button

//for the Calculator

Calculator obj = new Calculator();

obj.ShowDialog();

}

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

{// it'll be activated where a new window will open when you click the button

//for to close the dashboard and to exit the applications

this.Close();

}

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

{// it'll be activated where a new window will open when you click the button

//for the IP Calculator

IPcalc obj = new IPcalc();

obj.ShowDialog();

}

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

{// it'll be activated where a new window will open when you click the button

//for the Money Exchange

Cambio obj = new Cambio();

obj.ShowDialog();

}

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

{// it'll be activated where a new window will open when you click the button

//for the Temperature Calculator

Temp obj = new Temp();

obj.ShowDialog();

}

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

{// it'll be activated where a new window will open when you click the button

//for the Lotto Max

LMax obj = new LMax();

obj.ShowDialog();

}

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

{// it'll be activated where a new window will open when you click the button

//for the Lotto 649

L649 obj = new L649();

obj.ShowDialog();

}

}

}

1. Calculator:
2. using System;
3. using System.Collections.Generic;
4. using System.ComponentModel;
5. using System.Data;
6. using System.Drawing;
7. using System.Linq;
8. using System.Text;
9. using System.Threading.Tasks;
10. using System.Windows.Forms;
11. using static System.Windows.Forms.VisualStyles.VisualStyleElement;
12. namespace Final\_Project
13. {
14. public partial class Calculator : Form
15. {
16. string input = string.Empty; //storing user input
17. String operand1 = string.Empty;
18. String operand2 = string.Empty;
19. char calc; // stores the value of the op
20. double result = 0.0; //get result
21. public Calculator()
22. {
23. InitializeComponent();
24. }
25. //if the number 1 is clicked then it'll generate the number 1
26. // and so on; if the number 0 is on the text box it will substitute
27. //by 1, if not it'll concatenate the number 1
28. private void button1\_Click(object sender, EventArgs e)
29. {
30. this.textBoxOutput.Text = "";
31. input += "1";
32. this.textBoxOutput.Text += input;
33. }
34. private void button2\_Click(object sender, EventArgs e)
35. {
36. this.textBoxOutput.Text = "";
37. input += "2";
38. this.textBoxOutput.Text += input;
39. }
40. private void button3\_Click(object sender, EventArgs e)
41. {
42. this.textBoxOutput.Text = "";
43. input += "3";
44. this.textBoxOutput.Text += input;
45. }
46. private void button4\_Click(object sender, EventArgs e)
47. {
48. this.textBoxOutput.Text = "";
49. input += "4";
50. this.textBoxOutput.Text += input;
51. }
52. private void button5\_Click(object sender, EventArgs e)
53. {
54. this.textBoxOutput.Text = "";
55. input += "5";
56. this.textBoxOutput.Text += input;
57. }
58. private void button6\_Click(object sender, EventArgs e)
59. {
60. this.textBoxOutput.Text = "";
61. input += "6";
62. this.textBoxOutput.Text += input;
63. }
64. private void button7\_Click(object sender, EventArgs e)
65. {
66. this.textBoxOutput.Text = "";
67. input += "7";
68. this.textBoxOutput.Text += input;
69. }
70. private void button8\_Click(object sender, EventArgs e)
71. {
72. this.textBoxOutput.Text = "";
73. input += "8";
74. this.textBoxOutput.Text += input;
75. }
76. private void button9\_Click(object sender, EventArgs e)
77. {
78. this.textBoxOutput.Text = "";
79. input += "9";
80. this.textBoxOutput.Text += input;
81. }
82. private void button10\_Click(object sender, EventArgs e)
83. {
84. this.textBoxOutput.Text = "";
85. input += "0";
86. this.textBoxOutput.Text += input;
87. }
88. private void button11\_Click(object sender, EventArgs e)
89. {
90. input += ".";
91. }
92. private void button12\_Click(object sender, EventArgs e)
93. {
94. operand1 = input;
95. calc = '\*';
96. input = string.Empty;
97. }
98. private void button13\_Click(object sender, EventArgs e)
99. {
100. operand1 = input;
101. calc = '/';
102. input = string.Empty;
103. }
104. private void button14\_Click(object sender, EventArgs e)
105. {
106. operand1 = input;
107. calc = '-';
108. input = string.Empty;
109. }
110. private void button15\_Click(object sender, EventArgs e)
111. {
112. operand1 = input;
113. calc = '+';
114. input = string.Empty;
115. }
116. //code for the calcs fot the choosen operation
117. private void button16\_Click(object sender, EventArgs e)
118. {
119. operand2 = input;
120. double num1, num2;
121. double.TryParse(operand1, out num1);
122. double.TryParse(operand2, out num2);
123. string equation = "";
124. if (calc == '+')
125. {
126. result = num1 + num2;
127. textBoxOutput.Text = result.ToString();
128. equation = num1 + " + " + num2 + " = " + result;
129. textBox1.Text = equation;
130. }
131. else if (calc == '-')
132. {
133. result = num1 - num2;
134. textBoxOutput.Text = result.ToString();
135. equation = num1 + " - " + num2 + " = " + result;
136. textBox1.Text = equation;
137. }
138. else if (calc == '\*')
139. {
140. result = num1 \* num2;
141. textBoxOutput.Text = result.ToString();
142. equation = num1 + " \* " + num2 + " = " + result;
143. textBox1.Text = equation;
144. }
145. else if (calc == '/')
146. {
147. if (num2 != 0)
148. {
149. result = num1 / num2;
150. textBoxOutput.Text = result.ToString();
151. equation = num1 + " / " + num2 + " = " + result;
152. textBox1.Text = equation;
153. }
154. else
155. {
156. textBoxOutput.Text = "ERROR DIV BY ZERO";
157. }
158. }
159. }
160. private void textBoxOutput\_TextChanged(object sender, EventArgs e)
161. {
162. }
163. //clear the space
164. private void button17\_Click(object sender, EventArgs e)
165. {
166. this.textBoxOutput.Text = "";
167. this.input = string.Empty;
168. this.operand1 = string.Empty;
169. this.operand2 = string.Empty;
170. }
171. }
172. }
173. IP calculator
174. using System;
175. using System.Collections.Generic;
176. using System.ComponentModel;
177. using System.Data;
178. using System.Drawing;
179. using System.Linq;
180. using System.Text;
181. using System.Text.RegularExpressions;
182. using System.Threading.Tasks;
183. using System.Windows.Forms;
184. namespace Final\_Project
185. {
186. public partial class IPcalc : Form
187. {
188. //declaring the obj to store the time when
189. //the app opens
190. DateTime currentDate = DateTime.Now;
191. public IPcalc()
192. {
193. InitializeComponent();
194. }
195. private void label2\_Click(object sender, EventArgs e)
196. {
197. }
198. private void button1\_Click(object sender, EventArgs e)
199. {
200. {
201. //the regex obj is declared here
202. Regex myObj = new Regex(@"^([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})$");
203. if (myObj.IsMatch(textBox1.Text.Trim()) == true)
204. {//the number will be trimmed and if put correctly will be displayed
205. MessageBox.Show(textBox1.Text.Trim() + "\nThe IP is correct. It's a valid IP!");
206. }
207. else
208. {//if it's wrong it'll show a message if the correct way
209. 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)");
210. textBox1.Focus();
211. }
212. }
213. }
214. private void button2\_Click(object sender, EventArgs e)
215. {
216. textBox1.Text = "";
217. }
218. private void button3\_Click(object sender, EventArgs e)
219. {
220. this.Close();
221. }
222. private void label1\_Click(object sender, EventArgs e)
223. {
224. }
225. private void IPcalc\_Load(object sender, EventArgs e)
226. {//time that the form is loaded
227. labelToday.Text = "Today: " + DateTime.Now.ToLongDateString();
228. }
229. }
230. }
231. Temp Converter
232. using System;
233. using System.Collections.Generic;
234. using System.ComponentModel;
235. using System.Data;
236. using System.Drawing;
237. using System.IO;
238. using System.Linq;
239. using System.Text;
240. using System.Threading.Tasks;
241. using System.Windows.Forms;
242. namespace Final\_Project
243. {
244. public partial class Temp : Form
245. {
246. public Temp()
247. {
248. InitializeComponent();
249. }
250. TempCalc Cal1 = new TempCalc();
251. //true = C to F
252. //false = F to C
253. private void btnConvert\_Click(object sender, EventArgs e)
254. {
255. try
256. {//calculating the temperature
257. Cal1.Initval = Convert.ToDouble(txtInput.Text);
258. if (Cal1.Convertstatus)
259. Cal1.Finalval = ((Cal1.Initval \* 9 / 5) + 32);
260. else
261. Cal1.Finalval = (Cal1.Initval - 32) \* 5 / 9;
262. txtResult.Text = Cal1.Finalval.ToString();
263. txtboxMessage.Text = Cal1.Message();
264. //starting exportation
265. //declaring the structures to be used into classes
266. //declaring the showing dates structure
267. //declaring the path to write the file
268. DateTime dateodnow = DateTime.Now;
269. FileStream directory = new FileStream(@"../../txtfiles/TempConversions.txt", FileMode.Append, FileAccess.Write);
270. StreamWriter file = new StreamWriter(directory);
271. if (Cal1.Convertstatus)
272. file.WriteLine(txtInput.Text + " C = " + txtResult.Text + " " + " F , " + dateodnow.ToShortDateString() + " " + dateodnow.ToLongTimeString());
273. else
274. file.WriteLine(txtInput.Text + " F = " + txtResult.Text + " " + " C , " + dateodnow.ToShortDateString() + " " + dateodnow.ToLongTimeString());
275. file.Close();
276. directory.Close();
277. }
278. catch (Exception ex)
279. {
280. MessageBox.Show("Error :\n" + ex.Message + "\nPlease put only numbers.", "Error");
281. txtInput.Focus();
282. }
283. }
284. private void btnExit\_Click(object sender, EventArgs e)
285. {
286. if (MessageBox.Show("Do you want to quit the application TempApp?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")
287. {
288. this.Close();
289. }
290. }
291. private void radiobtnCtoF\_CheckedChanged(object sender, EventArgs e)
292. {
293. labelResult.Text = "F";
294. labelInput.Text = "C";
295. txtInput.Text = txtResult.Text = "";
296. Cal1.Convertstatus = true;
297. }
298. private void radiobtnFtoC\_CheckedChanged(object sender, EventArgs e)
299. {
300. labelResult.Text = "C";
301. labelInput.Text = "F";
302. txtInput.Text = txtResult.Text = "";
303. Cal1.Convertstatus = false;
304. }
305. private void btnReadFile\_Click(object sender, EventArgs e)
306. {
307. try
308. {//the object which read the file and an instant put in
309. //the indicated file
310. if (!File.Exists(@"../../txtfiles/TempConversions.txt")) throw new Exception("File not created yet. Please do one convertion first.");
311. FileStream directory = new FileStream(@"../../txtfiles/TempConversions.txt", FileMode.Open, FileAccess.Read);
312. StreamReader file = new StreamReader(directory);
313. string textoprint = "FROM\tTO\t\tDATE\n" + file.ReadToEnd();
314. file.Close();
315. directory.Close(); //the list will be printed on a new window
316. MessageBox.Show(textoprint, "TempConversions.txt");
317. }
318. catch (Exception ex)
319. { MessageBox.Show(ex.Message, "Error"); }
320. }
321. private void Temp\_Load(object sender, EventArgs e)
322. {
323. }
324. public class TempCalc
325. {
326. private bool convertstatus;
327. private double initval, finalval;
328. public bool Convertstatus { get; set; }
329. public double Initval { get; set; }
330. public double Finalval { get; set; }
331. //messages showing depending on the value input
332. public string Message()
333. {
334. if ((Convertstatus && Initval == 100) || (!Convertstatus && Initval == 212))
335. {
336. return "Water boils";
337. }
338. else if ((Convertstatus && Initval == 40) || (!Convertstatus && Initval == 104))
339. {
340. return "Hot Bath";
341. }
342. else if ((Convertstatus && Initval == 37) || (!Convertstatus && Initval == 98.6))
343. {
344. return "Body temperature";
345. }
346. else if ((Convertstatus && Initval == 30) || (!Convertstatus && Initval == 86))
347. {
348. return "Beach weather";
349. }
350. else if ((Convertstatus && Initval == 21) || (!Convertstatus && Initval == 70))
351. {
352. return "Room temperature";
353. }
354. else if ((Convertstatus && Initval == 10) || (!Convertstatus && Initval == 50))
355. {
356. return "Cool Day";
357. }
358. else if ((Convertstatus && Initval == 0) || (!Convertstatus && Initval == 32))
359. {
360. return "Freezing point of water";
361. }
362. else if ((Convertstatus && Initval == -18) || (!Convertstatus && Initval == 0))
363. {
364. return "Very Cold Day";
365. }
366. else if (Initval == -40)
367. {
368. return "Extremely Cold Day\n(and the same number!)";
369. }
370. else { return ""; }
371. }
372. }
373. }
374. }
375. Money Exchange
376. using System;
377. using System.Collections.Generic;
378. using System.ComponentModel;
379. using System.Data;
380. using System.Drawing;
381. using System.IO;
382. using System.Linq;
383. using System.Text;
384. using System.Threading.Tasks;
385. using System.Windows.Forms;
386. namespace Final\_Project
387. {
388. public partial class Cambio : Form
389. /\*credits for the imgs from the flags: https://www.flaticon.com/br/packs/countrys-flags \*/
390. {
391. public Cambio()
392. {
393. InitializeComponent();
394. }
395. private void Cambio\_Load(object sender, EventArgs e)
396. {
397. }
398. private void groupBox1\_Enter(object sender, EventArgs e)
399. {
400. }
401. private void btnExit\_Click(object sender, EventArgs e)
402. {//a pop up to exit the application when you click the button to get out
403. if (MessageBox.Show("Do you want to quit the application Money Exchange?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")
404. {
405. this.Close();
406. }
407. }
408. public string ConvertValues(string FROMopt, string TOopt)
409. {
410. //verify if it's the same what it was chosen
411. if (FROMopt == TOopt)
412. { return (Convert.ToDouble(textFROM.Text)).ToString(); } //Just to remove if there is 0 before
413. else
414. {
415. //creation of a dictionary to check and compare if it's an error
416. Dictionary<string, double> ExchangeFactor = new Dictionary<string, double>();
417. ExchangeFactor.Add("CADUSD", 0.797171);
418. ExchangeFactor.Add("USDCAD", 1.25444);
419. ExchangeFactor.Add("CADEUR", 1.48608);
420. ExchangeFactor.Add("EURCAD", 0.672911);
421. ExchangeFactor.Add("CADGBP", 1.72956);
422. ExchangeFactor.Add("GBPCAD", 0.578180);
423. ExchangeFactor.Add("CADWON", 0.93877);
424. ExchangeFactor.Add("WONCAD", 1.07);
425. ExchangeFactor.Add("USDEUR", 0.844274);
426. ExchangeFactor.Add("EURUSD", 1.18445);
427. ExchangeFactor.Add("USDGBP", 0.725285);
428. ExchangeFactor.Add("GBPUSD", 1.37874);
429. ExchangeFactor.Add("WONUSD", 0.77);
430. ExchangeFactor.Add("USDWON", 1.29583);
431. ExchangeFactor.Add("EURGBP", 0.859034);
432. ExchangeFactor.Add("GBPEUR", 1.16410);
433. ExchangeFactor.Add("EURWON", 0.72);
434. ExchangeFactor.Add("WONEUR", 0.160561);
435. ExchangeFactor.Add("GBPWON", 1.58414);
436. ExchangeFactor.Add("WONGBP", 0.63);
437. return (Convert.ToDouble(textFROM.Text) \* ExchangeFactor[FROMopt + TOopt]).ToString("0.00");
438. }
439. }
440. private void btnReadFile\_Click(object sender, EventArgs e)
441. {
442. try
443. {//declaring obj to manipulate the file
444. if (!File.Exists(@"../../txtfiles/MoneyConversions.txt")) throw new Exception("File not created yet. Please do a convertion first.");
445. FileStream directory = new FileStream(@"../../txtfiles/MoneyConversions.txt", FileMode.Open, FileAccess.Read);
446. StreamReader file = new StreamReader(directory);
447. string textoprint = "FROM\tTO\t\tDATE\n" + file.ReadToEnd();
448. file.Close();
449. directory.Close();
450. MessageBox.Show(textoprint, "MoneyConversions.txt");
451. }
452. catch (Exception ex)
453. { MessageBox.Show(ex.Message, "Error"); }
454. }
455. private void btnConvert\_Click(object sender, EventArgs e)
456. {
457. string FROMoption = "", TOoption = "";
458. try
459. {
460. //verifying the conversion and if's not right the exception will act
461. //checks the option "FROM"
462. foreach (RadioButton rbtnFrom in groupBox1.Controls.OfType<RadioButton>())
463. { if (rbtnFrom.Checked) { FROMoption = rbtnFrom.Text; break; } }
464. //same as above: checks the option "TO"
465. foreach (RadioButton rbtnTo in groupBox2.Controls.OfType<RadioButton>())
466. { if (rbtnTo.Checked) { TOoption = rbtnTo.Text; break; } }
467. if (TOoption == "") { throw new InvalidOperationException("Please select one 'TO' option."); }
468. textTO.Text = this.ConvertValues(FROMoption, TOoption);
470. DateTime dateNOW = DateTime.Now; //store the time the form is loaded
471. FileStream directory = new FileStream(@"../../txtfiles/MoneyConversions.txt", FileMode.Append, FileAccess.Write);
472. StreamWriter file = new StreamWriter(directory);
473. file.WriteLine(textFROM.Text + " " + FROMoption + " = " + textTO.Text + " " + TOoption + ", " + dateNOW.ToShortDateString() + " " + dateNOW.ToLongTimeString()); ;
474. file.Close();
475. directory.Close();
476. }
477. catch (Exception ex)
478. {
479. MessageBox.Show(ex.Message);
480. }
481. }
482. private void textFROM\_TextChanged(object sender, EventArgs e)
483. {
484. }
485. }
486. }
487. Lotto Max
488. using System;
489. using System.Collections.Generic;
490. using System.ComponentModel;
491. using System.Data;
492. using System.Drawing;
493. using System.IO;
494. using System.Linq;
495. using System.Text;
496. using System.Threading.Tasks;
497. using System.Windows.Forms;
498. namespace Final\_Project
499. {
500. public partial class LMax : Form
501. {
502. DateTime currentDateTime = DateTime.Now;
503. FileStream fs = null;
504. string path = @"../../txtfiles/LottoNumbers.txt";
505. public LMax()
506. {
507. InitializeComponent();
508. }
509. private void btnGenerate\_Click(object sender, EventArgs e)
510. {
511. fs = new FileStream(path, FileMode.Append, FileAccess.Write);
512. StreamWriter textOut = new StreamWriter(fs);
513. Random random = new Random(); //constructor of the obj random
514. int value;
515. string[] g = new string[8];
516. MaxBox.Text = "";
517. for (int i = 0; i < 7; i++)
518. {
519. value = random.Next(1, 49); //generate the random number
520. MaxBox.Text += value.ToString() + "\r\n";
521. g[i] = value.ToString();
522. }
523. showNumbers.Text = "";
524. for (int i = 0; i < 7; i++)
525. {
526. value = random.Next(0, 10);
527. showNumbers.Text += value.ToString();
528. }
529. textOut.Write("Max" + "\t" + "|");
530. for (int i = 0; i < 7; i++)
531. {
532. textOut.Write(g[i] + ",");
533. }
534. textOut.Write(" Extra (" + g[7] + ")" + "|");
535. textOut.WriteLine(" " + currentDateTime);
536. textOut.Close();
537. }
538. private void btnreadfile\_Click(object sender, EventArgs e)
539. {
540. // read the data from the file and store it in the list
541. try
542. {
543. if (!File.Exists(@"../../txtfiles/LottoNumbers.txt")) throw new Exception("File not created yet. Please generate some numbers first.");
544. if (!File.Exists(@"../../txtfiles/LottoNumbers.txt")) throw new Exception("File not created yet. Please generate some lotto numbers first.");
545. FileStream directory = new FileStream(@"../../txtfiles/LottoNumbers.txt", FileMode.Open, FileAccess.Read);
546. StreamReader file = new StreamReader(directory);
547. string textoprint = file.ReadToEnd();
548. file.Close();
549. directory.Close();
550. MessageBox.Show(textoprint, "All Winning Numbers");
551. }// close the input stream for the text file
552. catch (Exception ex)
553. { MessageBox.Show(ex.Message, "Error"); }
555. }
556. private void btnExit\_Click(object sender, EventArgs e)
557. {
558. byte btnVal = 0;
559. btnVal = Convert.ToByte(MessageBox.Show("Do you want to exit?", "Lotto Max", MessageBoxButtons.OKCancel));
560. if (btnVal == 1)
561. {
562. this.Close();
563. }
564. }
565. }
566. }
567. Lotto 649
568. using System;
569. using System.Collections.Generic;
570. using System.ComponentModel;
571. using System.Data;
572. using System.Drawing;
573. using System.IO;
574. using System.Linq;
575. using System.Text;
576. using System.Threading.Tasks;
577. using System.Windows.Forms;
578. namespace Final\_Project
579. {
580. public partial class L649 : Form
581. {//declaring the structures, date and paths that will be with the generated
582. //file
583. DateTime currentDateTime = DateTime.Now;
584. FileStream fs = null;
585. string path = @"../../txtfiles/LottoNumbers.txt";
586. public L649()
587. {
588. InitializeComponent();
589. }
590. private void btnGenerate\_Click(object sender, EventArgs e)
591. {//declaring the random number, the variables for the numbers
592. //and the obj to manipulate the files
593. fs = new FileStream(path, FileMode.Append, FileAccess.Write);
594. StreamWriter textOut = new StreamWriter(fs);
595. Random random = new Random();
596. int value;
597. string[] g = new string[8];
598. //number is generated amd stored into the array, checks
599. //if it's a repeated number
600. showbox.Text = "";
601. for (int i = 0; i < 7; i++)
602. {
603. value = random.Next(1, 49);
604. showbox.Text += value.ToString() + "\r\n";
605. g[i] = value.ToString();
606. }
607. //clear the label for each iteration
608. //check if there's a repeated number etc
609. showNumbers.Text = "";
610. for (int i = 0; i < 7; i++)
611. {
612. value = random.Next(0, 10);
613. showNumbers.Text += value.ToString();
614. }
615. textOut.Write("649" + "\t" + "|");
616. for (int i = 0; i < 6; i++)
617. {
618. textOut.Write(g[i] + ",");
619. }
620. textOut.Write(" Extra (" + g[6] + ")" + "|");
621. textOut.WriteLine(" " + currentDateTime);
622. textOut.Close();
623. }
624. private void btnExit\_Click(object sender, EventArgs e)
625. {
626. byte btnVal = 0;
627. btnVal = Convert.ToByte(MessageBox.Show("Do you want to exit?", "Lotto Max", MessageBoxButtons.OKCancel));
628. if (btnVal == 1)
629. {
630. this.Close();
631. }
632. }
633. private void btnreadfile\_Click(object sender, EventArgs e)
634. {
635. // read the data from the file and store it in the list
636. try
637. {
638. if (!File.Exists(@"../../txtfiles/LottoNumbers.txt")) throw new Exception("File not created yet. Please generate some numbers first.");
639. if (!File.Exists(@"../../txtfiles/LottoNumbers.txt")) throw new Exception("File not created yet. Please generate some lotto numbers first.");
640. FileStream directory = new FileStream(@"../../txtfiles/LottoNumbers.txt", FileMode.Open, FileAccess.Read);
641. StreamReader file = new StreamReader(directory);
642. string textoprint = file.ReadToEnd();
643. file.Close();
644. directory.Close();
645. MessageBox.Show(textoprint, "All Winning Numbers");
646. }// close the input stream for the text file
647. catch (Exception ex)
648. { MessageBox.Show(ex.Message, "Error"); }
649. }
650. private void button1\_Click(object sender, EventArgs e)
651. {
652. }
653. }
654. }
655. **Present the classes and/or methods that you create or you did use in the project.**

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Calculator | Class using string, double, Parse for the operations on the Calculator app; |
| 1. Button1\_Click – button16\_Click | Methods as it shows: numbers 0 to 9, the operations (+, -, X, /) and the other symbols (=, ., Clear) |
| 1. IPcal | Class using Regex to associate with the IP and show if is valid or not |
| 1. button1\_Click | A method with the Regex object declared, put correctly or showing the error |
| 1. button2\_Click | Show textBox1 |
| 1. button3\_Click | Close application |
| 1. IPcalc\_Load | Class to show the date and time |
| 1. Dashboard | Control what the user chooses |
| 1. Cambio | User chooses from the money exchange form |
| 1. btnExit\_Click | Show a pop up asking with the user wants to exit the application |
| 1. ConvertValues | Class that verify what was chosen and save/storing the numbers and data on the Dictionary |
| 1. btnReadFile\_Click | Method to read the file, declaring the object to print in the file |
| 1. btnConvert\_Click | The string FROMopt and TOopt to try with the options that the dictionary created and convert the information, cheking FROM and TO |
| 1. TempApp | Used for controlling what the user chooses from the Temp App converter form |
| 1. btnConvert\_Click | Method to calculate the temperature, declare the structures, the path to read the file etc |
| 1. btnExit\_Click | Close the application |
| 1. btnReadFile\_Click | After checking the buttons C to F and F to C the object read the file and will print a list |
| 1. TempCalc | Class used to read the operations of the temperature requested at the conversion and shows the correct message |
| 1. L649 | Class that will declare the structures, date and path of the chosen application |
| 1. LMax | Class that will declare the structures, date and path of the chosen application |
| 1. btnGenerate\_Click | Method to declare the random number, the object to manipulate the file, the variables etc |
| 1. btnExit\_Click | Exit the application |
| 1. btnreadfile\_Click | Read the file and store the data in the list |

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

My calculator didn’t work at first and I was breaking my head trying to find what it was wrong so I had to redo everything

My images on the dashboard were ok, but when I start/debug they become cut like 1/3 so I asked Kathleen for help and she said about the Stretch on “BackgroundImageLayout” so now it’s ok

My timestamp thing on the IP didn’t appear so after thinking I realized that I didn’t write the code on the correct button and put the right information to get the command

In the exchange money application, it was complicated to do the calc because the values are so different and change every time so I put my understanding of the exchange

The design was easy but couldn’t put the color on the text box at the temperature application so I did a simple version