Faculty of Information Technology
University of Moratuwa

Degree of Bachelor of Information Technology (BIT)External

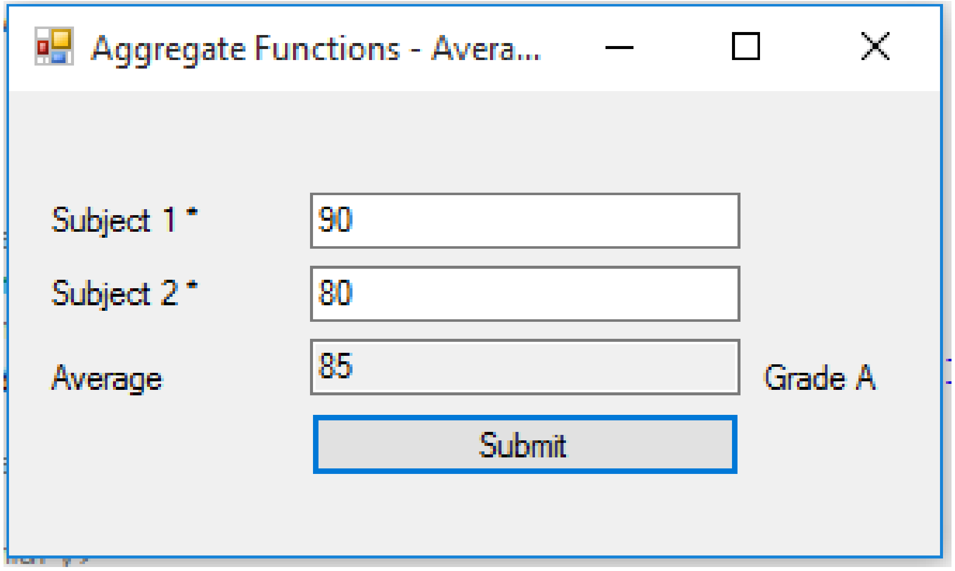
ITE1112: Visual Application Programming – Activity 2

Learning Objectives: Understand fundamentals of validations

Question 1:

Create the simple application to implement an Aggregation Function-Average for the two subject's marks. When you submit the marks of the two-subject average should be displayed in given text box.

Your final output should be as below



Aggregate Functions - Avera...

Subject 1 * 90

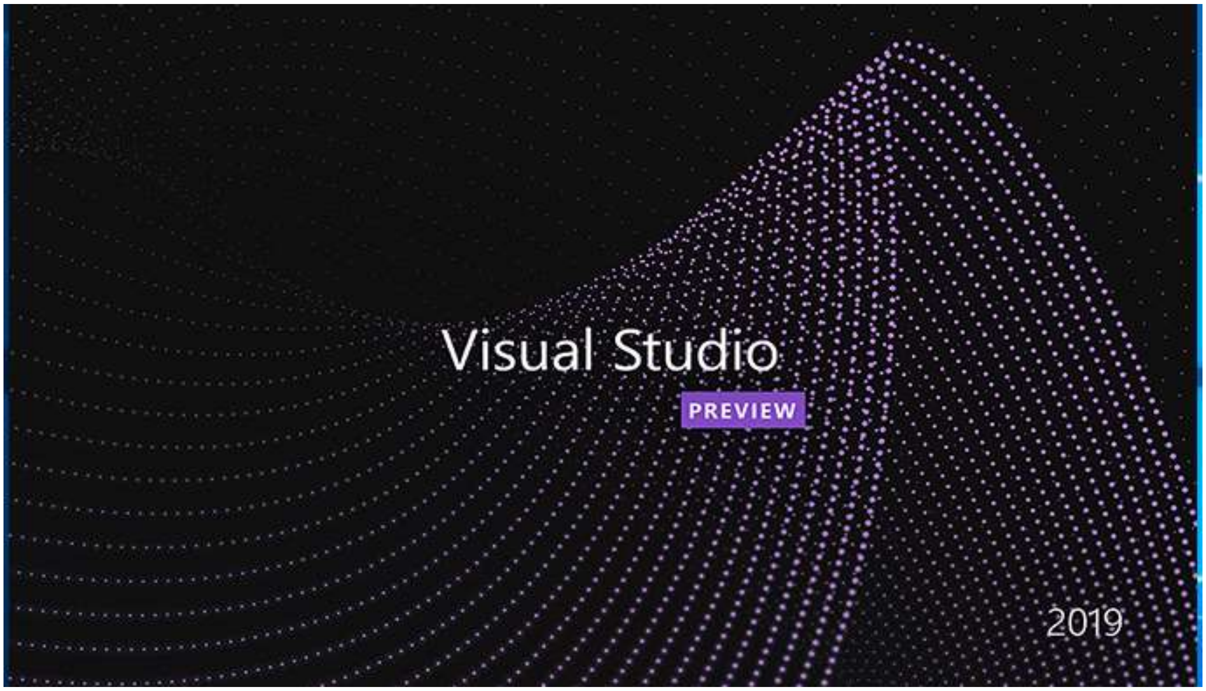
Subject 2 * 80

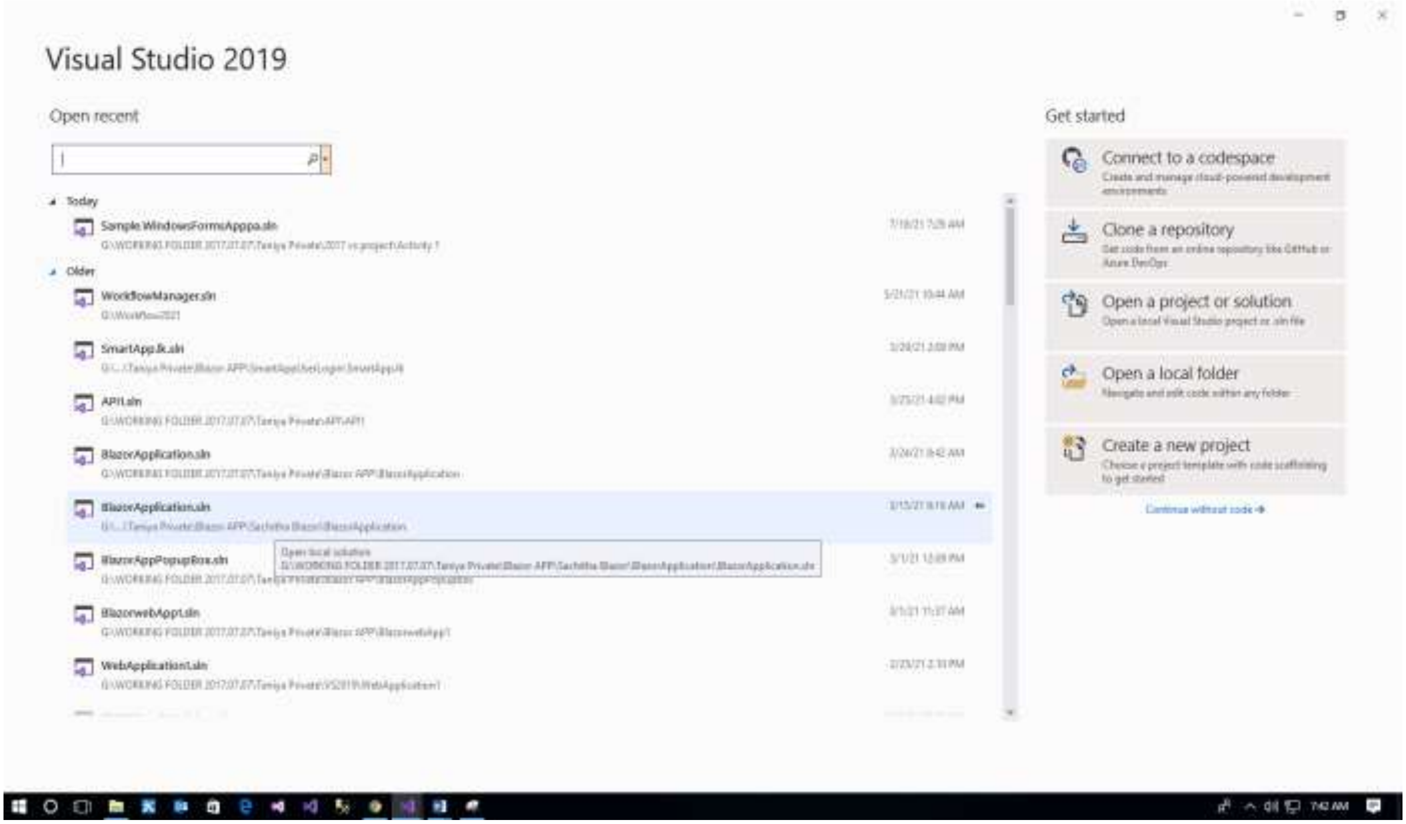
Average 85 Grade A

Answer

Steps for developing the above application:

1. On the start page of the visual studio 2019, you can create new project as below.





Visual Studio 2019

Open recent

Today

SampleWindowsFormsAppqa.sln — G:\WORKING FOLDER 2017.07.07\Taniya Private\2017 vs project\Activity 1 — 7/18/21 7:28 AM

Older

WorkflowManager.sln — G:\Workflow2021 — 5/21/21 10:44 AM

SmartApp.lk.sln — G:\...\Taniya Private\Blazor APP\SmartApp\SetLogin\SmartApplk — 3/29/21 2:08 PM

API.sln — G:\WORKING FOLDER 2017.07.07\Taniya Private\API\API — 3/25/21 4:02 PM

BlazorApplication.sln — G:\WORKING FOLDER 2017.07.07\Taniya Private\Blazor APP\BlazorApplication — 3/24/21 8:42 AM

BlazorApplication.sln — G:\...\Taniya Private\Blazor APP\Sachitha Blazor\BlazorApplication — 3/15/21 8:16 AM

Open local solution
G:\WORKING FOLDER 2017.07.07\Taniya Private\Blazor APP\Sachitha Blazor\BlazorApplication\BlazorApplication.sln

BlazorAppPopupBox.sln — G:\WORKING FOLDER 2017.07.07\Taniya Private\Blazor APP\BlazorAppPopupBox — 3/1/21 12:09 PM

BlazorwebApp1.sln — G:\WORKING FOLDER 2017.07.07\Taniya Private\Blazor APP\BlazorwebApp1 — 3/1/21 11:37 AM

WebApplication1.sln — G:\WORKING FOLDER 2017.07.07\Taniya Private\VS2019\WebApplication1 — 2/23/21 2:33 PM

Get started

Connect to a codespace — Create and manage cloud-powered development environments

Clone a repository — Get code from an online repository like GitHub or Azure DevOps

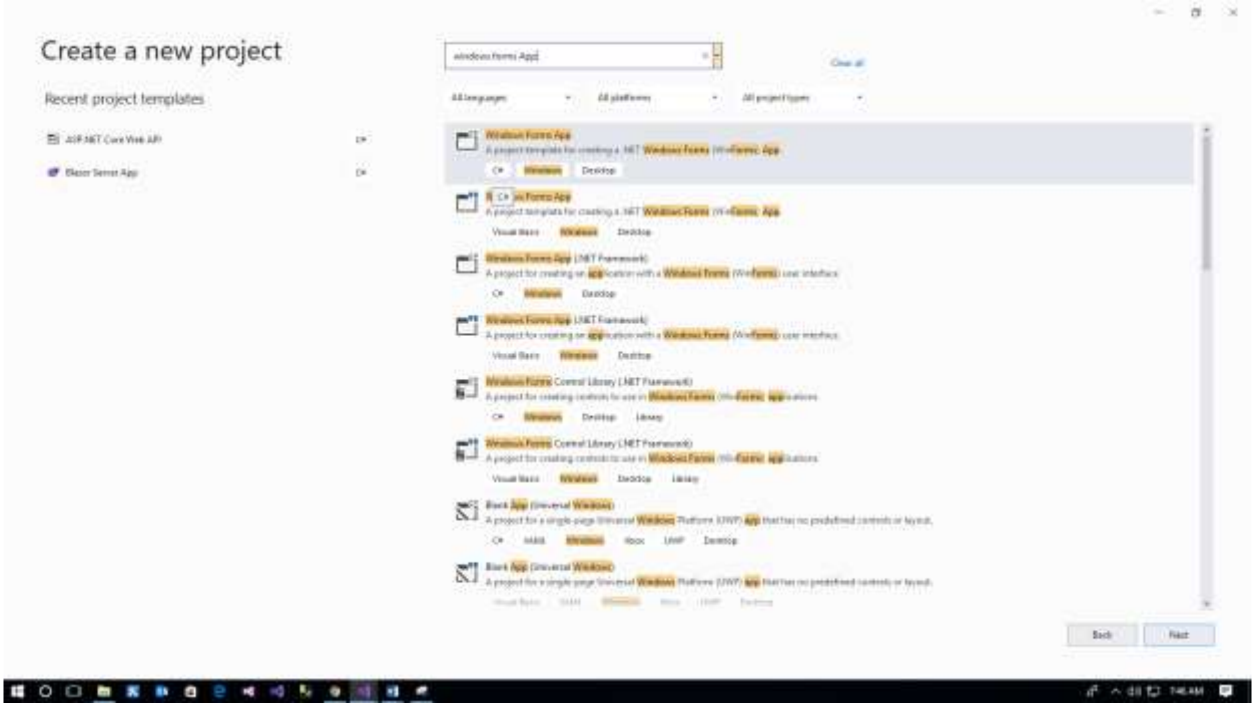
Open a project or solution — Open a local Visual Studio project or .sln file

Open a local folder — Navigate and edit code within any folder

Create a new project — Choose a project template with code scaffolding to get started

Continue without code →

2. In the search box type **Windows form App.** Then, select **Windows form App**



Create a new project

Recent project templates

ASP.NET Core Web API — C#

Blazor Server App — C#

windows forms App

Clear all

All languages ▾ All platforms ▾ All project types ▾

Windows Forms App — A project template for creating a .NET Windows Forms (WinForms) App. — C# Windows Desktop

Windows Forms App — A project template for creating a .NET Windows Forms (WinForms) App. — Visual Basic Windows Desktop

Windows Forms App (.NET Framework) — A project for creating an application with a Windows Forms (WinForms) user interface — C# Windows Desktop

Windows Forms App (.NET Framework) — A project for creating an application with a Windows Forms (WinForms) user interface — Visual Basic Windows Desktop

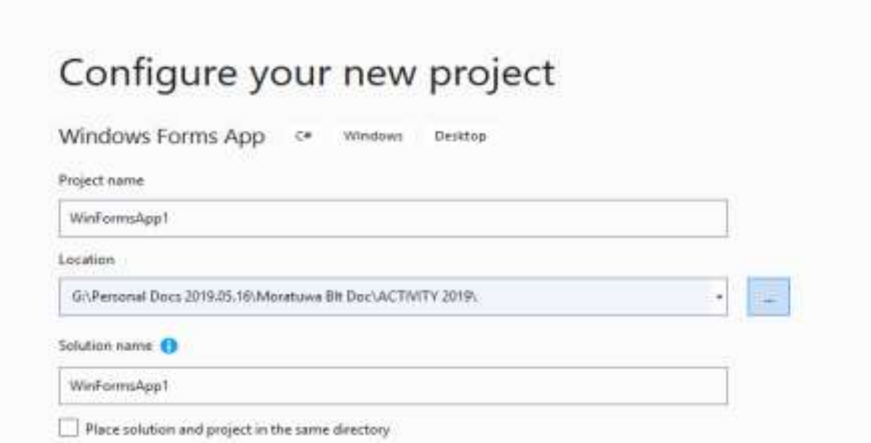
Windows Forms Control Library (.NET Framework) — A project for creating controls to use in Windows Forms (WinForms) applications — C# Windows Desktop Library

Windows Forms Control Library (.NET Framework) — A project for creating controls to use in Windows Forms (WinForms) applications — Visual Basic Windows Desktop Library

Blank App (Universal Windows) — A project for a single-page Universal Windows Platform (UWP) app that has no predefined controls or layout. — C# XAML Windows Xbox UWP Desktop

Blank App (Universal Windows) — A project for a single-page Universal Windows Platform (UWP) app that has no predefined controls or layout. — Visual Basic XAML Windows Xbox UWP Desktop

3. Config your project as follow
    - Name the application as you need. Ex: WinformApp1



Configure your new project

Windows Forms App   C#   Windows   Desktop

Project name

WinFormsApp1

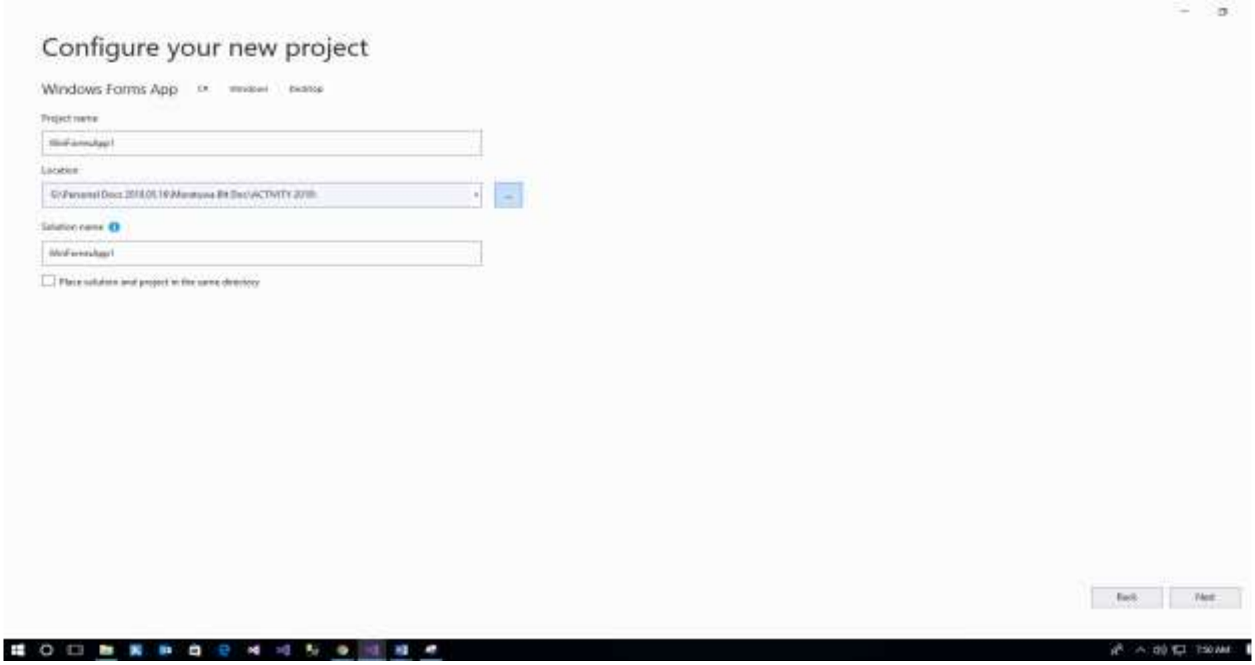
Location

G:\Personal Docs 2019.05.16\Moratuwa Blt Doc\ACTIVITY 2019\

Solution name

WinFormsApp1

□ Place solution and project in the same directory



Configure your new project

Windows Forms App   C#   Windows   Desktop

Project name

WinFormsApp1

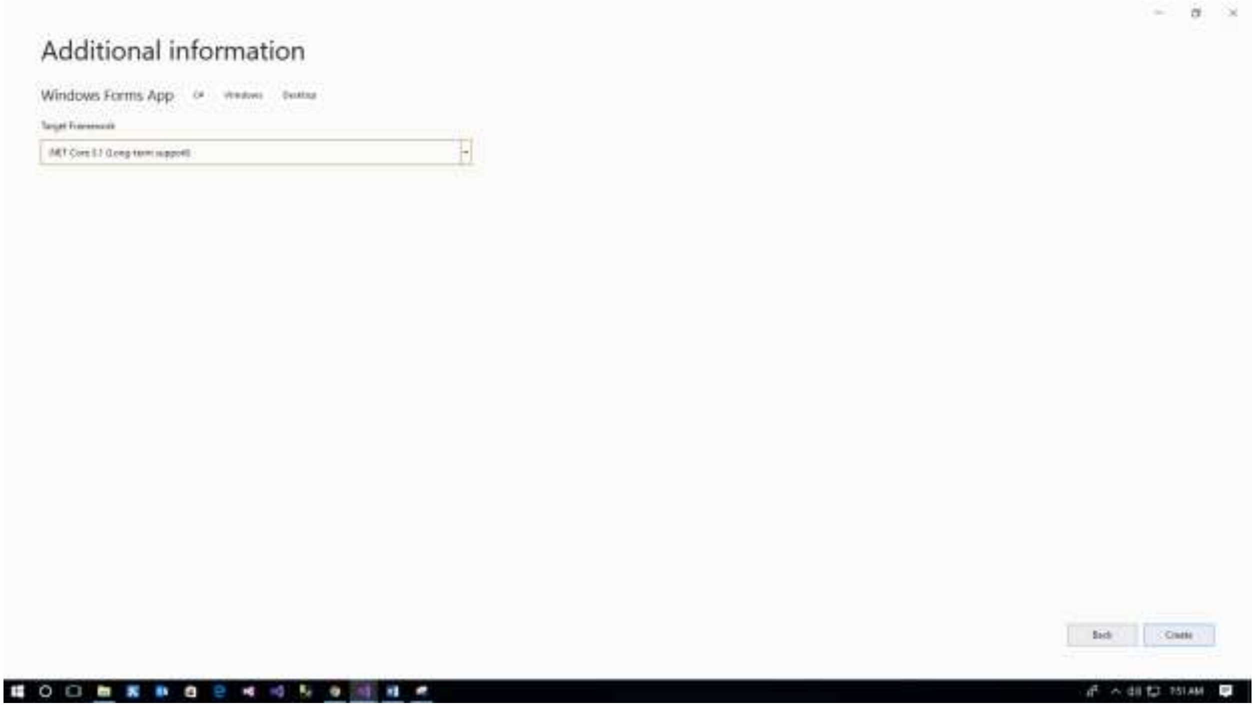
Location

G:\Personal Docs 2019.05.16\Moratuwa Blt Doc\ACTIVITY 2019\

Solution name

WinFormsApp1

□ Place solution and project in the same directory



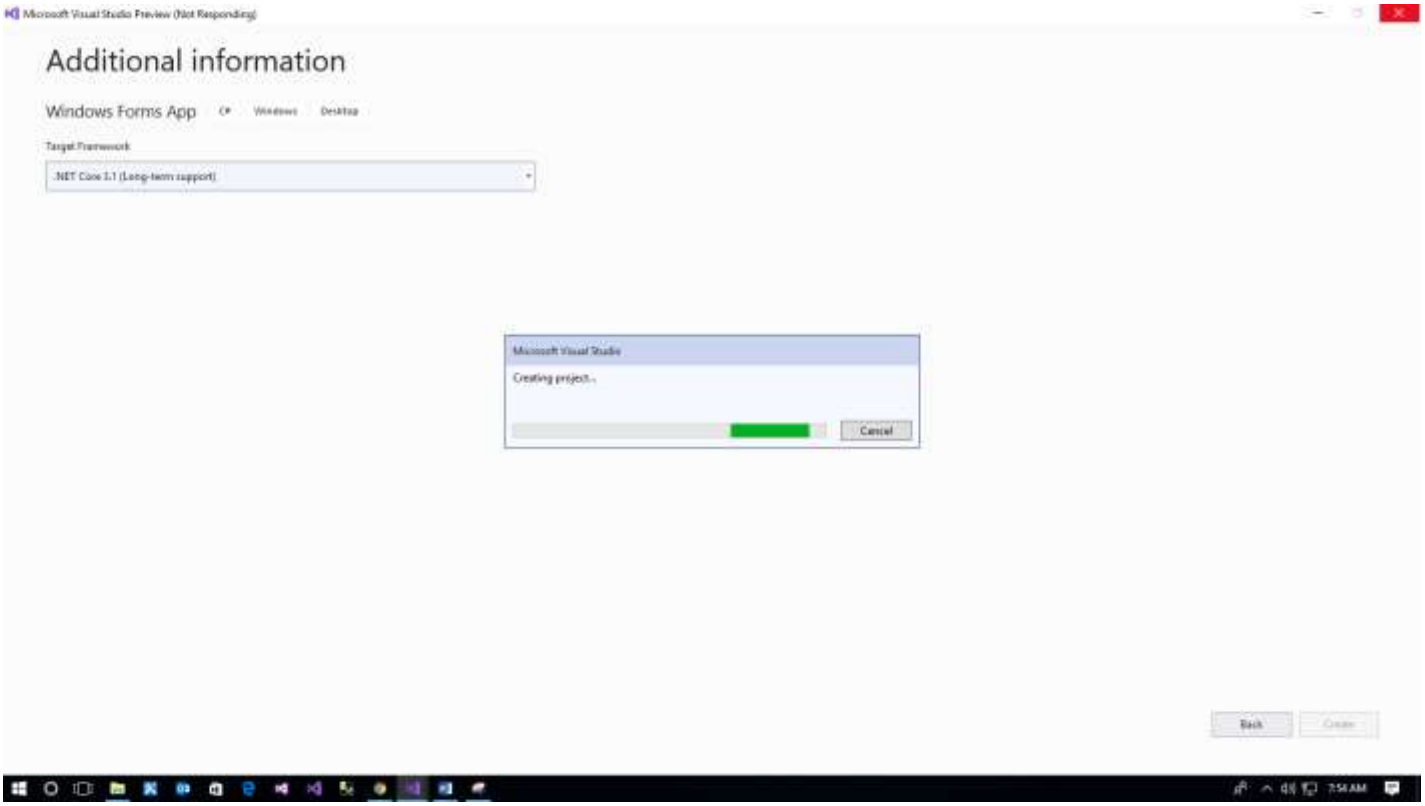
Additional information

Windows Forms App C# Windows Desktop

Target Framework

.NET Core 3.1 (Long-term support)

Then Project will be created as follow



Microsoft Visual Studio Preview (Not Responding)

Additional information

Windows Forms App C# Windows Desktop

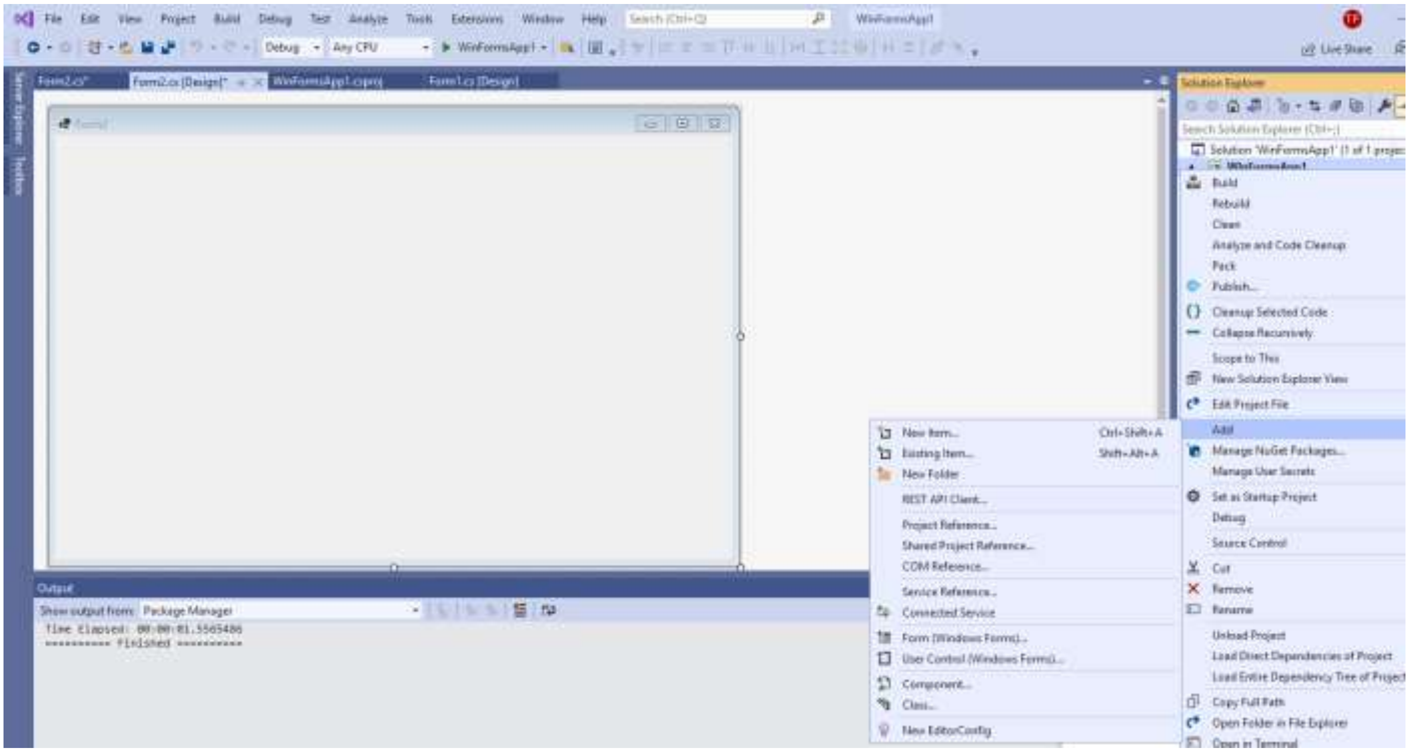
Target Framework

.NET Core 3.1 (Long-term support)

Microsoft Visual Studio

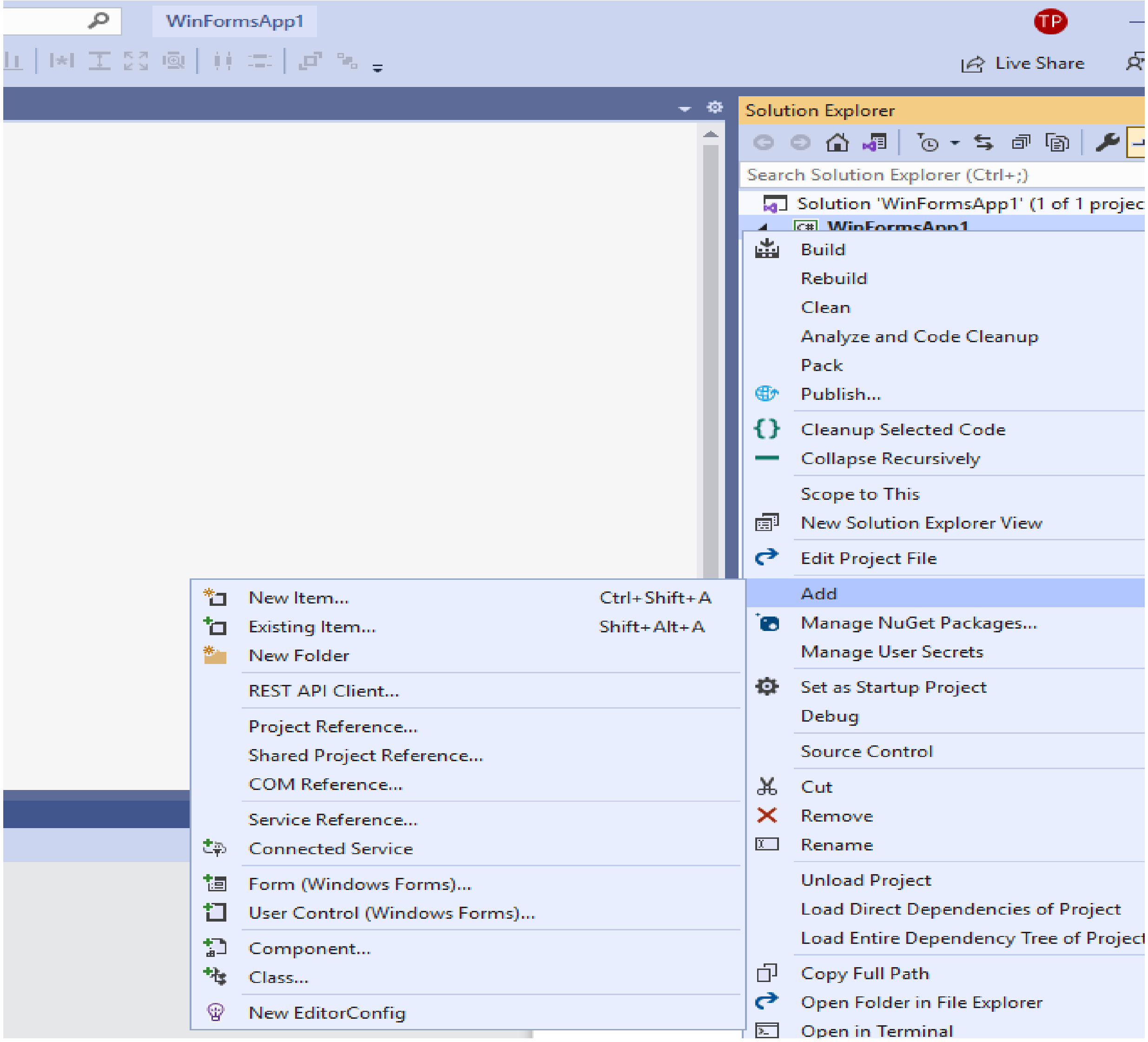
Creating project...

4. Click on the project and you can Add new forms for the project as follow

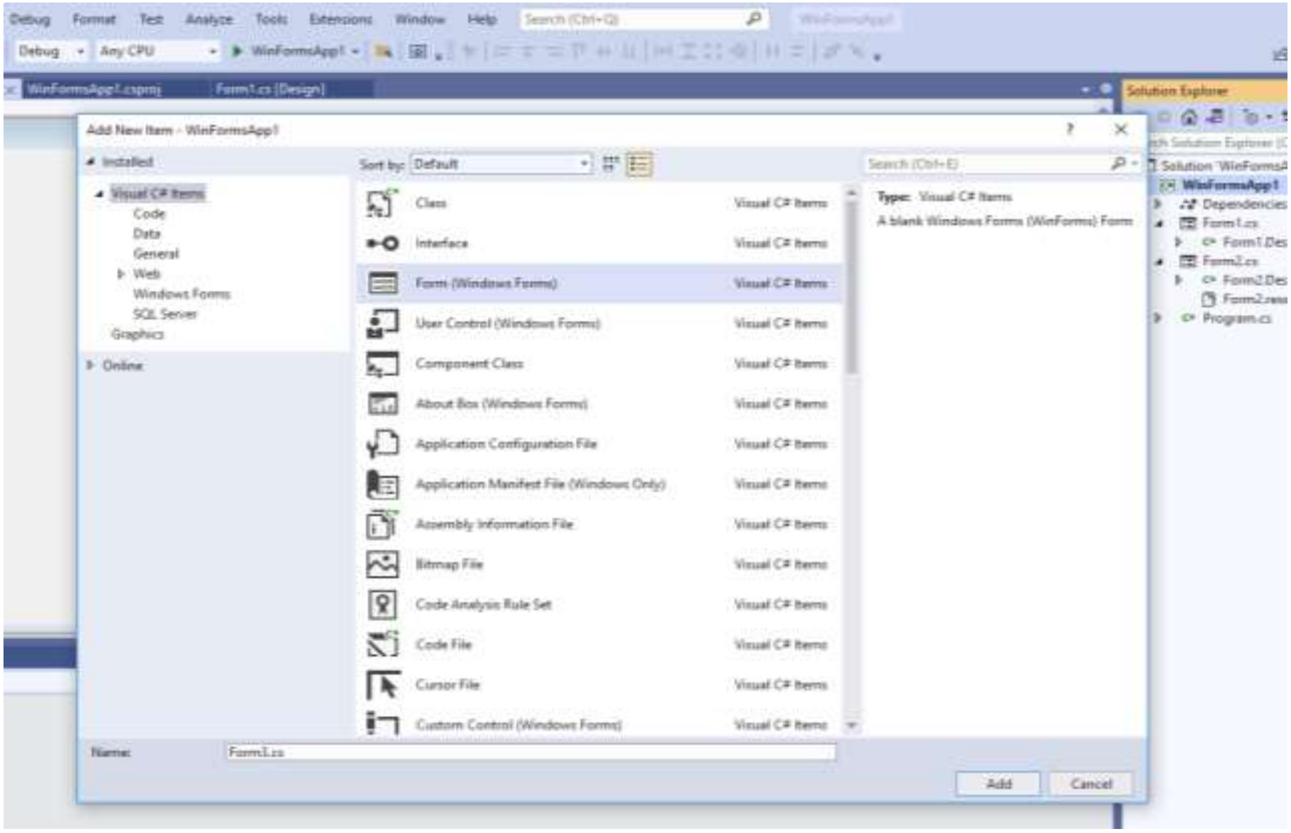


```
Time Elapsed: 00:00:01.5565486
========== Finished ==========
```

Then select new Item



Next Select Form (Windows form)

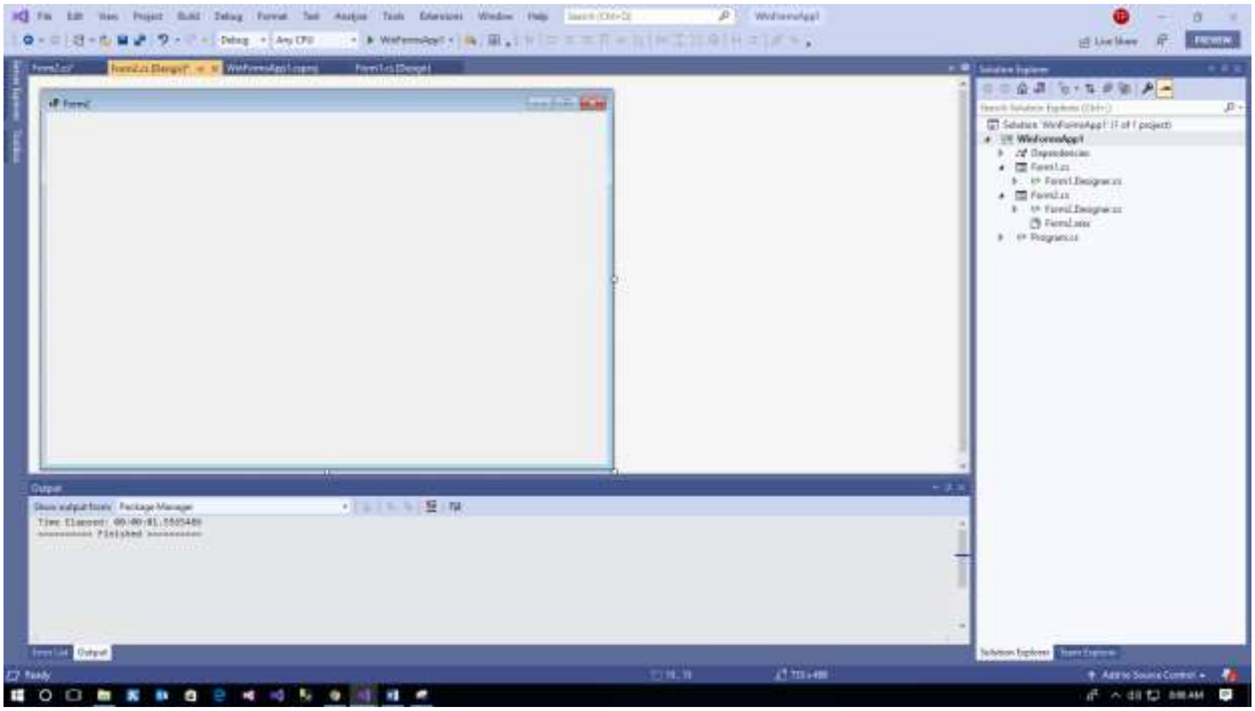


Type: Visual C# Items

A blank Windows Forms (WinForms) Form

Name: Form3.cs

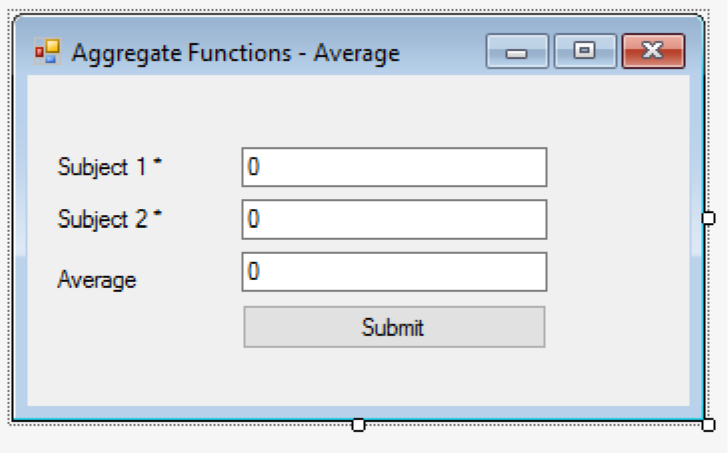
5. Next you ready to design the on the windows form with Toolbox.



```
Time Elapsed: 00:00:01.5565480
========== Finished ==========
```

As below you can design your form by dragging tools on the forms form the toolbox: (Hope now you have some idea how to design windows application with practice in previous lab sheet.)

6. Create new application and default form design as below with toolbox.



Aggregate Functions - Average

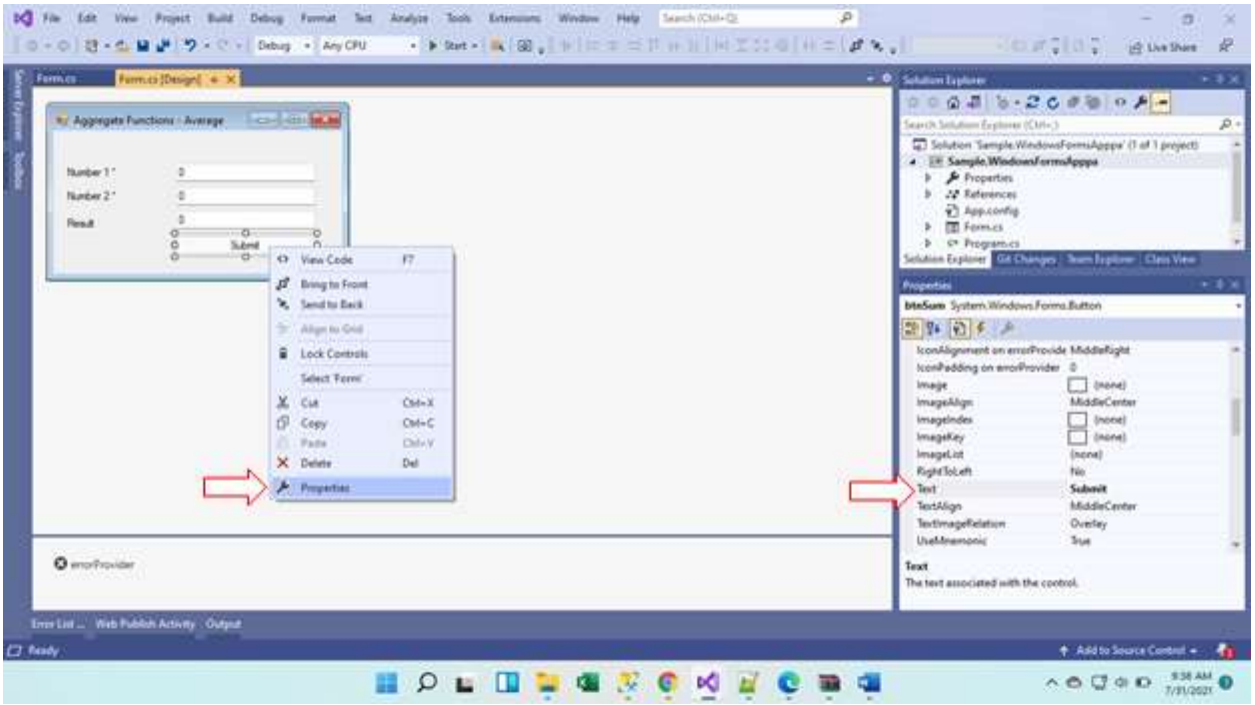
Subject 1 * 0

Subject 2 * 0

Average 0

By practicing in previous activity now you know how to design form with tool box.

7. **Next you name all control and add display text for buttons and labels and add other features with property window.**



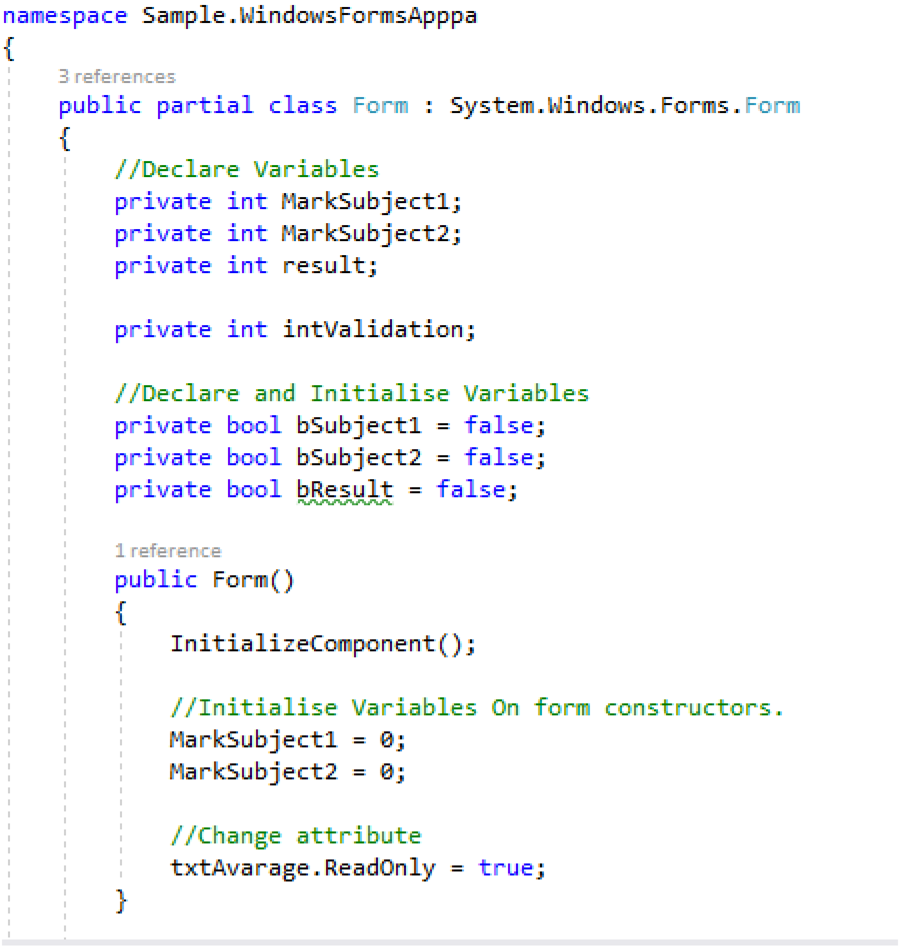
btnSum System.Windows.Forms.Button

IconAlignment on errorProvide	MiddleRight
IconPadding on errorProvider	0
Image	(none)
ImageAlign	MiddleCenter
ImageIndex	(none)
ImageKey	(none)
ImageList	(none)
RightToLeft	No
Text	**Submit**
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseMnemonic	True

Text
The text associated with the control.

8. After you compete the naming for all controls next right click on the form and select the **view Code.**

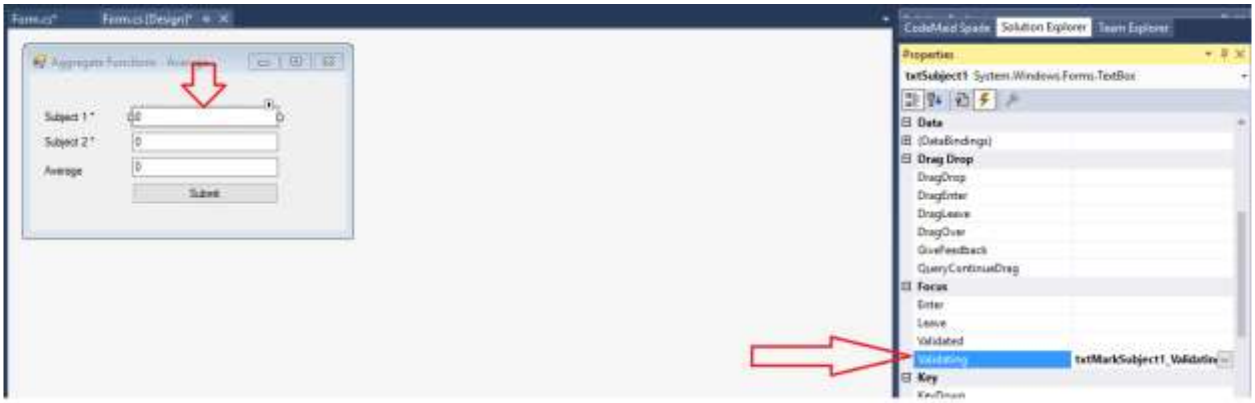
   - Now you can view form.cs. In form.cs you can initiate the variable as below.
   - Declare variable and initialize the variable following below coding.
   - Use access modifiers as privet (variable cannot access out of the form.cs).
   - Do read only for the txtAverage (then textbox cannot edit)



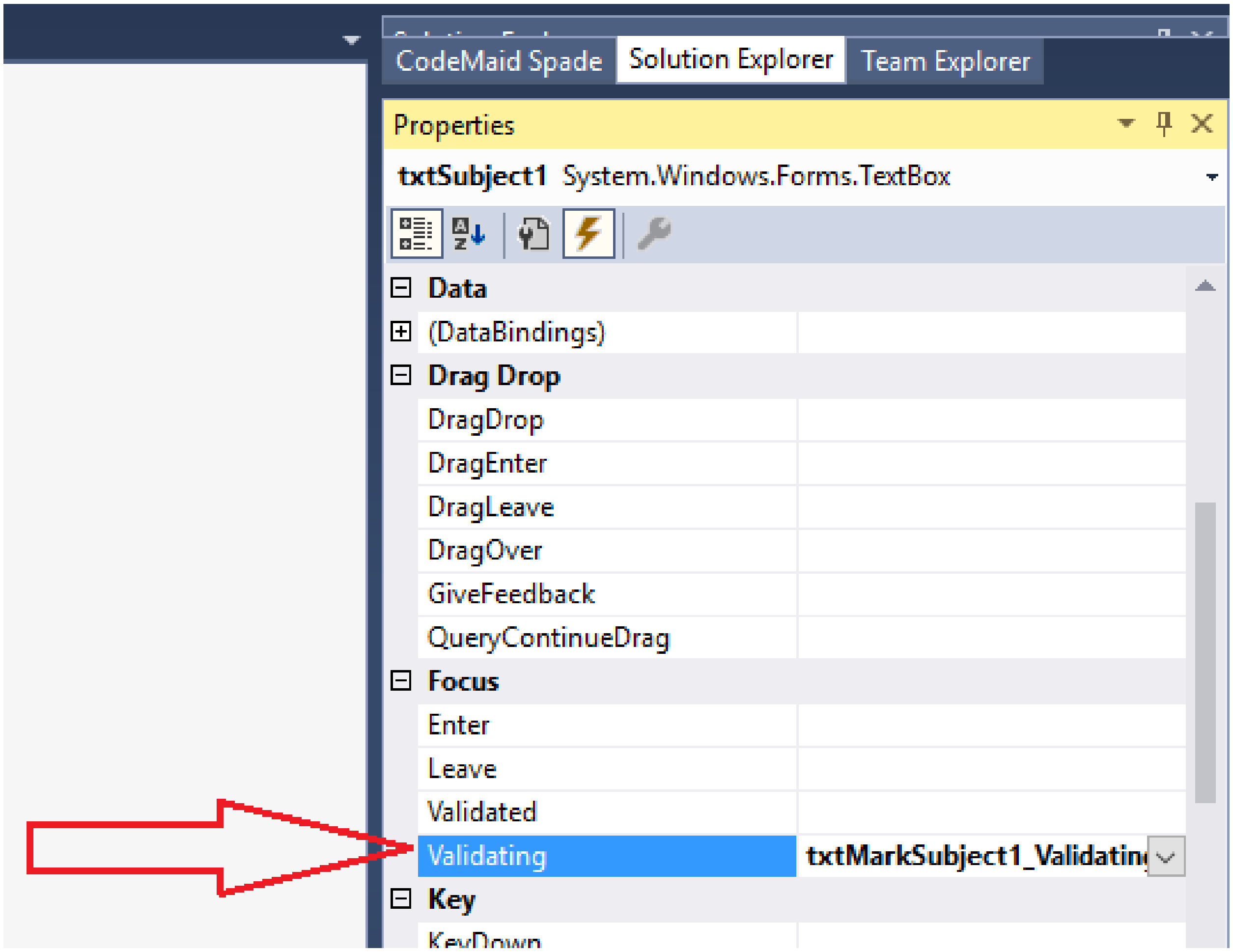
```
namespace Sample.WindowsFormsApppa
{
    3 references
    public partial class Form : System.Windows.Forms.Form
    {
        //Declare Variables
        private int MarkSubject1;
        private int MarkSubject2;
        private int result;

        private int intValidation;

        //Declare and Initialise Variables
        private bool bSubject1 = false;
        private bool bSubject2 = false;
        private bool bResult = false;

        1 reference
        public Form()
        {
            InitializeComponent();

            //Initialise Variables On form constructors.
            MarkSubject1 = 0;
            MarkSubject2 = 0;

            //Change attribute
            txtAvarage.ReadOnly = true;
        }
```

9. Next you need to write event for the validation.

- Select the validating event under the focus menu in the property event.
- Next you can validate the text field as below.



Validating txtMarkSubject1_Validatin



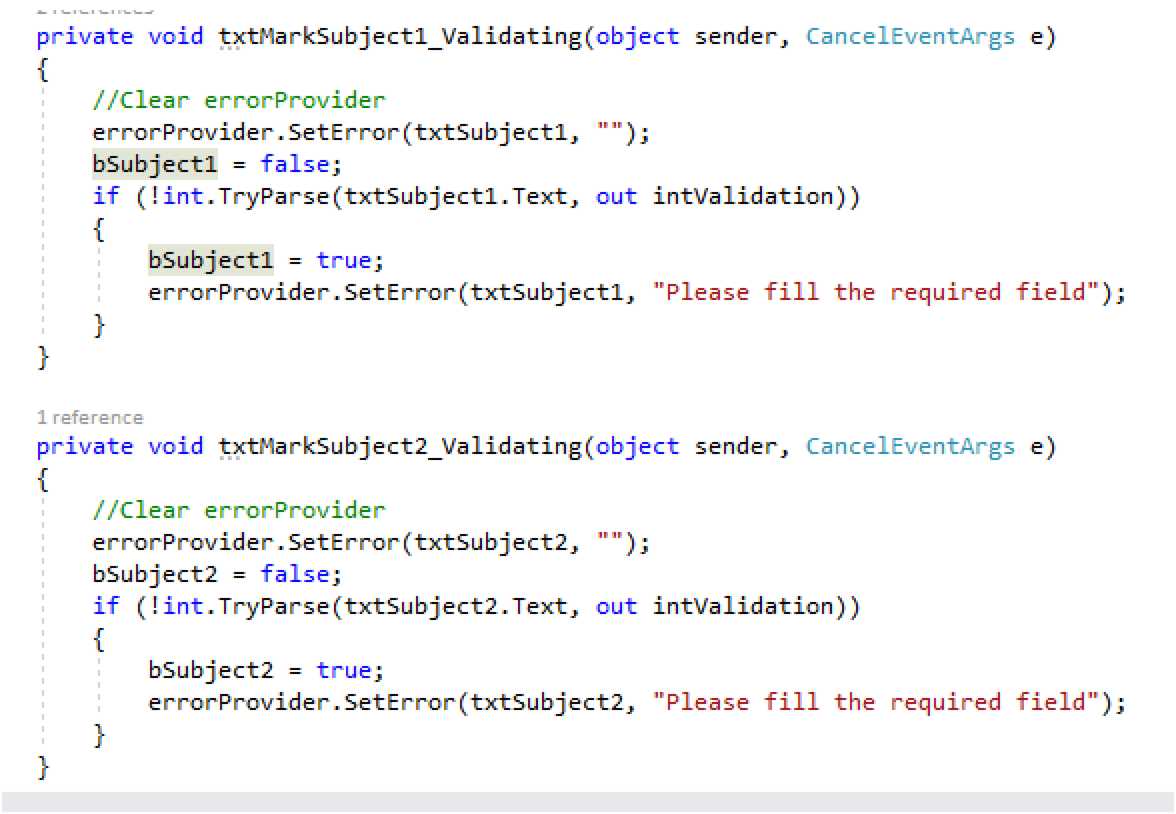
Properties

txtSubject1 System.Windows.Forms.TextBox

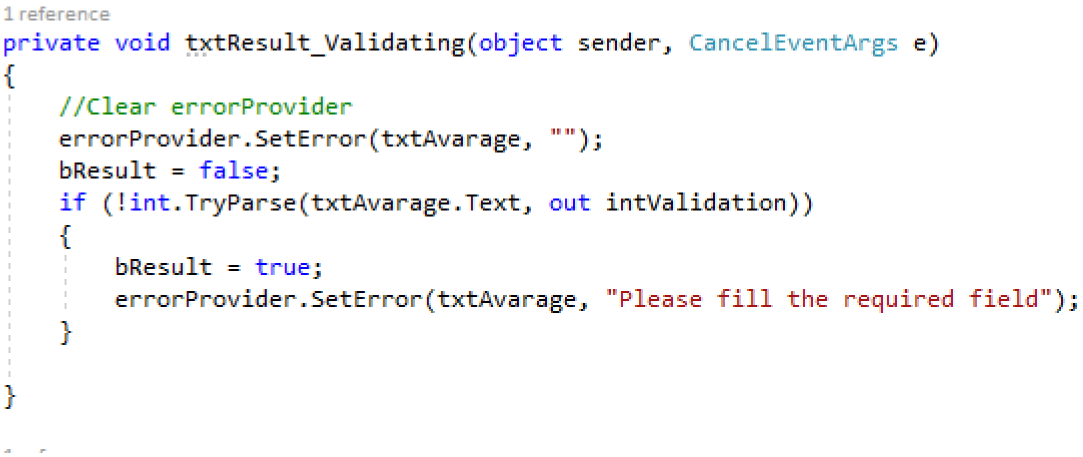
⊟ Data
⊞ (DataBindings)
⊟ Drag Drop
DragDrop
DragEnter
DragLeave
DragOver
GiveFeedback
QueryContinueDrag
⊟ Focus
Enter
Leave
Validated
Validating txtMarkSubject1_Validatin
⊟ Key
KeyDown

Code for validating event

- Here you can use if statement for the validating.



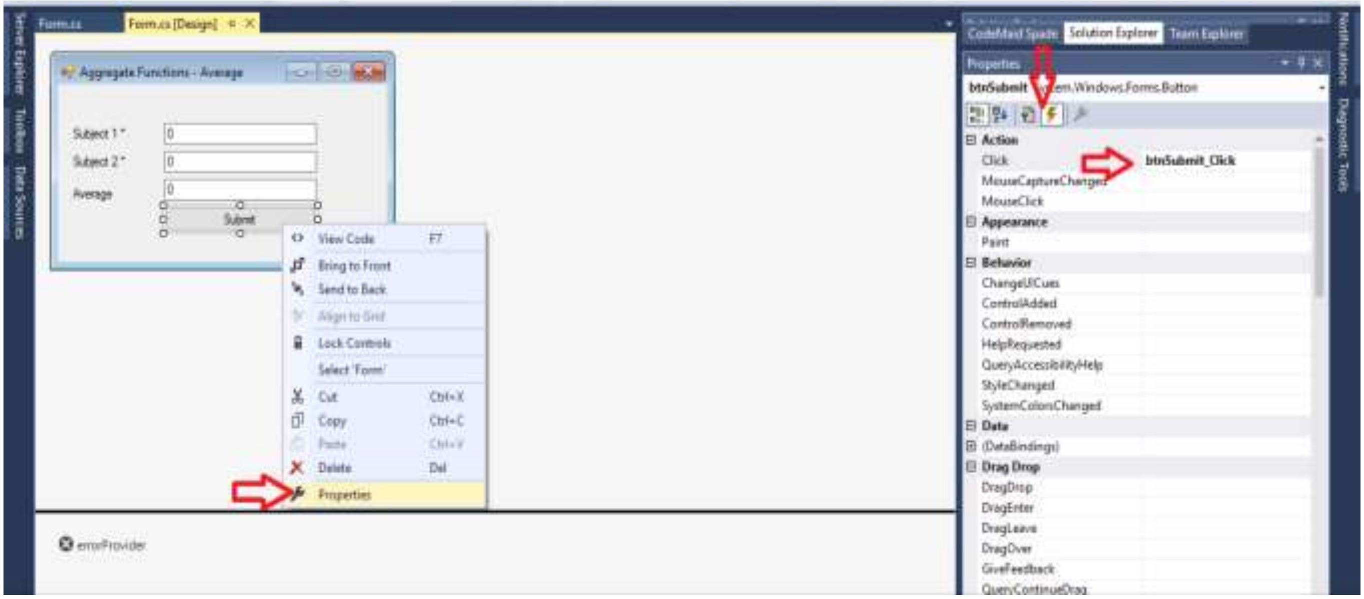
```
private void txtMarkSubject1_Validating(object sender, CancelEventArgs e)
{
    //Clear errorProvider
    errorProvider.SetError(txtSubject1, "");
    bSubject1 = false;
    if (!int.TryParse(txtSubject1.Text, out intValidation))
    {
        bSubject1 = true;
        errorProvider.SetError(txtSubject1, "Please fill the required field");
    }
}

1 reference
private void txtMarkSubject2_Validating(object sender, CancelEventArgs e)
{
    //Clear errorProvider
    errorProvider.SetError(txtSubject2, "");
    bSubject2 = false;
    if (!int.TryParse(txtSubject2.Text, out intValidation))
    {
        bSubject2 = true;
        errorProvider.SetError(txtSubject2, "Please fill the required field");
    }
}
```



```
1 reference
private void txtResult_Validating(object sender, CancelEventArgs e)
{
    //Clear errorProvider
    errorProvider.SetError(txtAvarage, "");
    bResult = false;
    if (!int.TryParse(txtAvarage.Text, out intValidation))
    {
        bResult = true;
        errorProvider.SetError(txtAvarage, "Please fill the required field");
    }

}
```

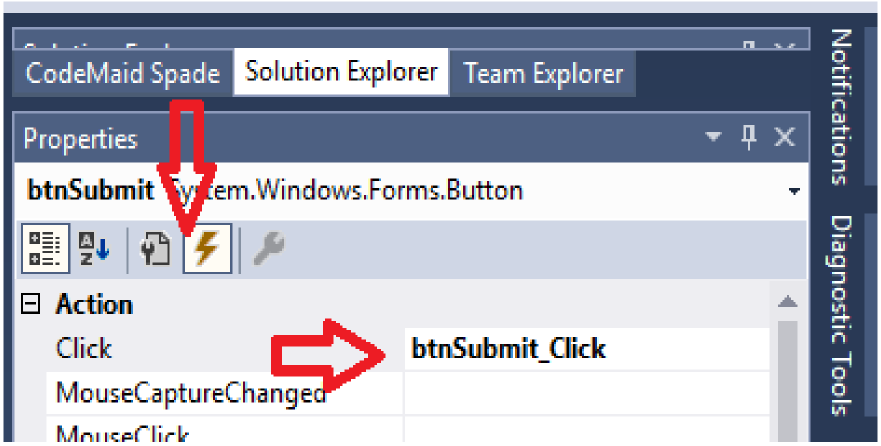
10. You can write submit button click event as below.

- Right click on the **submit button** and select the properties
- Next select the event in the property window.
- After that select the **click** event.
- Next write **btnsubmit_click** event



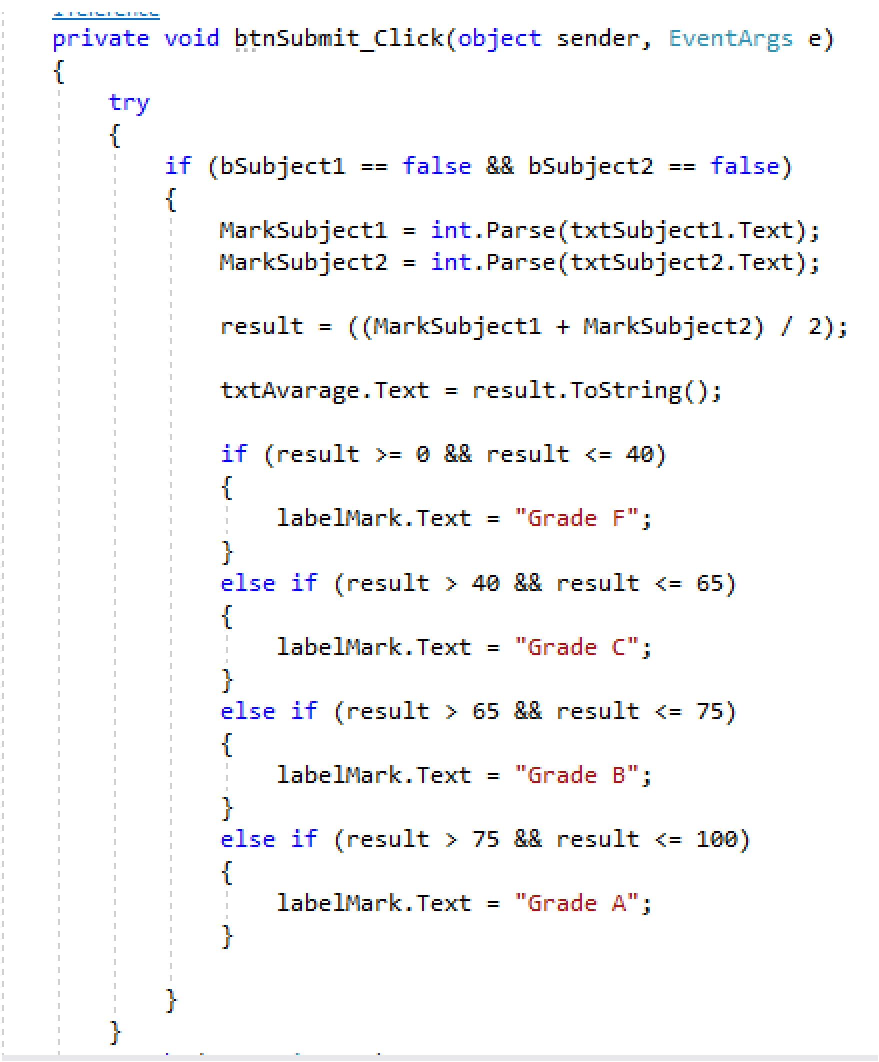
Subject 1 * 0
Subject 2 * 0
Average 0

Click btnSubmit_Click



Click btnSubmit_Click

Coding for Submit button click event

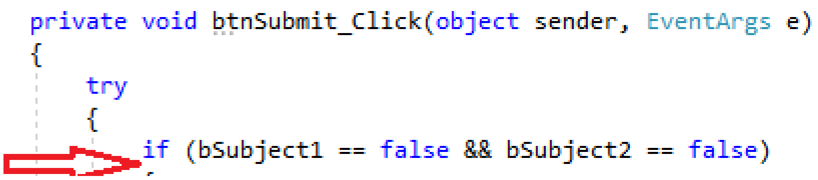


```
private void btnSubmit_Click(object sender, EventArgs e)
{
    try
    {
        if (bSubject1 == false && bSubject2 == false)
        {
            MarkSubject1 = int.Parse(txtSubject1.Text);
            MarkSubject2 = int.Parse(txtSubject2.Text);

            result = ((MarkSubject1 + MarkSubject2) / 2);

            txtAvarage.Text = result.ToString();

            if (result >= 0 && result <= 40)
            {
                labelMark.Text = "Grade F";
            }
            else if (result > 40 && result <= 65)
            {
                labelMark.Text = "Grade C";
            }
            else if (result > 65 && result <= 75)
            {
                labelMark.Text = "Grade B";
            }
            else if (result > 75 && result <= 100)
            {
                labelMark.Text = "Grade A";
            }

        }
    }
```

Here conditions have checked with If statement. Be practice if statement well with this application.

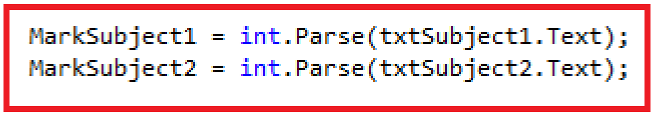
Steps:

- First you need check validating variables in the if statement.



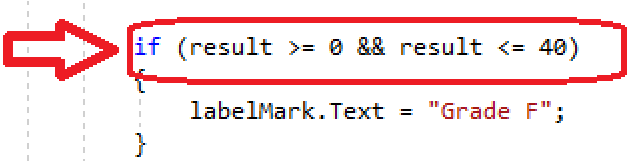
```
private void btnSubmit_Click(object sender, EventArgs e)
{
    try
    {
        if (bSubject1 == false && bSubject2 == false)
```

- Next you need to assign text box value to the related variable in if statement one.



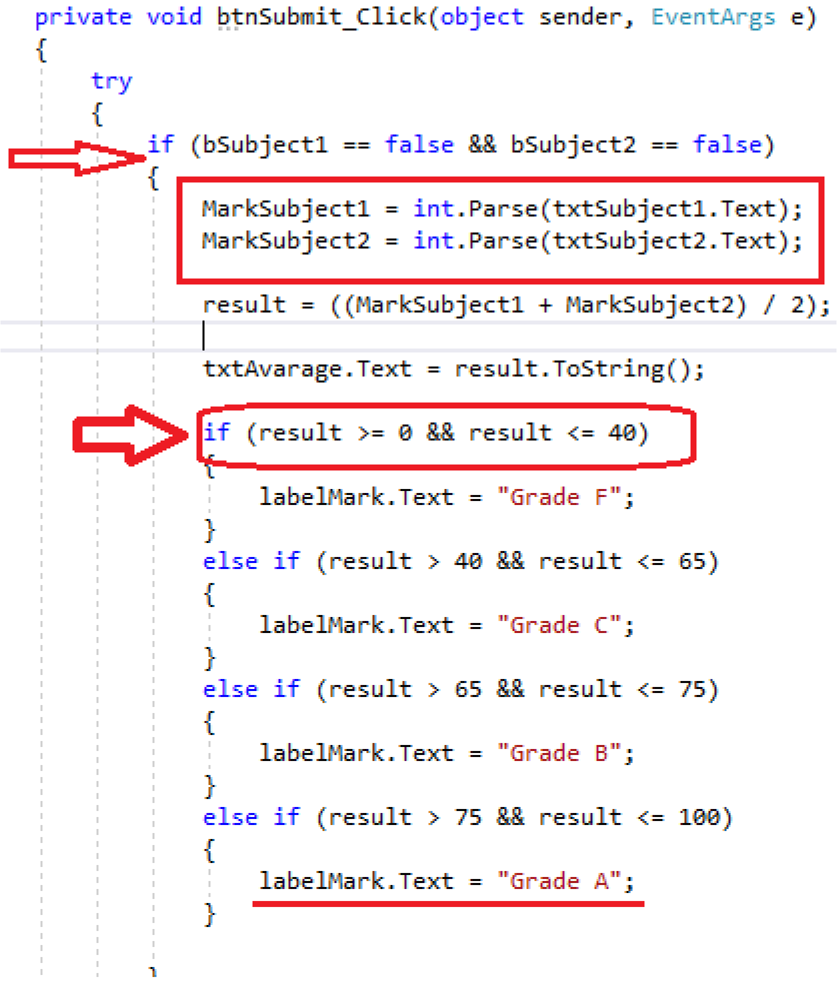
```
MarkSubject1 = int.Parse(txtSubject1.Text);
MarkSubject2 = int.Parse(txtSubject2.Text);
```

- After that you need to check the result variable in another if statement.



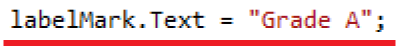
```
if (result >= 0 && result <= 40)
{
    labelMark.Text = "Grade F";
}
```

- Same way checks the result further as need in if else and else statement

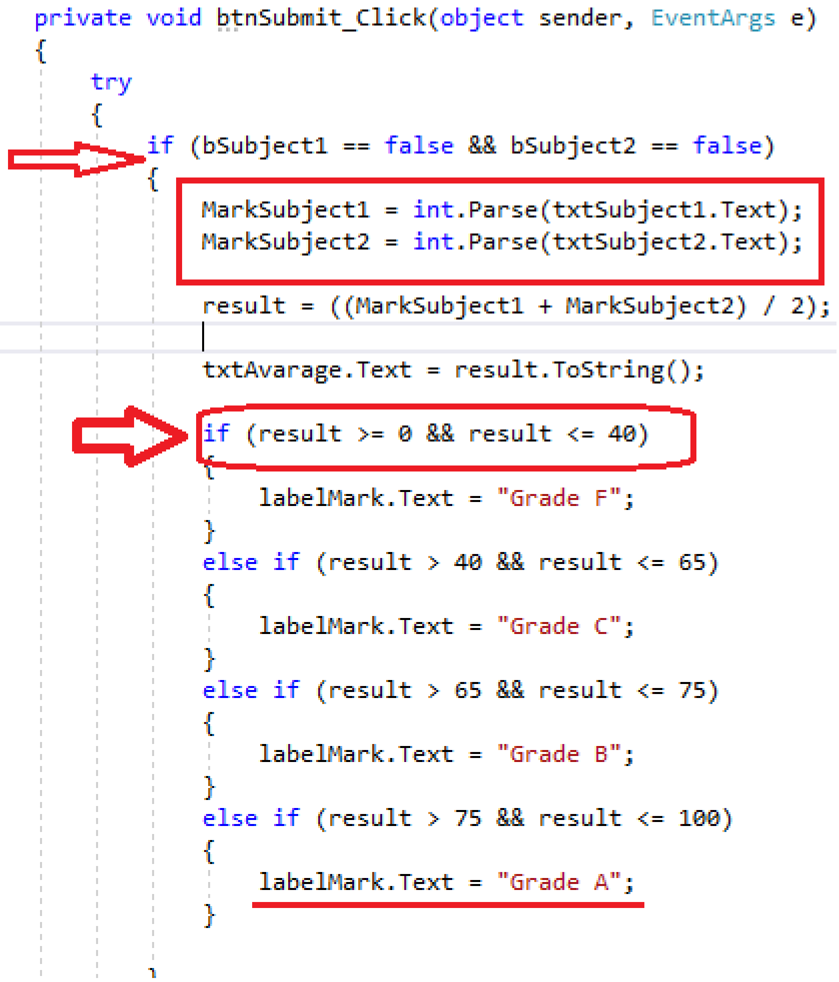


```
private void btnSubmit_Click(object sender, EventArgs e)
{
    try
    {
        if (bSubject1 == false && bSubject2 == false)
        {
            MarkSubject1 = int.Parse(txtSubject1.Text);
            MarkSubject2 = int.Parse(txtSubject2.Text);

            result = ((MarkSubject1 + MarkSubject2) / 2);

            txtAvarage.Text = result.ToString();

            if (result >= 0 && result <= 40)
            {
                labelMark.Text = "Grade F";
            }
            else if (result > 40 && result <= 65)
            {
                labelMark.Text = "Grade C";
            }
            else if (result > 65 && result <= 75)
            {
                labelMark.Text = "Grade B";
            }
            else if (result > 75 && result <= 100)
            {
                labelMark.Text = "Grade A";
            }

        }
```

- finally assign the result you need to display on label according to average mark.

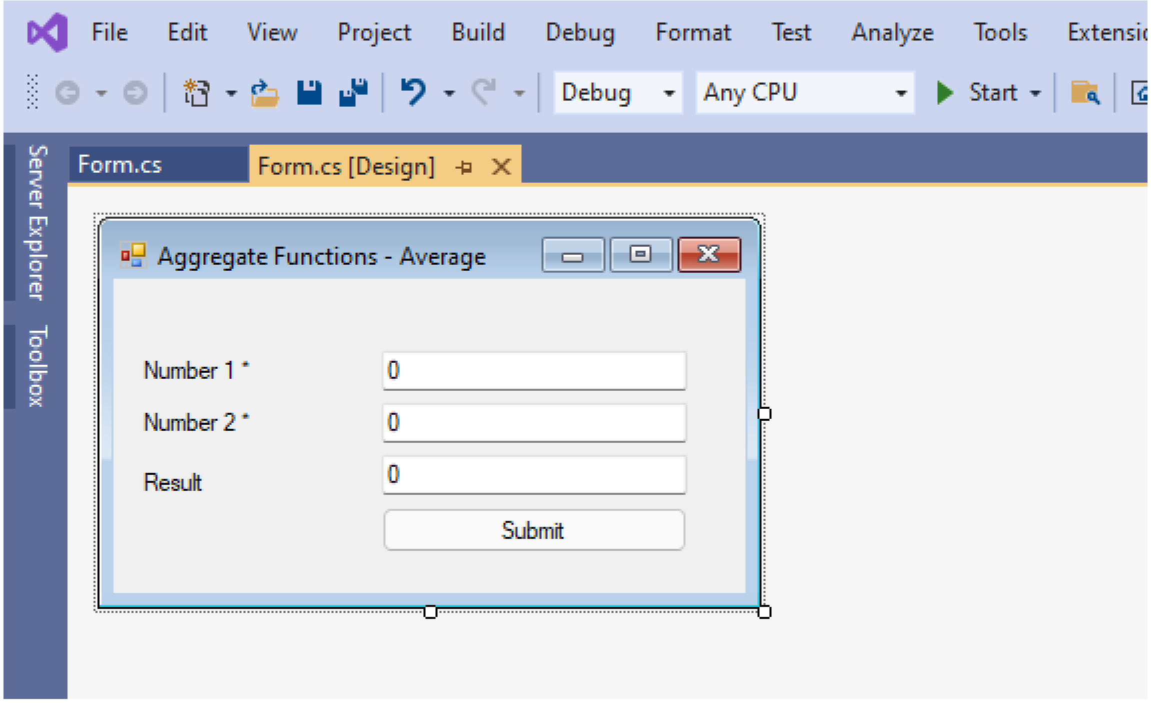


```
labelMark.Text = "Grade A";
```



```
private void btnSubmit_Click(object sender, EventArgs e)
{
    try
    {
        if (bSubject1 == false && bSubject2 == false)
        {
            MarkSubject1 = int.Parse(txtSubject1.Text);
            MarkSubject2 = int.Parse(txtSubject2.Text);

            result = ((MarkSubject1 + MarkSubject2) / 2);

            txtAvarage.Text = result.ToString();

            if (result >= 0 && result <= 40)
            {
                labelMark.Text = "Grade F";
            }
            else if (result > 40 && result <= 65)
            {
                labelMark.Text = "Grade C";
            }
            else if (result > 65 && result <= 75)
            {
                labelMark.Text = "Grade B";
            }
            else if (result > 75 && result <= 100)
            {
                labelMark.Text = "Grade A";
            }

        }
```

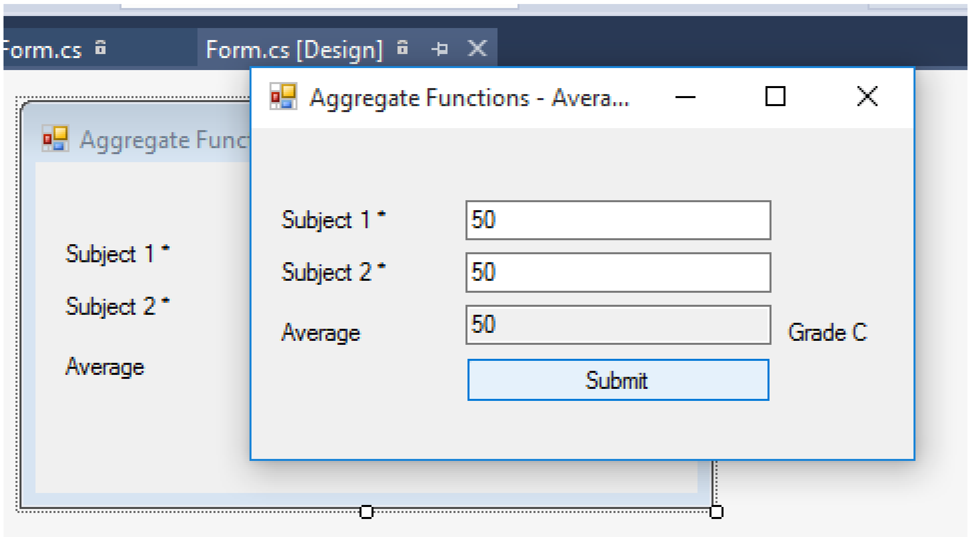
11. Finally run the application



Number 1 * 0

Number 2 * 0

Result 0



Subject 1 * 50

Subject 2 * 50

Average 50 Grade C

- End of the activity, you have got some ideas about how to develop simple application with functions, event, if statements with C# with visual studio.

- Please follow the above Steps further and rewrite different event using C# in visual studio.