Screen shots of the Maths Quiz Tutorial guided by the page from the Microsoft Support Website ()

The code here was lifted from the website but I have learnt a lot about the C# syntax here, for example:   
I learnt how to add textboxes and labels to the main form. I got more comfortable with the classes and methods.

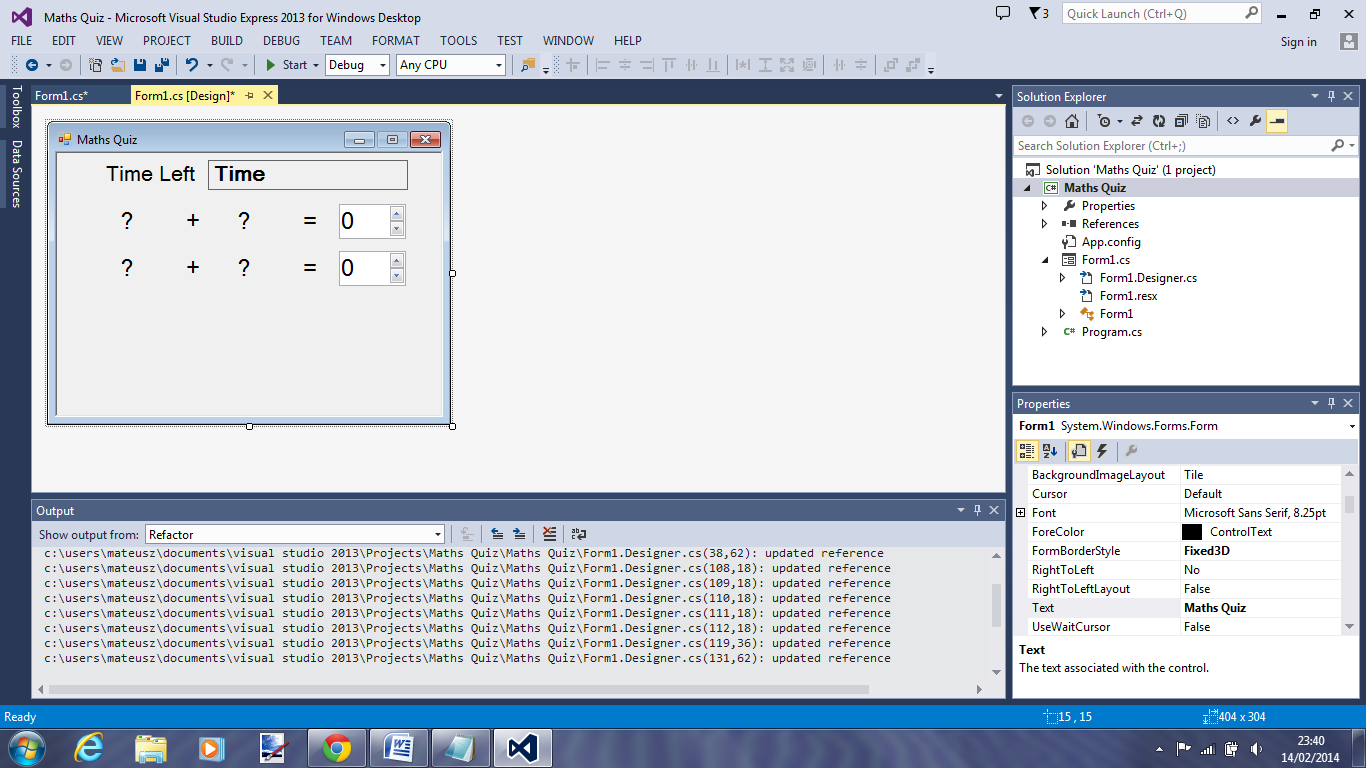
Declaring methods: private void *MethodName* () { ... }

Methods that are declared as private do not make a difference in this program example. Private means something is hidden from the client and cannot access it directly by calling the method. You would be about to call the private methods within the same class but could not be used by a class outside the namespace.

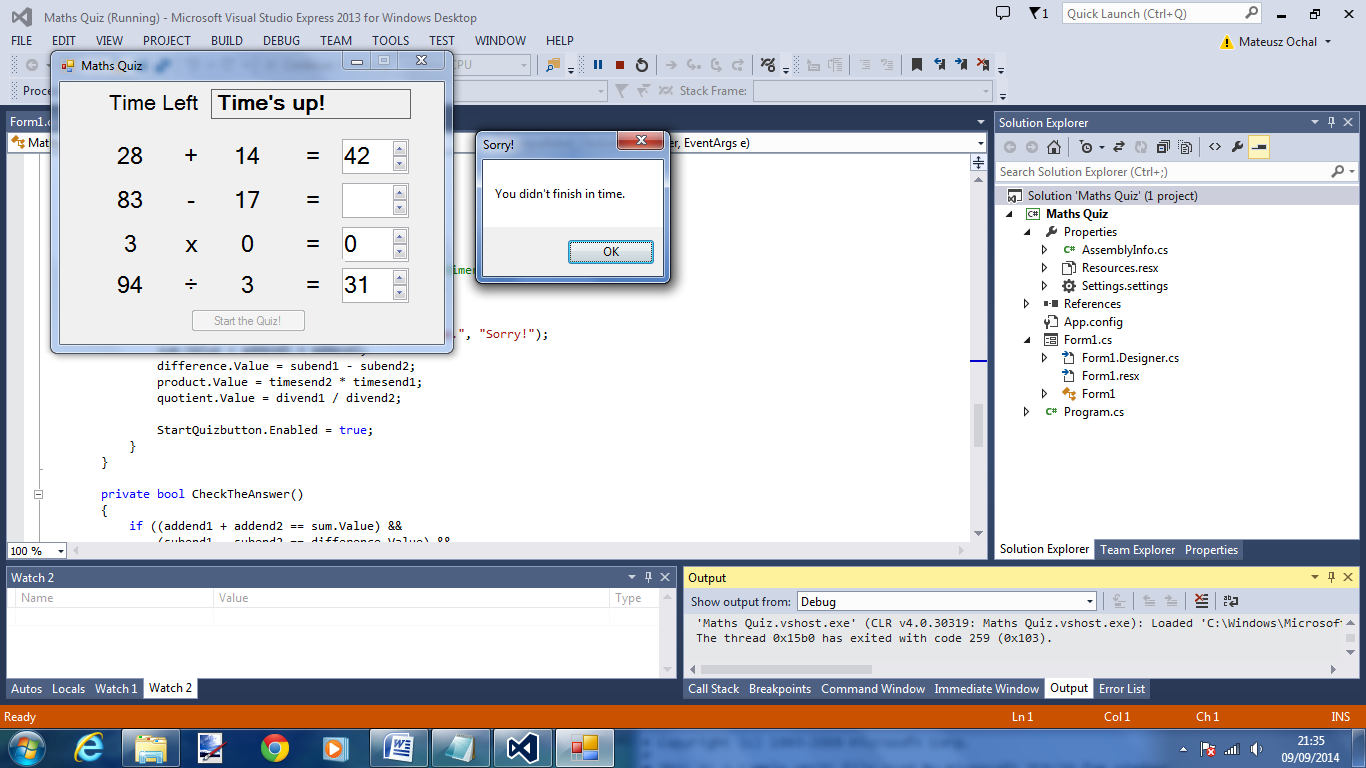
Void means that the method returns nothing. Otherwise known as a Procedure which is what we have leant about in Computing. The syntax is new and you can change the ‘void’ into ‘int’ or other type and the method will become a function.

Textboxes are treated as objects in C# which means they have fields (prosperities) and methods (functions and procedures).The dot after textbox, like textbox.text, lets you access the public fields and methods of an object and allows you to change or request the value stored in the property of that object. You can call methods likewise.

Screenshot of the designer and added textboxes and labels:



Final screenshot of working solution:



Code:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Maths\_Quiz

{

public partial class Form1 : Form

{

// Create a Random object called randomizer

// to generate random numbers.

Random randomizer = new Random();

// These integer variables store the numbers

// for the addition problem.

int addend1;

int addend2;

int subend1;

int subend2;

int timesend1,

timesend2,

divend1,

divend2;

int timeLeft;

public Form1()

{

InitializeComponent();

}

private void StarttheQuiz()

{

addend1 = randomizer.Next(51);

addend2 = randomizer.Next(51);

subend1 = randomizer.Next(100);

subend2 = randomizer.Next(100);

timesend1 = randomizer.Next(17);

timesend2 = randomizer.Next(17);

divend1 = randomizer.Next(211);

divend2 = randomizer.Next(11);

plusLeftLabel.Text = addend1.ToString();

plusRightLabel.Text = addend2.ToString();

MinusLeftlabel.Text = subend1.ToString();

MinusRightlabel.Text = subend2.ToString();

MultiLeftlabel.Text = timesend1.ToString();

MultiRightlabel.Text = timesend2.ToString();

DivLeftLabel.Text = divend1.ToString();

DivRightlabel.Text = divend2.ToString();

sum.Value = 0;

difference.Value = 0;

product.Value = 0;

quotient.Value = 0;

timeLeft = 40;

timerlabel.Text = (timeLeft.ToString() + " seconds");

timer.Start();

}

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

{

}

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

{

}

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

{

StarttheQuiz();

StartQuizbutton.Enabled = false;

}

private void timer\_Tick(object sender, EventArgs e)

{

if (CheckTheAnswer())

{

timer.Stop();

MessageBox.Show("You got all the answers right!",

"Congratulations!");

StartQuizbutton.Enabled = true;

}

else if (timeLeft > 0)

{

// Display the new time left

// by updating the Time Left label.

timeLeft = timeLeft - 1;

timerlabel.Text = timeLeft + " seconds";

}

else

{

// If the user ran out of time, stop the timer, show

// a MessageBox, and fill in the answers.

timer.Stop();

timerlabel.Text = "Time's up!";

MessageBox.Show("You didn't finish in time.", "Sorry!");

sum.Value = addend1 + addend2;

difference.Value = subend1 - subend2;

product.Value = timesend2 \* timesend1;

quotient.Value = divend1 / divend2;

StartQuizbutton.Enabled = true;

}

}

private bool CheckTheAnswer()

{

if ((addend1 + addend2 == sum.Value) &&

(subend1 - subend2 == difference.Value) &&

(timesend1 \* timesend2 == product.Value) &&

(divend1 / divend2 == quotient.Value))

return true;

else

return false;

}

private void answer\_Enter(object sender, EventArgs e)

{

// Select the whole answer in the NumericUpDown control.

NumericUpDown answerBox = sender as NumericUpDown;

if (answerBox != null)

{

int lengthOfAnswer = answerBox.Value.ToString().Length;

answerBox.Select(0, lengthOfAnswer);

}

}

}

}