

BIL2005

Görsel

Programlama

Dilleri

VISUAL PROGRAMMING LANGUAGES

- Department of Computer Science

VISUAL PROGRAMMING LANGUAGES

- **Lecturer:**

- Assoc. Prof. Dr. Emel KURUOĞLU KANDEMİR
- emel.kuruoglu@deu.edu.tr

- **Lecturer Assistant:**

- Research Assistant Alican Doğan
- alican.dogan@deu.edu.tr

VISUAL PROGRAMMING LANGUAGES

Course Evaluation

- Homework + Quiz 35%
- Midterm 25%
- Final 40%

Course Hours

- Tuesday 08:30 – 12:00 / 13.00 – 16.30

Attendance

- Students have to attend %70 of all lectures.
- $14 \times 2 = 28$ signatures
- $28 \times 0.3 = 8.4$
- Students with 9 non-attendance will fail...

VISUAL PROGRAMMING LANGUAGES

Introduction to Visual Programming Environment

Introduction to Visual Studio

Introduction to Object Oriented Programming

- Classes, Objects
- Properties
- Methods

Form applications

- Form, Button, Text Box, List Box, Check Box, Combo Box, etc.

C# programming language

- Data Types and Variables,
- Control Flow,
- Loops,
- Arrays,


Visual Programming

Microsoft Visual Studio

Microsoft Visual Studio

- Integrated Development Environment (IDE)
- Graphical User Interface
- Windows Form Applications
- Forms,
 - Button
 - textBox
 - checkBox
 - comboBox
 - listBox, etc.
- Environment for programming languages
 - C, C++,
 - C#, F#
 - Visual Basic, Python, Azure ML
- Programming support
 - Code completion,
 - Tools and tips,
 - Easy coding
- .NET Framework
 - Application development platform

 Visual Studio 2019

 Neowin

Visual Programming Environment

Microsoft Visual Studio – History

Product name	Codename	Version number	Release date
Visual Studio 97	Boston	5.0	February 1997
Visual Studio 6.0	Aspen	6.0	June 1998
Visual Studio .NET (2002)	Rainier	7.0	February 13, 2002
Visual Studio .NET 2003	Everett	7.1	April 24, 2003
Visual Studio 2005	Whidbey	8.0	November 7, 2005
Visual Studio 2008	Orcas	9.0	November 19, 2007
Visual Studio 2010	Dev10/Rosario	10.0	April 12, 2010
Visual Studio 2012	Dev11	11.0	September 12, 2012
Visual Studio 2013	Dev12	12.0	October 17, 2013
Visual Studio 2015	Dev14	14.0	July 20, 2015
Visual Studio 2017	Dev15	15.0	March 7, 2017
Visual Studio 2019		16.0	April 2, 2019

Object Oriented Programming

Object Oriented Programming (OOP)

- A programming paradigm
- Based on «*objects*»
- Objects are instants of a «*class*» that define the type of an object
- Objects have
 - Data (attribute/property)
 - Procedures (method)
- Example

class Ev

{

```
    int OdaSayisi;  
    int SalonSayisi;  
    int KapiNo;  
    string Sokakadi;  
    string il_ilce;
```

}

class

car

methods

refuel() getFuel
setSpeed() getSpeed()
drive()

attributes

fuel
maxspeed

Classes and Objects

```
class Daire
{
    double Yaricap;
    double Alan;
    double Cevre;
    double Alanbul (double r)
    {
        double alan;
        alan = 3.14 * r * r;
        return alan;
    }
}
```

```
Daire d = new Daire();
d.Yaricap = 2.34;
double alan = d.Alanbul (d.Yaricap);
```

Classes and Objects

- Computer programs implement algorithms that manipulate the data.
- In object-oriented programming, the programs that manipulate the properties of an object are the object's **methods**.

```
class Bicycle {  
    int cadence = 0;  
    int speed = 0;  
    int gear = 1;  
  
    void changeCadence(int newValue) {  
        cadence = newValue;  
    }  
  
    void changeGear(int newValue) {  
        gear = newValue;  
    }  
  
    void speedUp(int increment) {  
        speed = speed + increment;  
    }  
  
    void applyBrakes(int decrement) {  
        speed = speed - decrement;  
    }  
}
```

Classes and Objects

C# supports the following access modifiers:

- **Public** (*Access is not restricted*)
- **Internal** (*Access is limited to the current assembly*)
- **Protected internal** (*Access is limited to the current assembly OR types derived from the containing class*)
- **Protected** (*Access is limited to the containing class or types derived from the containing class*)
- **Private** (*Access is limited to the containing type*)

Microsoft C# (see sharp)

- ❑ C# is a modern, general-purpose, object-oriented programming language developed by Microsoft
- ❑ C# was developed by Anders Hejlsberg and his team during the development of .NET Framework.
- ❑ C# is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows use of various high-level languages on different computer platforms and architectures.

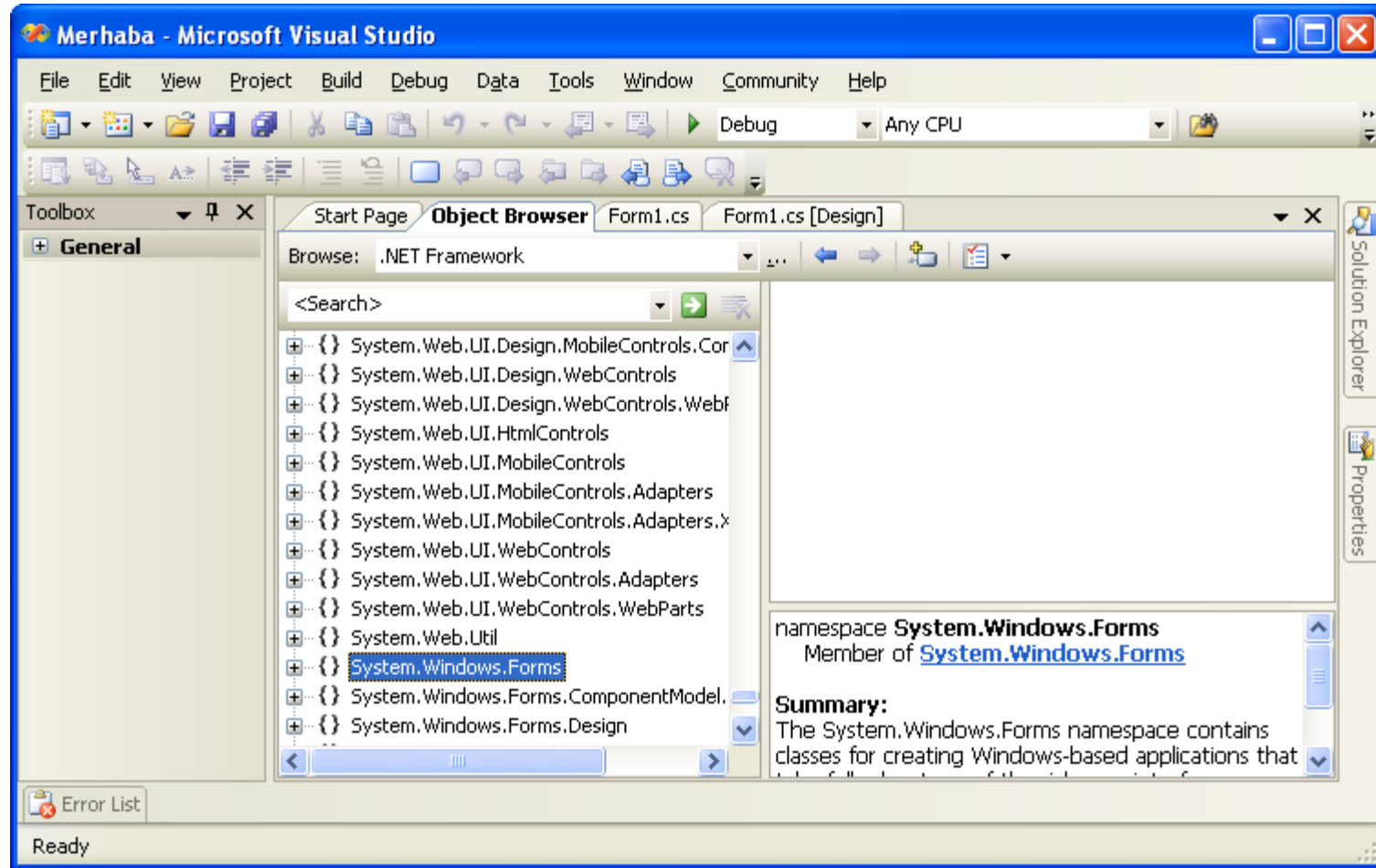
The following reasons make C# a widely used professional language:

- It is a modern, general-purpose programming language
- It is object oriented.
- It is component oriented.
- It is easy to learn.
- It is a structured language.
- It produces efficient programs.
- It can be compiled on a variety of computer platforms.
- It is a part of .NET Framework.

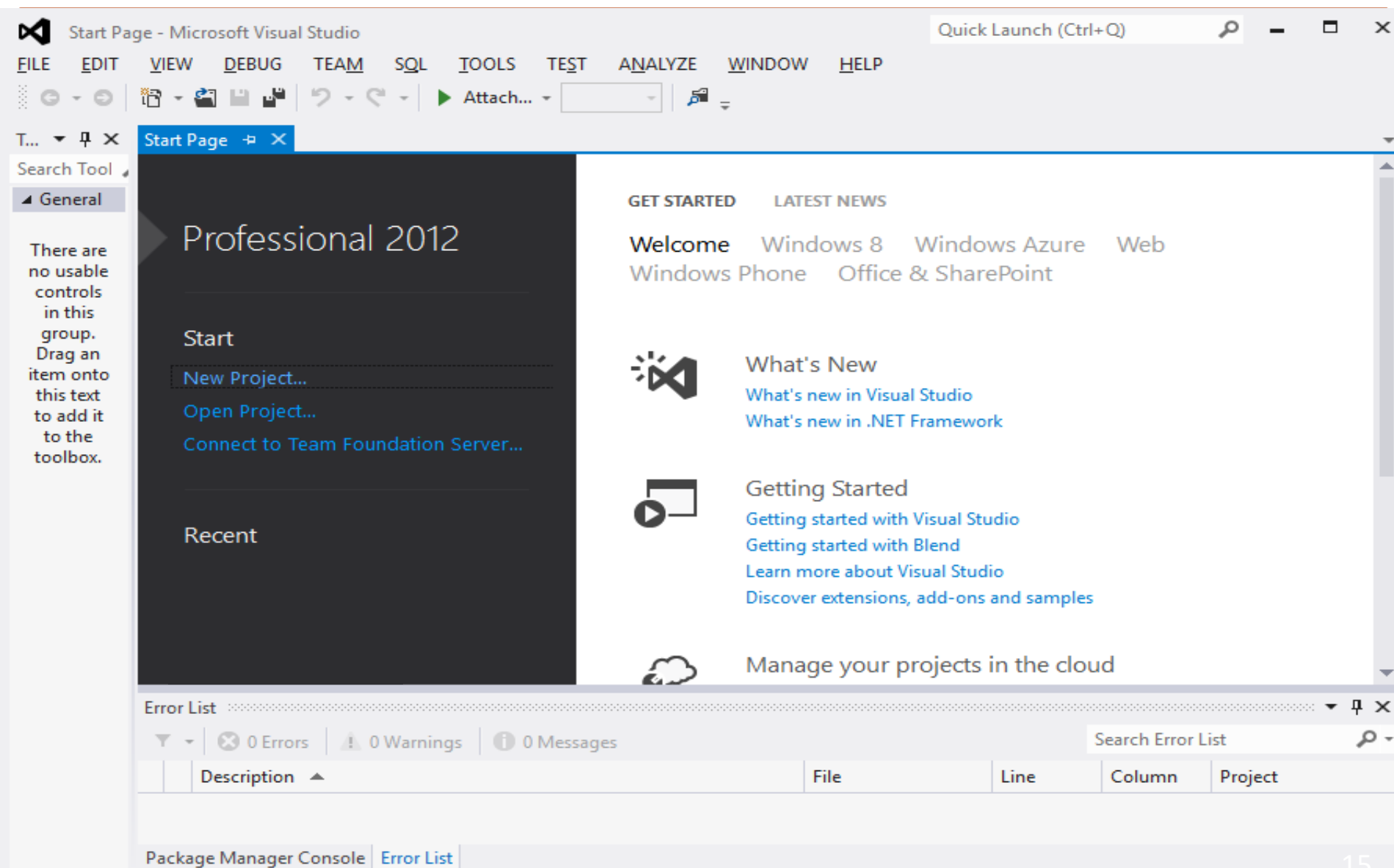
Visual Studio .NET Facilities

- ❖ Code Completion
- ❖ Easy Visual Design, Drag and Drop
- ❖ Listing all the attributes of object while writing Intellisense code
- ❖ Dynamic Help, Coding Examples
- ❖ Immediate Coding Error Warnings
- ❖ Possibility to use different programming languages in one development interface.
- ❖ HTML Editor in which Design and Code are done simultaneously.
- ❖ Improved Debug Tools
- ❖ Combined Compilation and Build Support
- ❖ Automatic Code Demonstration Order

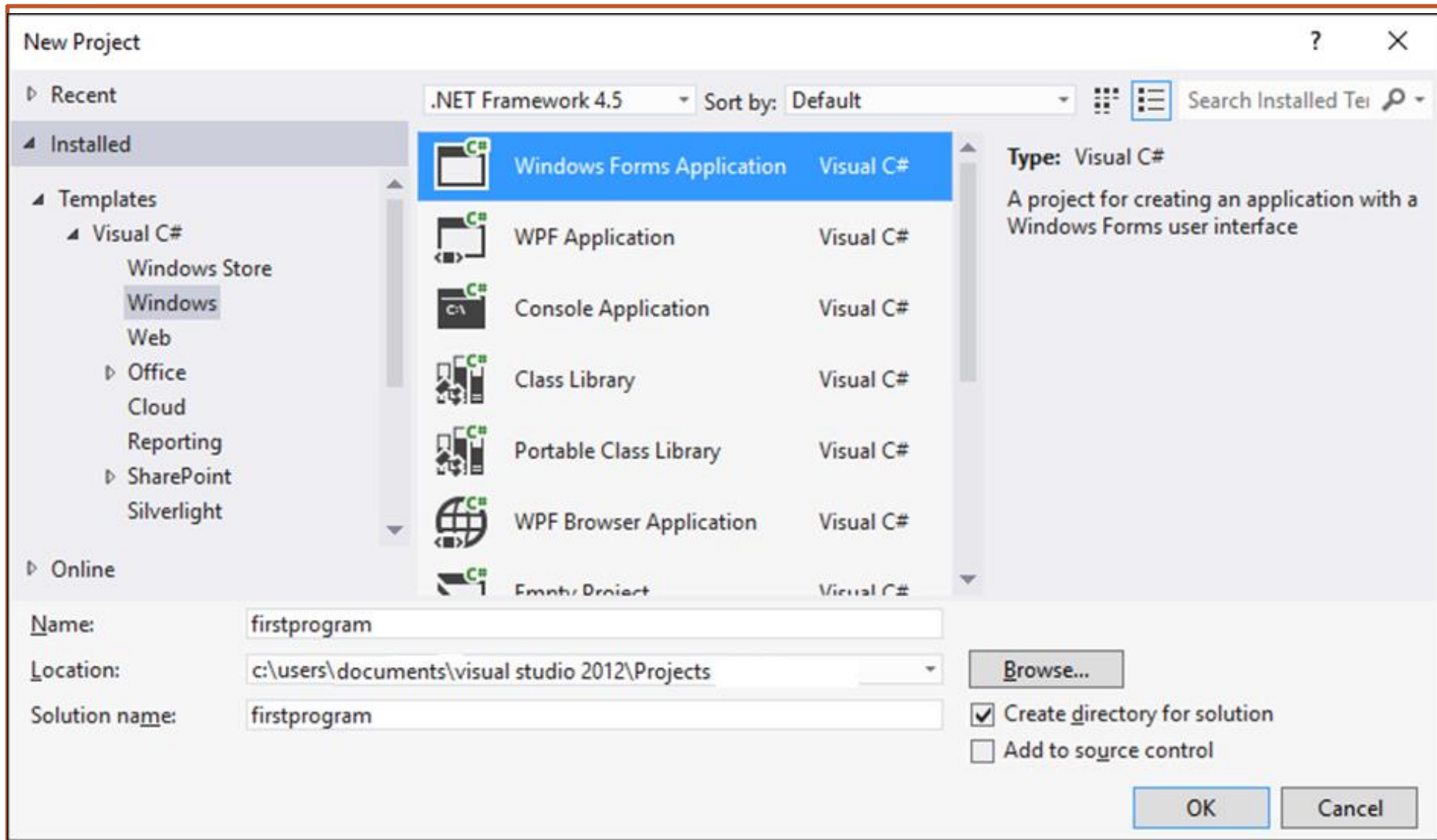
Object Browser

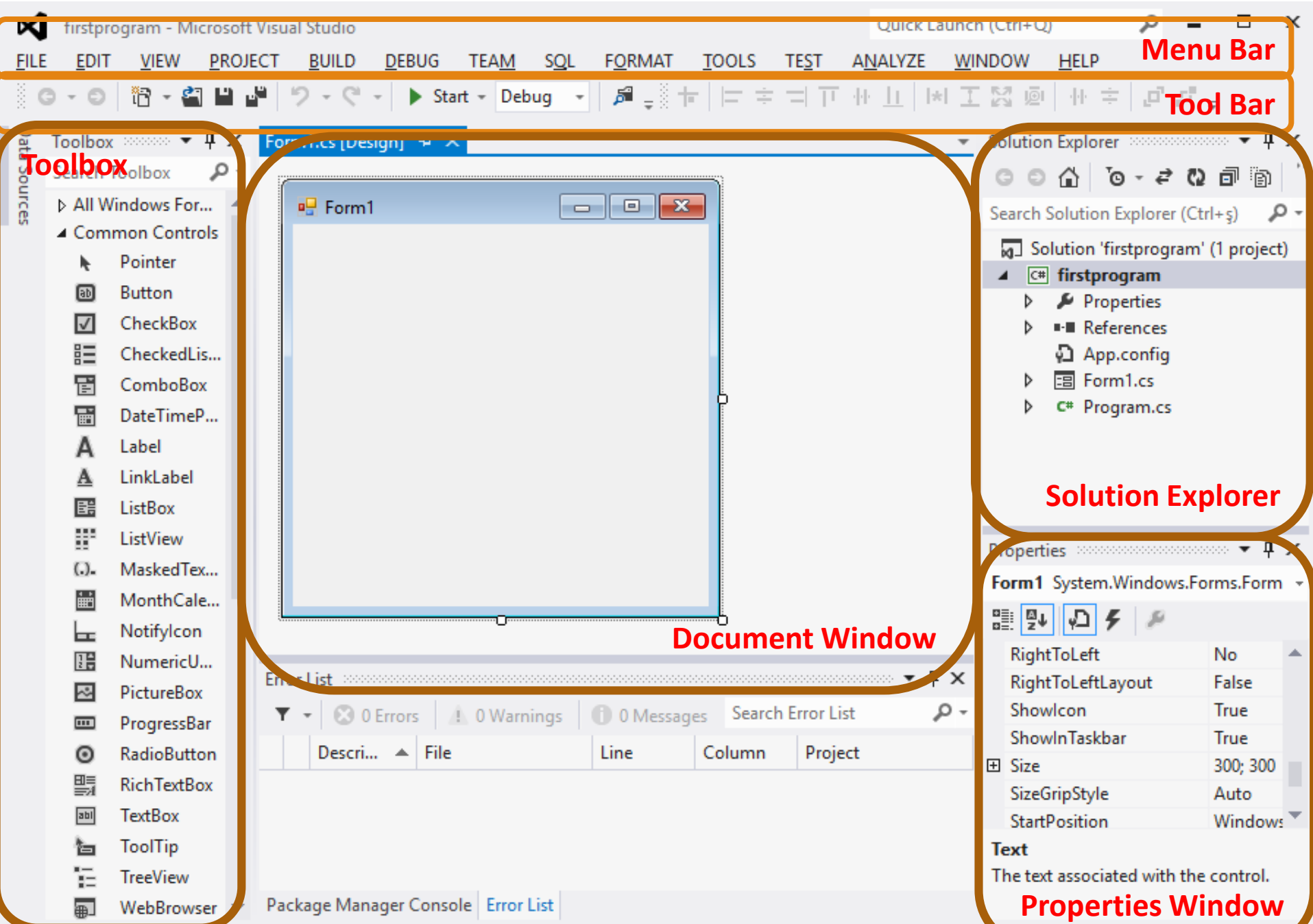


Microsoft Visual Studio

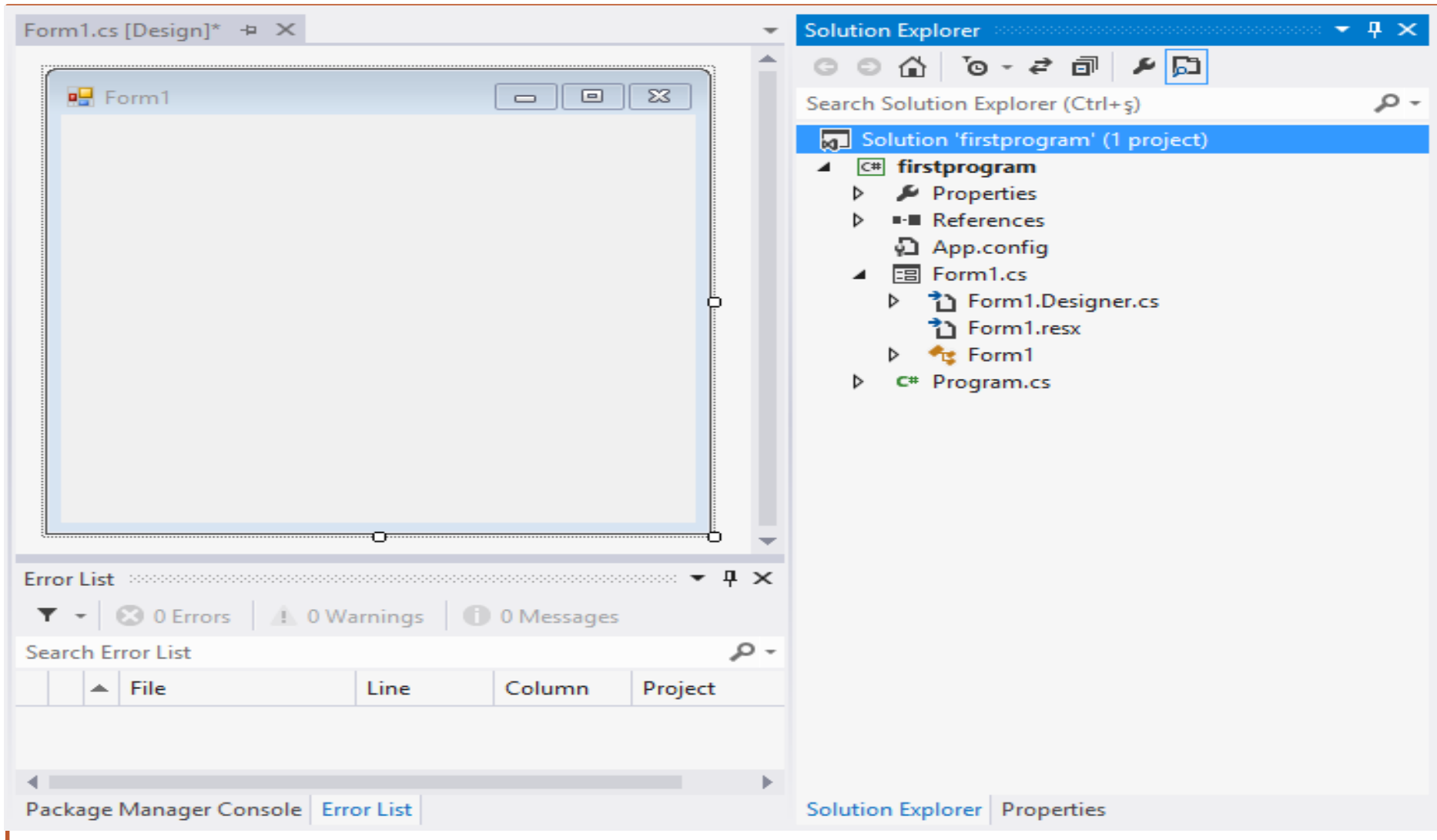


Microsoft Visual Studio



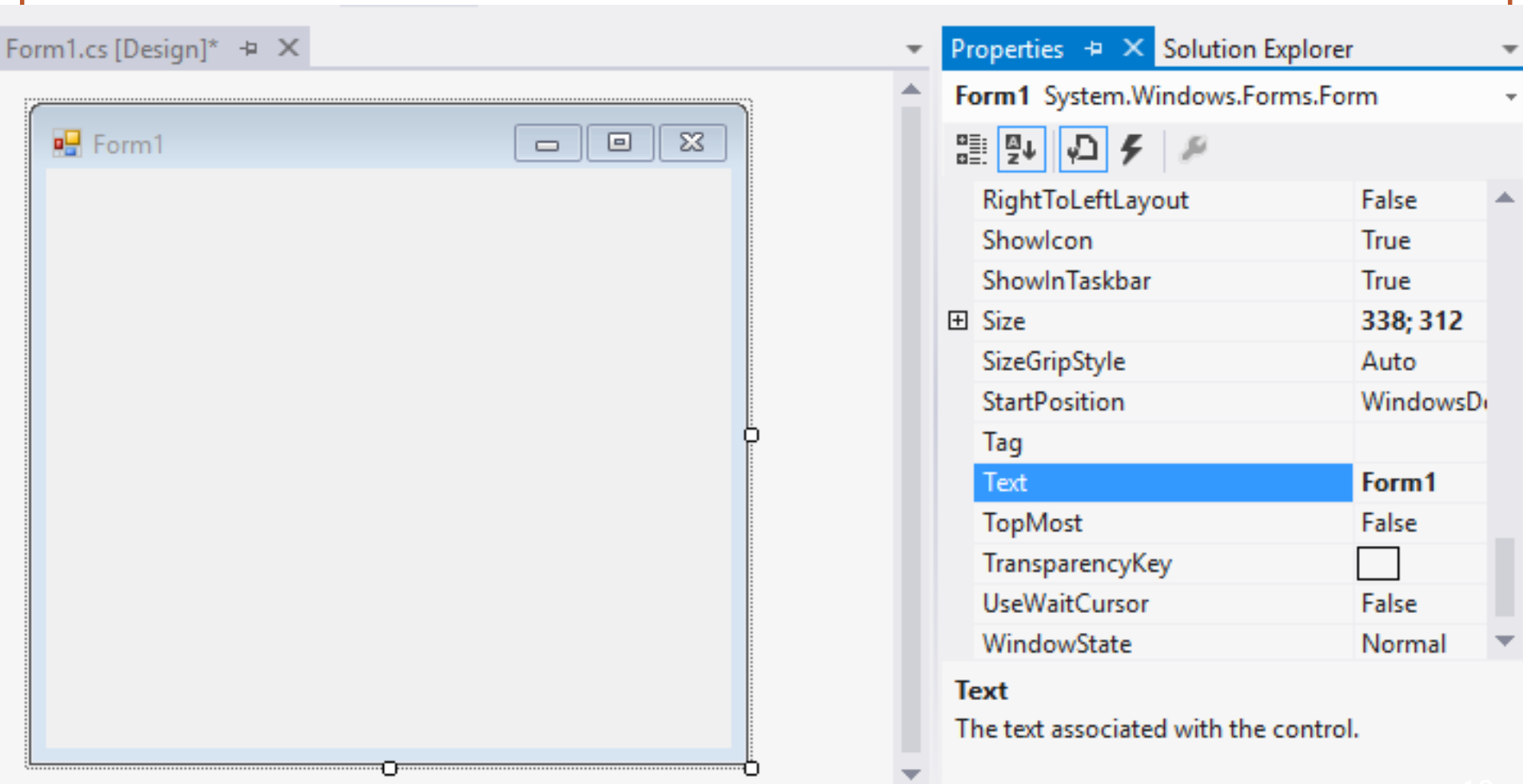


Solution Explorer

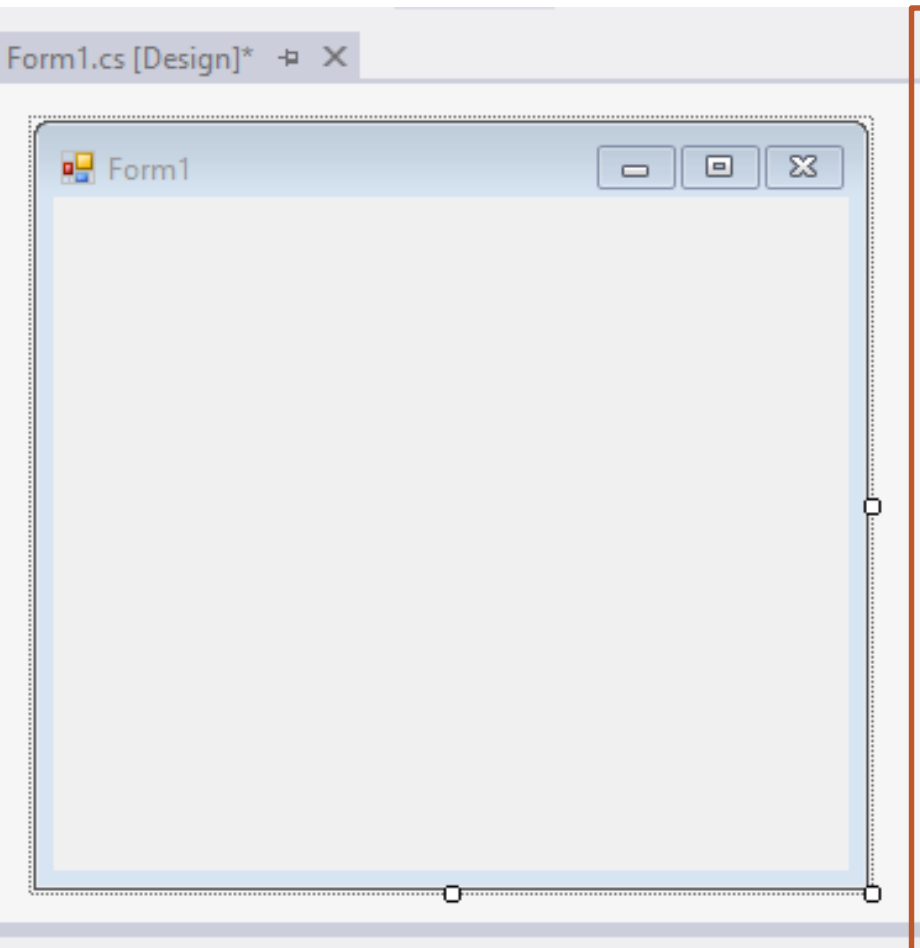


Properties Window

Used to change the properties of a selected object.



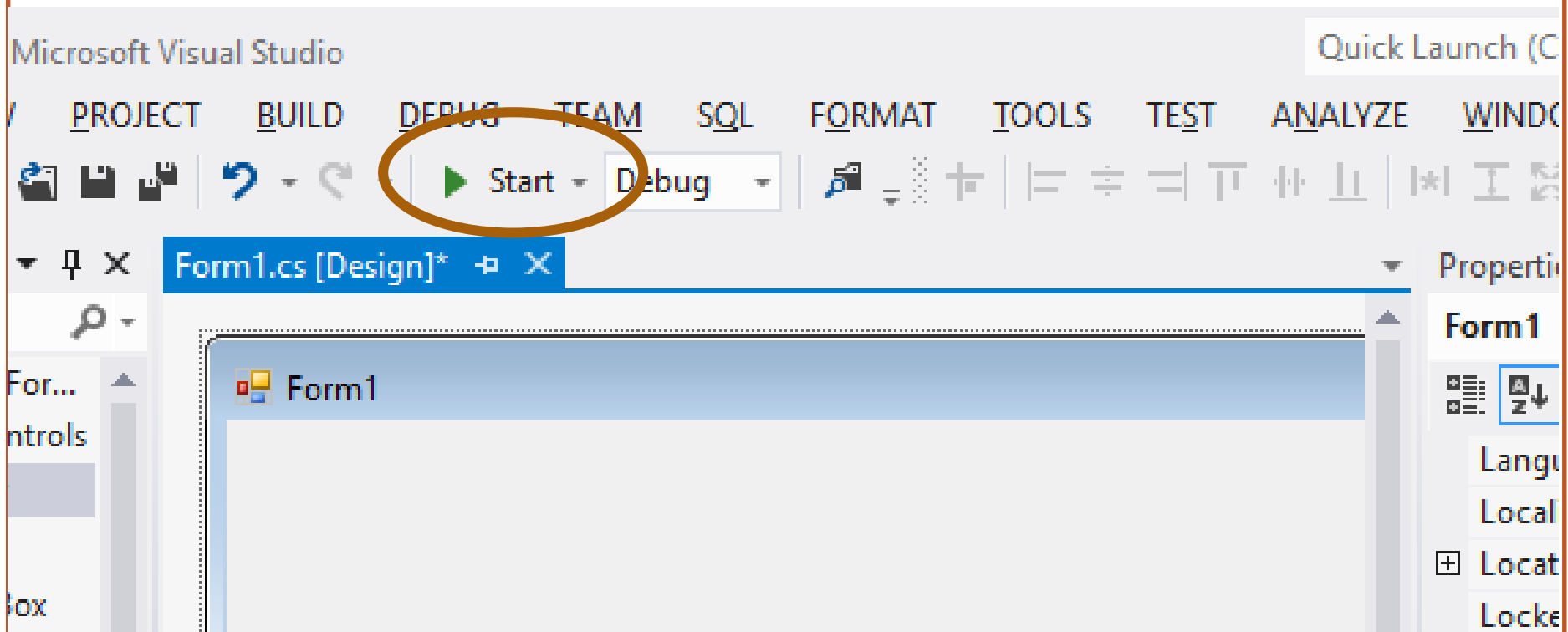
Form1 Object



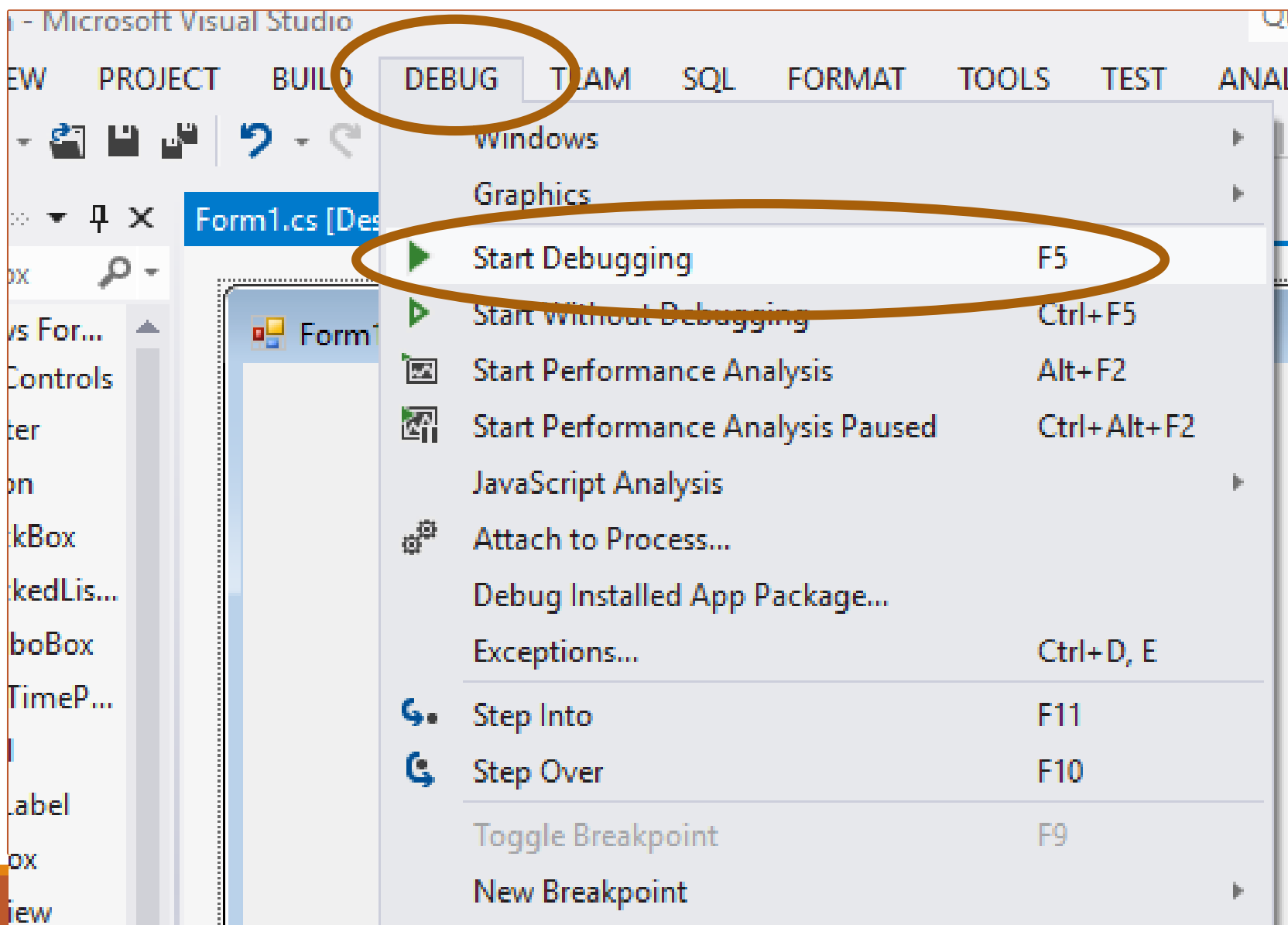
Form1 Properties

- Text
- BackColor
- Size (Width, Height)
- StartPosition
- Location (StartPosition=Manual)
- WindowState
- Opacity
- Font

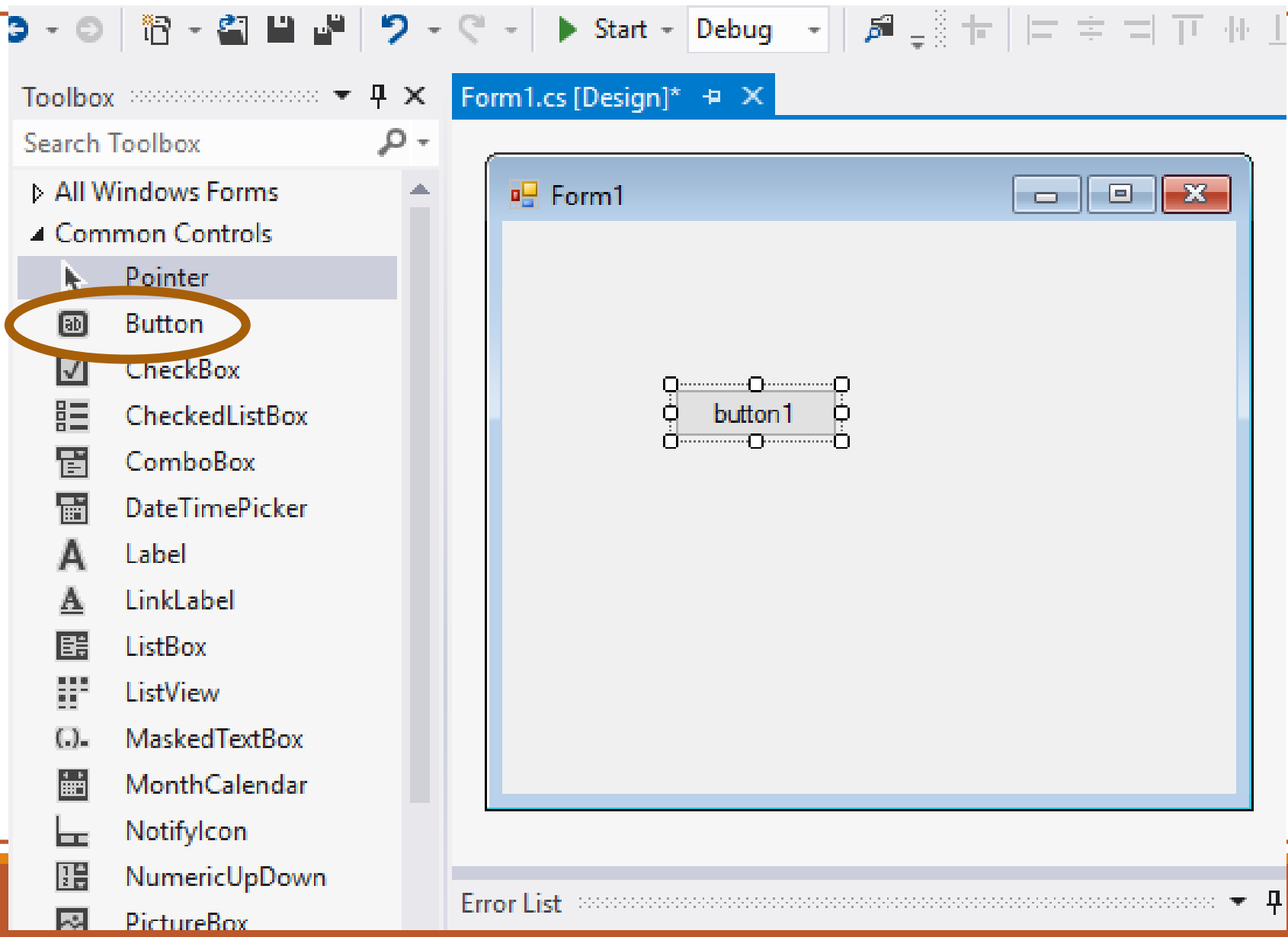
Running a Project



Running a Project

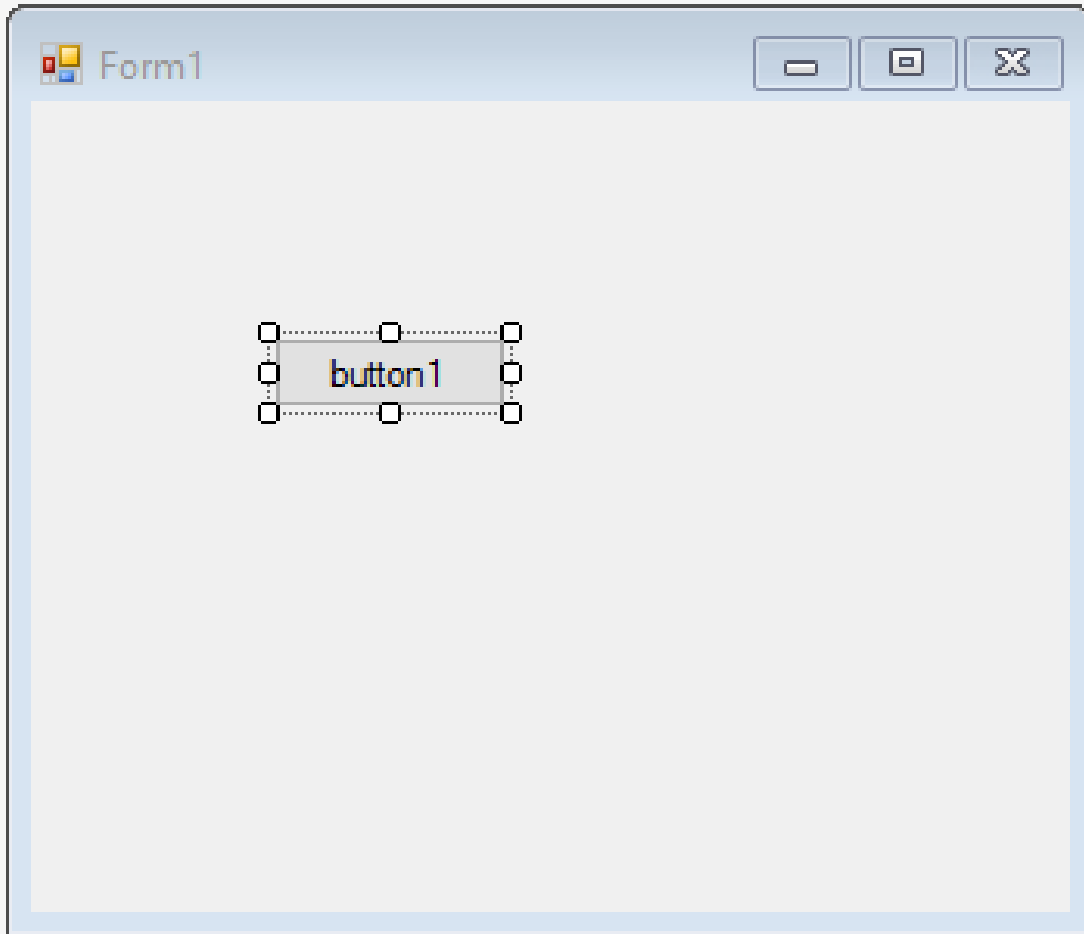


Button Object



Button Object

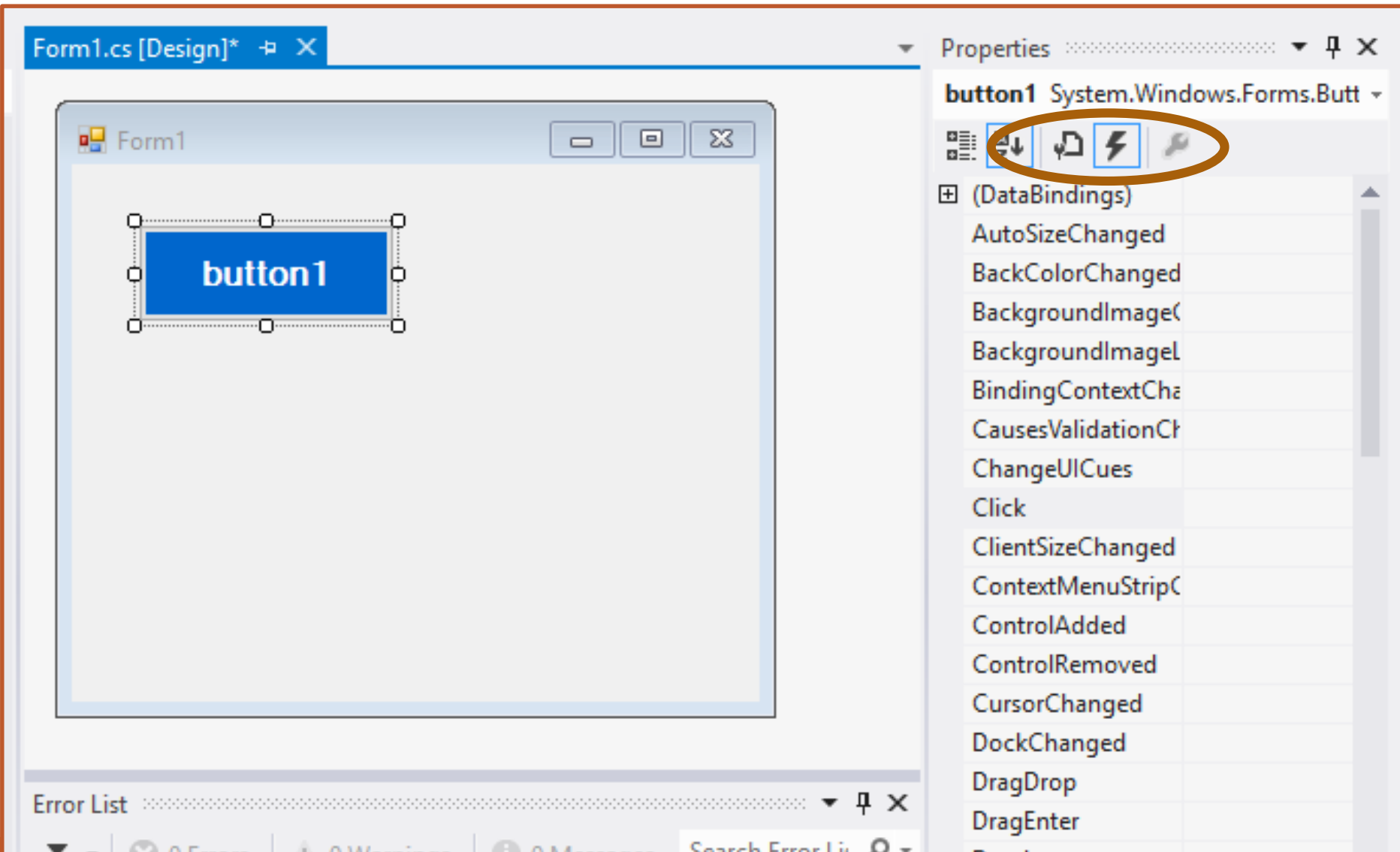
Form1.cs [Design]*



Button1 Properties

- Text
- Size
- BackColor
- ForeColor
- Font
- Location
- Enabled
- Visible

Button Object – Methods/Events



Button Object – Methods

```
Form1.cs* X Form1.cs [Design]*
firstprogram.Form1
button1_Click

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;

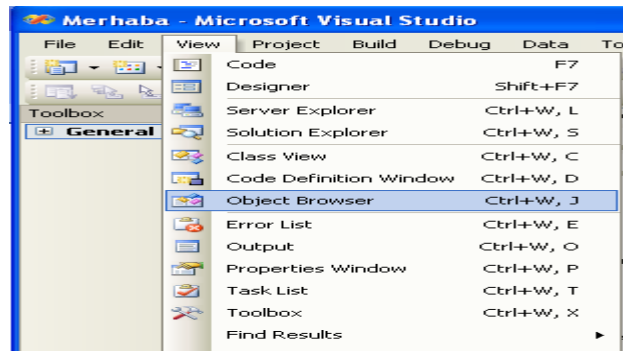
namespace firstprogram
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

Used Libraries

We should add libraries including necessary objects for our program with *using* label.

```
using System; 1* 2
using System.Collections.Generic;
using System.ComponentModel; 3
using System.Data; 4
using System.Drawing; 5
using System.Text; 6
using System.Windows.Forms; 7*
```



1-)The most general data types(integer, double, string) and libraries where classes are defined.

2-)Library where collection type interface and classes are defined.

3-)Library where behaviour and general expressions of component are defined during run-time and design phase.

4-)Library where classes of ADO.NET architecture which provides interaction with data sources.

5-)Library where basic graphic drawing classes are defined.

6-)Library where UTF8,ASCII character coding and classes helping string instructions are defined.

7-)Library where form, label, button etc which are necessary for Windows based applications are defined.

Event

- ❖ Event are frequently observed in object oriented programming languages like C#. It means an activity to happen for a particular object. For example, “Click” event of a “Button” object means clicking of a button. If we write a code for that event, this code will be run when user clicks the referred button in run-time. It will run if and only if this event occurs. i.e in this example, it does not work unless button is clicked.

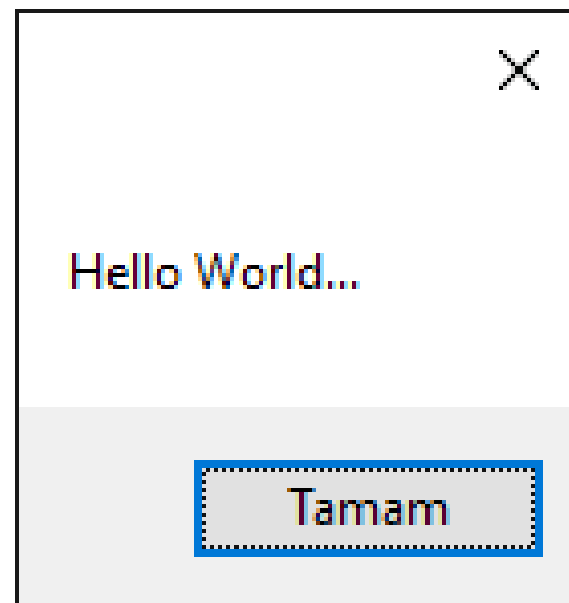
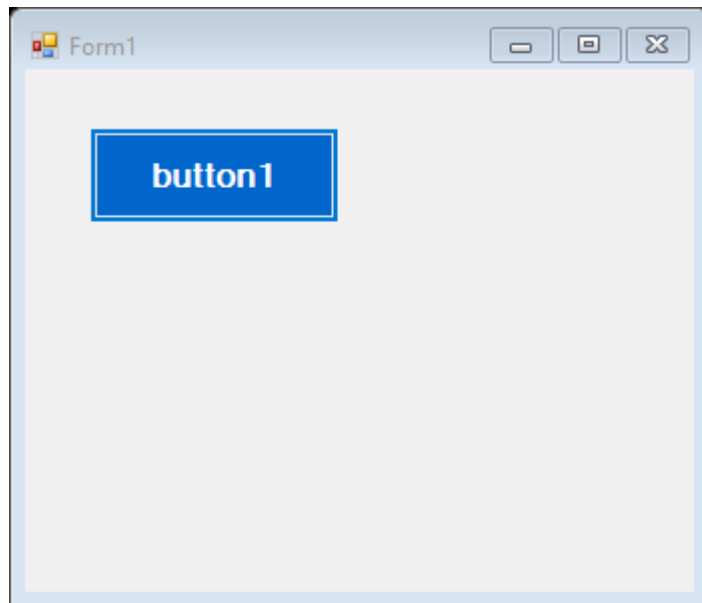
```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Denemeler
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

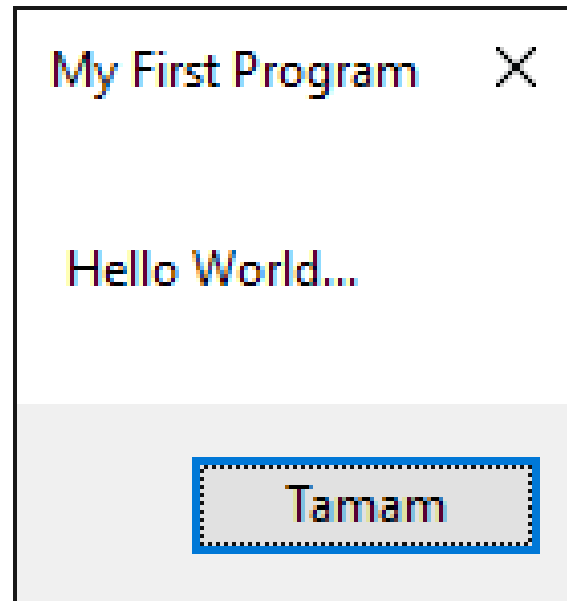
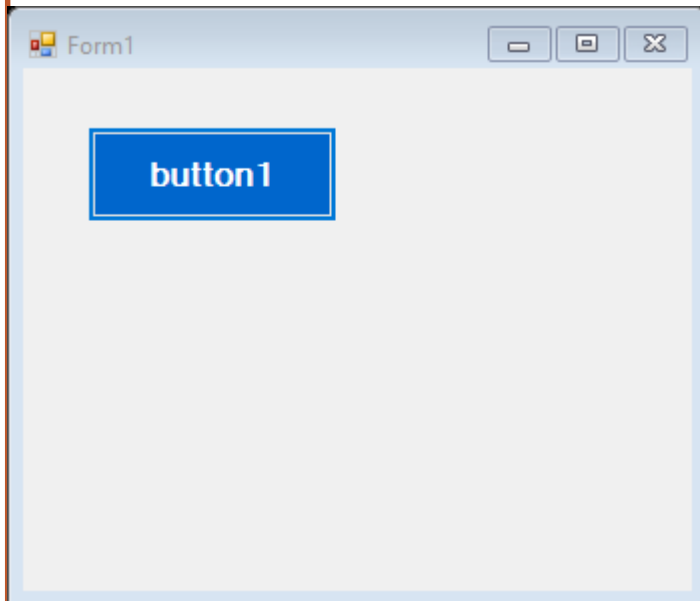
Button1 – Click Event

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello World...");
}
```



Button1 – Click Event

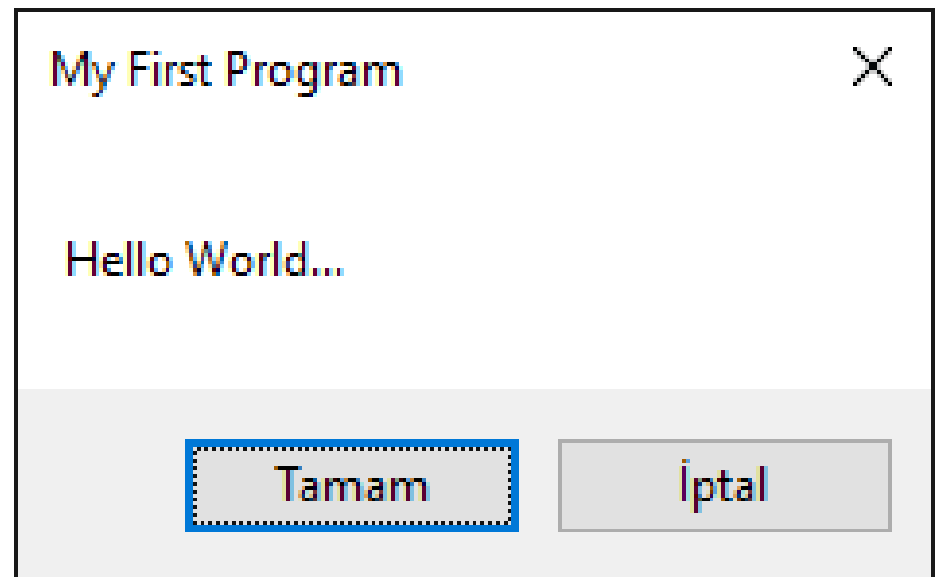
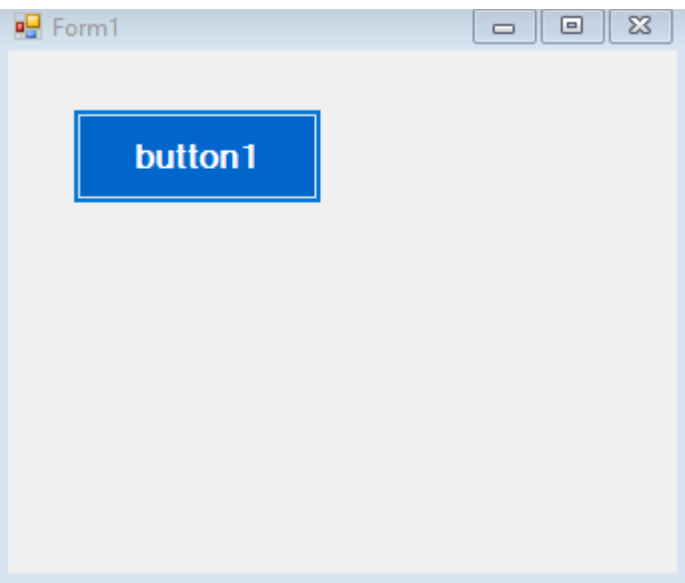
```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello World...", "My First Program");
}
```



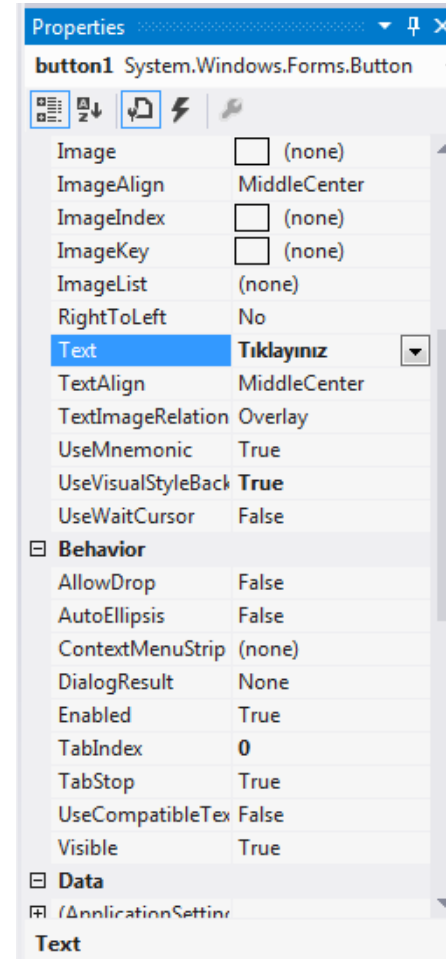
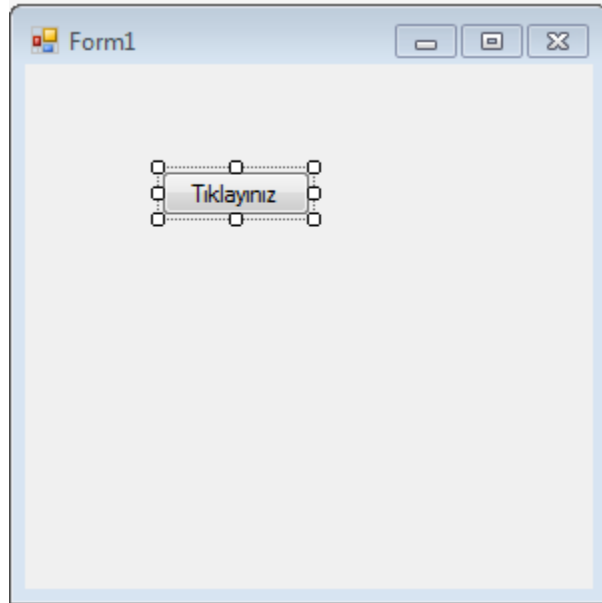
Button1 – Click Event

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello World...", "My First Program", MessageBoxButtons.);
}
```

- AbortRetryIgnore
- OK
- OKCancel
- RetryCancel
- YesNo
- YesNoCancel



Changing property of an object

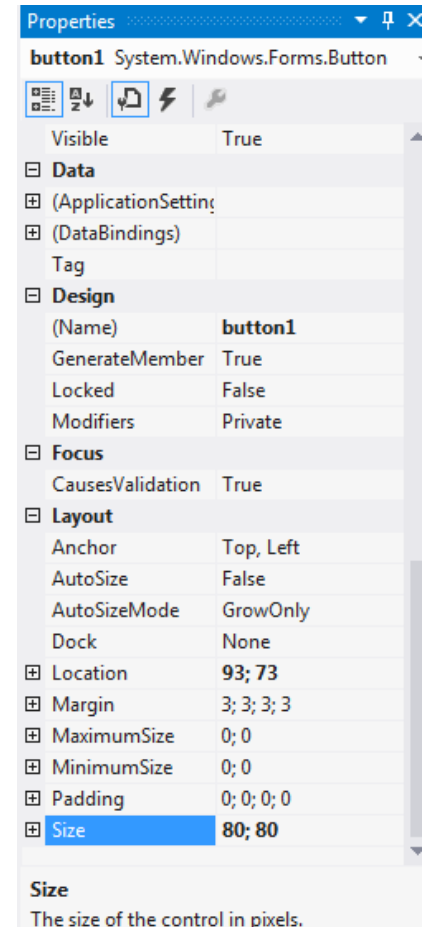
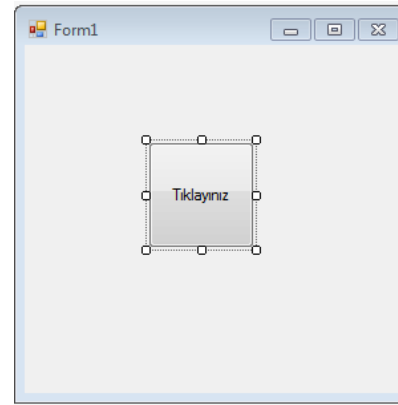


Changing property of an object

Exercises:

Write the necessary code for button1_Click to

- Change button1's width to 80
- Change button1's height to 80



Changing property of an object

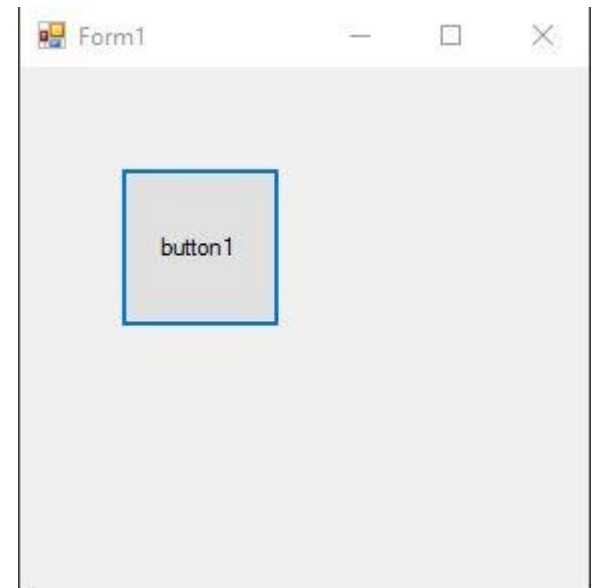
```
private void button1_Click(object sender, EventArgs e)
{
    button1.Text="Basma...";
}
```

Exercises:

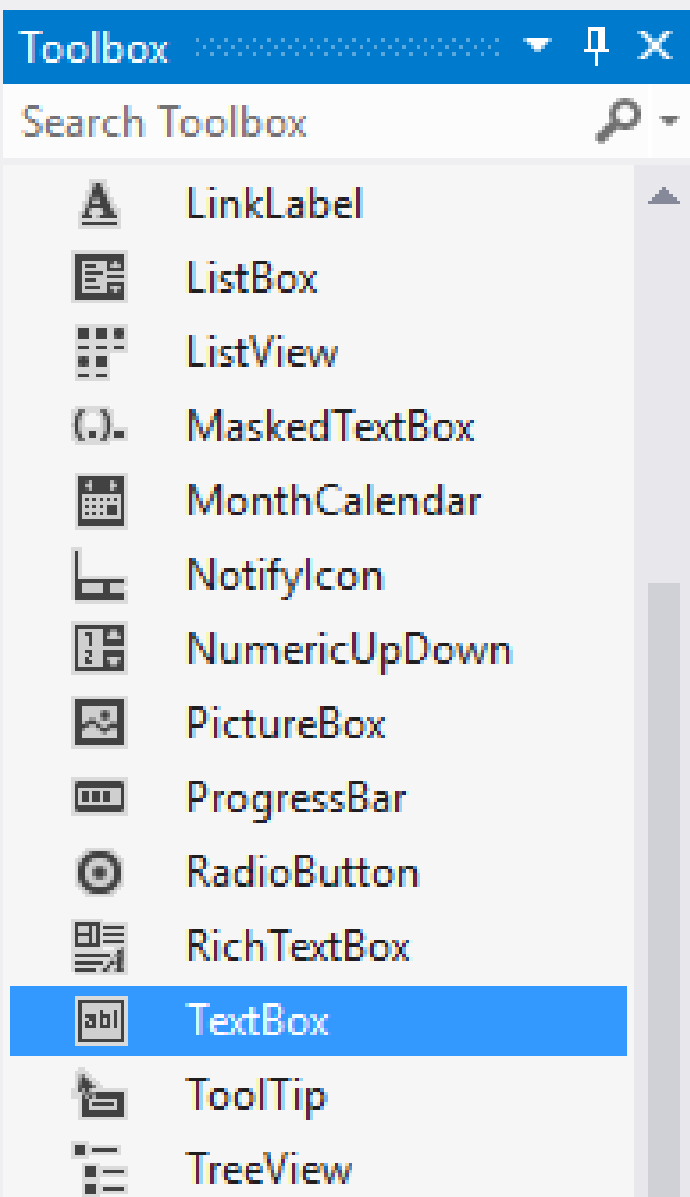
Write the necessary code for button1_Click to

- Change button1's width to 80,
- Change button1's height to 80,
- Move button1 right by 50,
- Move button1 down by 50

```
private void button1_Click(object sender, EventArgs e)
{
    button1.Width = 80;
    button1.Height = 80;
    button1.Left = +50;
    button1.Top = +50;
}
```

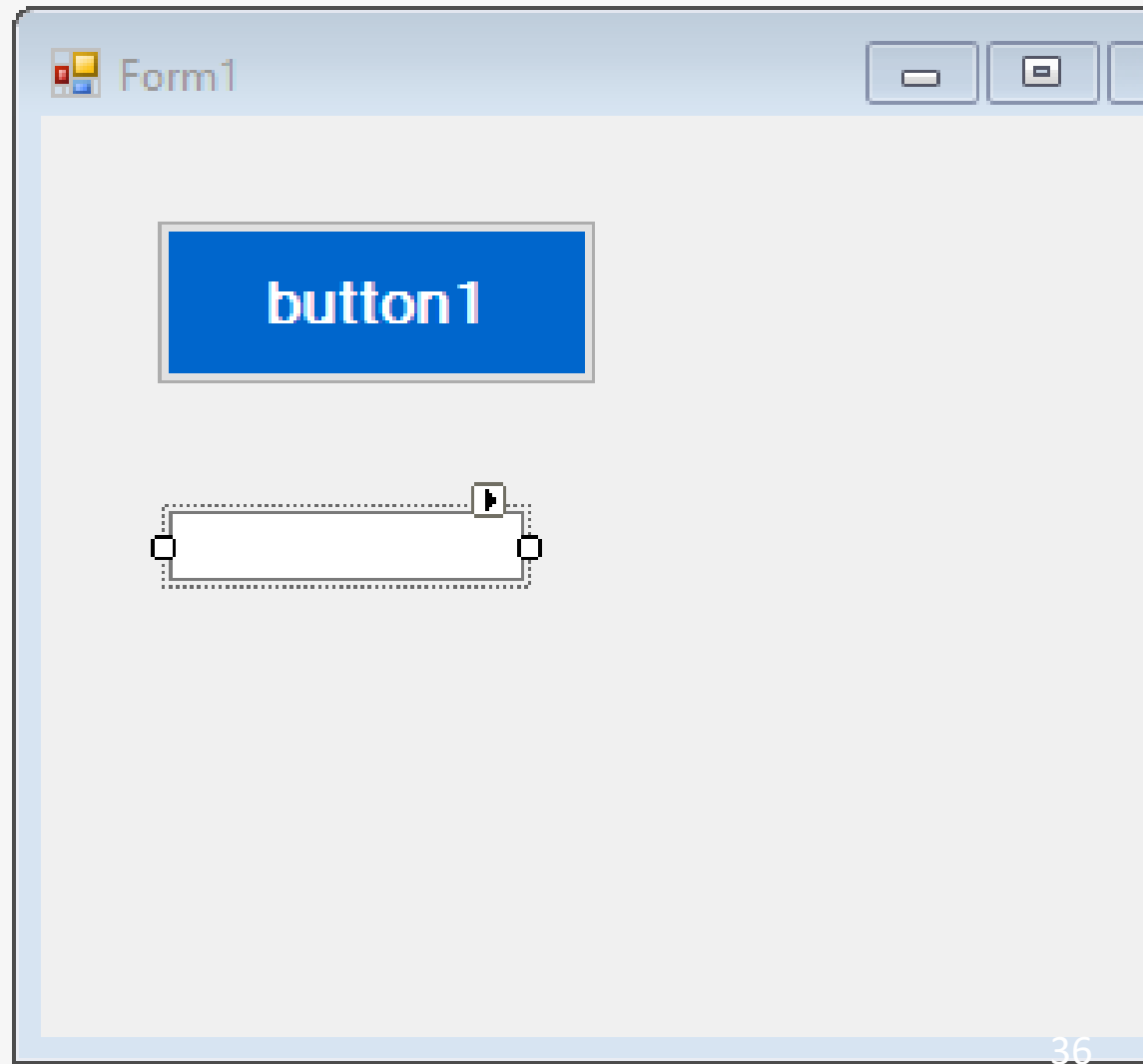


Textbox Object



Form1.cs*

Form1.cs [Design]*



Textbox Object

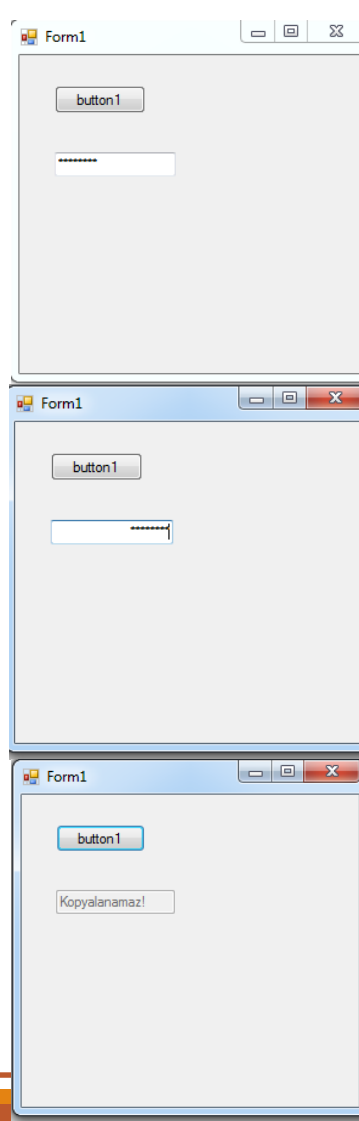
textBox1 Properties

- Text
- Font
- Size
- TextAlign
- Enabled
- ReadOnly
- PasswordChar

Textbox Object

textBox1 Properties

- Text
- Font
- Size
- TextAlign
- Enabled
- ReadOnly
- PasswordChar



Properties System.Windows.Forms.TextBox

Location	29; 80
Locked	False
Margin	3; 3; 3; 3
MaximumSize	0; 0
MaxLength	32767
MinimumSize	0; 0
Modifiers	Private
Multiline	False
PasswordChar	*
ReadOnly	False
RightToLeft	No
ScrollBars	None
ShortcutsEnabled	True
Size	100; 20
TabIndex	1
TabStop	True
Tag	
Text	
TextAlign	Right
UseSystemPassword	False
UseWaitCursor	False
Visible	True
WordWrap	True

TextAlign
Indicates how the text should be aligned for edit controls.

Properties System.Windows.Forms.TextBox

Enabled	True
Font	Microsoft Sans Serif;
ForeColor	WindowText
GenerateMember	True
HideSelection	True
ImeMode	NoControl
Lines	String[] Array
Location	29; 80
Locked	False
Margin	3; 3; 3; 3
MaximumSize	0; 0
MaxLength	32767
MinimumSize	0; 0
Modifiers	Private
Multiline	False
PasswordChar	*
ReadOnly	False
RightToLeft	No
ScrollBars	None
ShortcutsEnabled	True
Size	100; 20
TabIndex	1
TabStop	True
Tag	

PasswordChar
Indicates the character to display for password input for single-line edit controls.

Label Object

Toolbox

Search Toolbox

▶ All Windows Forms

▲ Common Controls

Pointer

Button

CheckBox

CheckedListBox

ComboBox

DateTimePicker

A Label

LinkLabel

ListBox

ListView

MaskedTextBox

MonthCalendar

Form1.cs*

Form1.cs [Design]

Form1

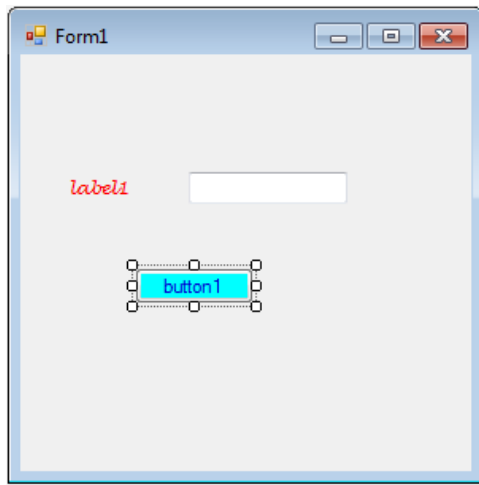
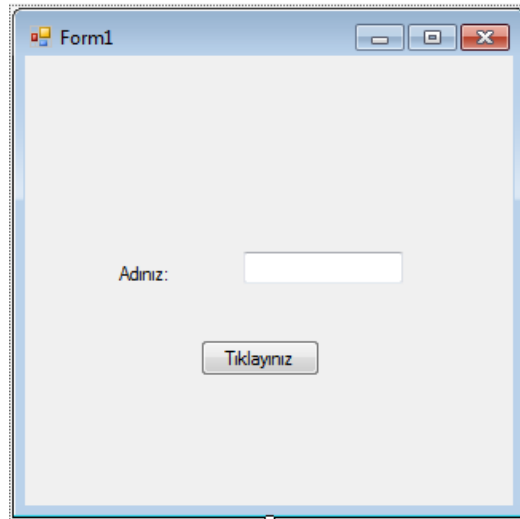
button1

label1

label1 Properties

- Text
- Font
- Enabled

Label Object



Properties

button1 System.Windows.Forms.Button

AccessibleName
AccessibleRole Default

Appearance

BackColor Cyan
BackgroundImage (none)
BackgroundImage Tile
Cursor Default

FlatAppearance

FlatStyle Standard

Font Microsoft Sans Serif;

ForeColor MediumBlue
Image (none)
ImageAlign MiddleCenter
ImageIndex (none)
ImageKey (none)
ImageList (none)
RightToLeft No
Text **button1**
TextAlign MiddleCenter
TextImageRelation Overlay
UseMnemonic True
UseVisualStyleBack False
UseWaitCursor False

Behavior

ForeColor
The foreground color of this component, which is used to display text.

Properties

label1 System.Windows.Forms.Label

Cursor Default
FlatStyle Standard

Font Microsoft Sans Serif;

ForeColor ControlText
Image (none)
ImageAlign MiddleCenter
ImageIndex (none)
ImageKey (none)
ImageList (none)
RightToLeft No

Text Adınız:

TextAlign TopLeft
UseMnemonic True
UseWaitCursor False

Behavior

AllowDrop False
AutoEllipsis False
ContextMenuStrip (none)
Enabled True
TabIndex 2
UseCompatibleTex False
Visible True

Data

(ApplicationSetting)

Text
The text associated with the control.

Exercises

www.csc.com.tr

Lütfen Giriş Yapınız

İsim:

Soyad:

Giriş

www.csc.com.tr

Lütfen Giriş Yapınız

İsim:

Soyad:

Giriş

Merhaba Ali Yılmaz

Tamam

Exercises

```
private void button1_Click(object sender, EventArgs e)
{
    textBox3.Text = textBox1.Text + textBox2.Text;
}
```

Exercises

Write the code for button1_Click to

- Put your name in textBox1,
- Change label1's text with the string written in textBox1
- Show the string written in textBox1 in a MessageBox window
- Change the caption of Form1 with the string written in textBox1

Exercises



First Program in C#



Basic Calculator

1st Number

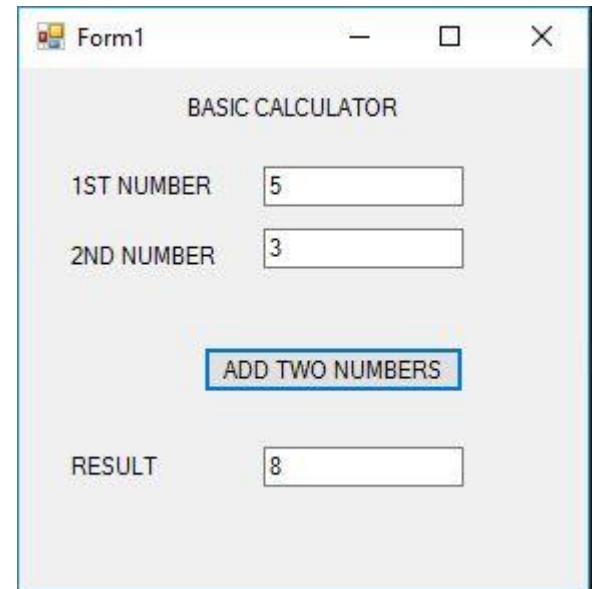
2nd Number

Add Two Numbers

Result

Exercises

```
private void button1_Click(object sender, EventArgs e)
{
    double x = Convert.ToDouble(textBox1.Text);
    double y = Convert.ToDouble (textBox2.Text);
    textBox3.Text = (x + y).ToString();
}
```



The screenshot shows a Windows application window titled "Form1". Inside the window is a form titled "BASIC CALCULATOR". The form contains three text boxes and one button. The first text box, labeled "1ST NUMBER", contains the value "5". The second text box, labeled "2ND NUMBER", contains the value "3". A button labeled "ADD TWO NUMBERS" is located below these two boxes. The third text box, labeled "RESULT", contains the value "8".

Exercises

```
private void button1_Click(object sender, EventArgs e)
{
    textBox3.Text = Convert.ToString(
        Convert.ToInt16(textBox1.Text) + Convert.ToInt16(textBox2.Text));
}
```