



GÖRSEL PROGRAMLAMA

Dr. Öğr. Üyesi Durmuş ÖZDEMİR
durmus.ozdemir@dpu.edu.tr

Hafta 1

Konu Başlıklarları

Hafta 1	Visual Studio 20XX IDE (Integrated Development Environment/Tümleşik Geliştirme Ortamı), Yeni Bir Proje Oluşturma
Hafta 2	Değişkenler ve Operatörler
Hafta 3	Tür Dönüşümleri ve Akış Kontrolleri
Hafta 4	Döngüler
Hafta 5	Form Kontrol Elemanları
Hafta 6	İletişim Kutuları
Hafta 7	Diziler
Hafta 8	Ara Sınav
Hafta 9	Hata Yakalama
Hafta 10	Çoklu Formlar
Hafta 11	Hazır Fonksiyonlar
Hafta 12	Resim İşlemleri
Hafta 13	Temel Veritabanı İşlemleri - I
Hafta 14	Temel Veritabanı İşlemleri - II

Not: Yukarıda belirtilen konu başlıklarının işleniş sıralaması değişkenlik gösterebilir.

Değerlendirme

Türü	Yıl Sonuna Etkisi
Vize	% 25
Proje	% 25
Ödev + Uygulamalar + Quiz	% 10
Final	% 40

Not: Değerlendirme katsayıları belirtilen aralıklar dersin işleniş durumuna ve ödevlerin ağırlık durumuna göre değişkenlik gösterebilir.

Kaynakça ve Ders Takibi

Kaynaklar	<p>Teach Yourself Visual C# 2008 in 24 Hours, James Foxall, Sams Publishing, 2008.</p> <p>Beginning Microsoft Visual C# 2008, Karli Watson, Christian Nagel et al., Wiley Publishing, 2008.</p> <p>DEITEL C# How to Program - PDFs</p> <p>** Ayrıca Ders notları ve dağıtılan diğer notlar ve dökümanlar.</p>
Ders Takibi	Dersi alan öğrenciler oys.dpu.edu.tr adresi üzerinden verilecek ödevleri ve uygulamaları takip etmek durumundadırlar.

IDE Nedir?

Visual Studio Microsoft firmasının program geliştirme ortamıdır.

(Integrated Development Environment (IDE))

Tümleşik Geliştirme Ortamı

- Program yazma,
- Derleme,
- Çalıştırma,
- Hata ayıklama

KURULUM

- MS Visual Studio 20xx Kurulum ve Tanıtımı
- <https://www.microsoft.com/en-cy/imagine>
- ogr.dpu.edu.tr adresiniz ile ücretsiz, Microsoft Imagine eski adıyla dreamspark
- Yeni Proje Oluşturma



Install Microsoft Visual Studio 2010

Install Microsoft Visual Studio 2010 features and required components.

1



Check for Service Releases

Check for the latest Service Releases to ensure optimal functionality of Microsoft Visual Studio 2010.

[View ReadMe](#)[Exit](#)



Microsoft®

Visual Studio 2010 Ultimate

Setup

Welcome to the Microsoft Visual Studio 2010 Ultimate installation wizard.

This wizard guides you through installing this program
and all required components.

Help Improve Setup

You can submit information about your Visual Studio
setup experiences to Microsoft. To participate, check the
box below.

- Yes, send information about my setup experiences to Microsoft
Corporation.

For more information, read the [Privacy Statement](#)

Loading completed. Click Next to continue.

2

< Previous

Next >

Cancel



Microsoft®

Visual Studio®

2010 Ultimate

Setup



Please exit all applications before continuing with the installation.

- Setup has detected that the following required components are already installed:

- Microsoft Application Error Reporting

- Setup will install the following components:

- VC 9.0 Runtime (x86)
- VC 10.0 Runtime (x86)
- Microsoft .NET Framework 4
- Microsoft Visual Studio 2010 Ultimate

To install, you must accept the license terms and enter a valid 25-character product key, where prompted.

Be sure to carefully read and understand all the rights and restrictions described in the license terms. You must accept the license terms before you can install the software.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO 2010 ULTIMATE AND TRIAL EDITION

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft updates,

Print...

Press the Page Down key to see more text.

I have read and accept the license terms.

3

I do not accept the license terms.

4

< Previous

Next >

Cancel



Microsoft Visual Studio 2010 Ultimate Setup

Select features to install:

 Full

Complete Visual Studio installation. Install all programming languages and tools.

5 **Custom**

Select which programming languages and tools to install on the next page.

Özelleştirmek isteyen
Custom seçeneğini seçerek
sadece yüklemek istemediği
dilleri seçebilir.

Feature description:

Installs the Visual Studio 2010 Ultimate integrated environment together with modeling, development, testing, and deployment components that can simplify the entire development process and help ensure high-quality solutions. Provides tools for building solutions on Windows, the Web, Azure, the Office system, SharePoint, SQL Server, and other platforms by using Visual Basic, Visual C#, Visual C++, or Visual F#.

Includes these advanced features: Test Impact Analysis,

Product install path:

C:\Program Files\Microsoft Visual Studio 10.0\

[Browse...](#)

Disk space requirements:

Volume	Disk Size	Available	Required	Remaining
C:	64.0 GB	55.4 GB	7.3 GB	48.1 GB

6[< Previous](#)**Install**[Cancel](#)



Installing Components:

- VC 9.0 Runtime (x86)**
 - || VC 10.0 Runtime (x86)
 - || Microsoft .NET Framework 4
 - || Microsoft Windows Installer 4.5 Update (x86) - Windows XP
 - || Microsoft Visual F# 2.0 Runtime
 - || Microsoft Visual Studio Macro Tools
 - || TFS Object Model (x86)
 - || .NET Framework 4 Multi-Targeting Pack
 - || Microsoft Visual Studio 2010 Ultimate
 - || Microsoft Web Deployment Tool (x86)
 - || Microsoft ASP.NET MVC 2 - Visual Studio 2010 Tools
 - || Microsoft ASP.NET MVC 2
 - || Microsoft Silverlight
 - || Microsoft Silverlight 3 SDK
 - || Microsoft Visual Studio 2010 Tools for Office Runtime (x86)
 - || Microsoft Office Developer Tools (x86)

Bu ekranda bilgisayarınızın hızına göre ve bazı özelliklerin önceden yüklenmiş olup olmadığına göre 10-15 dk. bekleyebilirsiniz.

Installing VC 9.0 Runtime (x86)...

A horizontal progress bar with a blue segment indicating the progress of the installation.

< Previous

Next >

Cancel



Installing Components:

The screenshot shows the 'Installing Components:' section of the setup wizard. On the left, a tree view lists several components checked for installation, including 'VC 9.0 Runtime (x86)', 'VC 10.0 Runtime (x86)', 'Microsoft .NET Framework 4', and 'Microsoft Windows Installer 4.5 Update (x86) - Windows XP'. A red arrow points to the Windows Installer update item. On the right, a modal dialog box titled 'Microsoft Visual Studio 2010 Ultimate Setup' contains the message: 'You must restart your computer to complete the installation. Setup will automatically continue after your computer has restarted.' Below the message are two buttons: 'Restart Now' and 'Cancel'. A red box highlights the 'Restart Now' button, and another red box highlights the explanatory text above it.

Installing Microsoft Windows Installer 4.5 Update (x86) - Windows XP...

< Previous

Next >

Cancel



Setup

Success

Visual Studio 2010 setup is complete.

- Setup completed, however, some warnings were detected.
Review the [Error Log](#).

It is also recommended that you choose to opt in to the [Microsoft Update](#) service to make sure you receive all the available updates for this and other Microsoft products.

- Read the [security notes](#)
- View the [readme notes](#)
- Examine the [installation log](#)

Fill up your tool box.

Install extensions from the [Visual Studio Gallery](#) that integrate with Visual Studio to further enhance your development productivity.

Help is just a click away!

Clicking the button below will launch the Help Library Manager which will guide you through configuring local help for Visual Studio. [More info...](#)

[Install Documentation](#)

Yardım yüklemek istiyorsanız Finish demeden önce buraya tıklayın. (Not: Yardım ingilizce)

7

< Previous

Finish

Cancel

Help Library Manager - Microsoft Help Viewer 1.0

Settings | Help

Install Content from Disk

	Actions	Status	Si
▼ .NET Development			
.NET Framework 4	Add		68
▼ Visual Studio 2010			
Application Lifecycle Management	Add		36
Extending Application Lifecycle Management	Add		69
JScript	Add		15
Office Development	Add		17
Sharepoint Development	Add		15
Team Foundation	Add		42

Click Add to select content for offline help.

Update Cancel



Install Content from Disk

Extending Application Lifecycle Management	Add	65
JScript	Add	15
Office Development	Add	17
Sharepoint Development	Add	15
Team Foundation	Add	42
Visual Basic	Add	16
Visual C#	Add	16
Visual C++	Add	15
Visual F#	Add	15
Visual Studio (All)	Add	19
Visual Studio Fundamentals	Add	15

Click Add to select content for offline help.



Install Content from Disk

Office Development	Add	17
Sharepoint Development	Add	15
Team Foundation	Add	42
Visual Basic	Add	16
Visual C#	Cancel	Update Pending
Visual C++	Add	15
Visual F#	Add	15
Visual Studio (All)	Cancel	Update Pending
Visual Studio Fundamentals	Add	15
Win32 and COM Development	Add	36

Estimated download size: 755 MB

Update

Cancel



Updating Local Library



Copying packages 8 of 35

Hide

Cancel



Updating Local Library

Finished updating

Finish



Change or Remove Microsoft Visual Studio 2010

Repair, reinstall, or install additional Microsoft Visual Studio 2010 features. You can also uninstall Microsoft Visual Studio 2010.



Check for Service Releases

Check for the latest Service Releases to ensure optimal functionality of Microsoft Visual Studio 2010.

Güncellemeleri kontrol etmek isterseniz
buraya tıklayın.

View ReadMe

Exit

Choose Default Environment Settings



Before you begin using the application for the first time, you need to specify the type of development activity you engage in the most, such as Visual Basic or Visual C#. This information is used to apply a predefined collection of settings to the development environment that is designed for your development activity.

You can choose to use a different collection of settings at any time. From the Tools menu, choose Import and Export Settings and then choose Reset all settings.

Choose your default environment settings:

General Development Settings

Project Management Settings

Visual Basic Development Settings

Visual C# Development Settings

Visual C++ Development Settings

Visual F# Development Settings

Web Development

Web Development (Code Only)

Description:

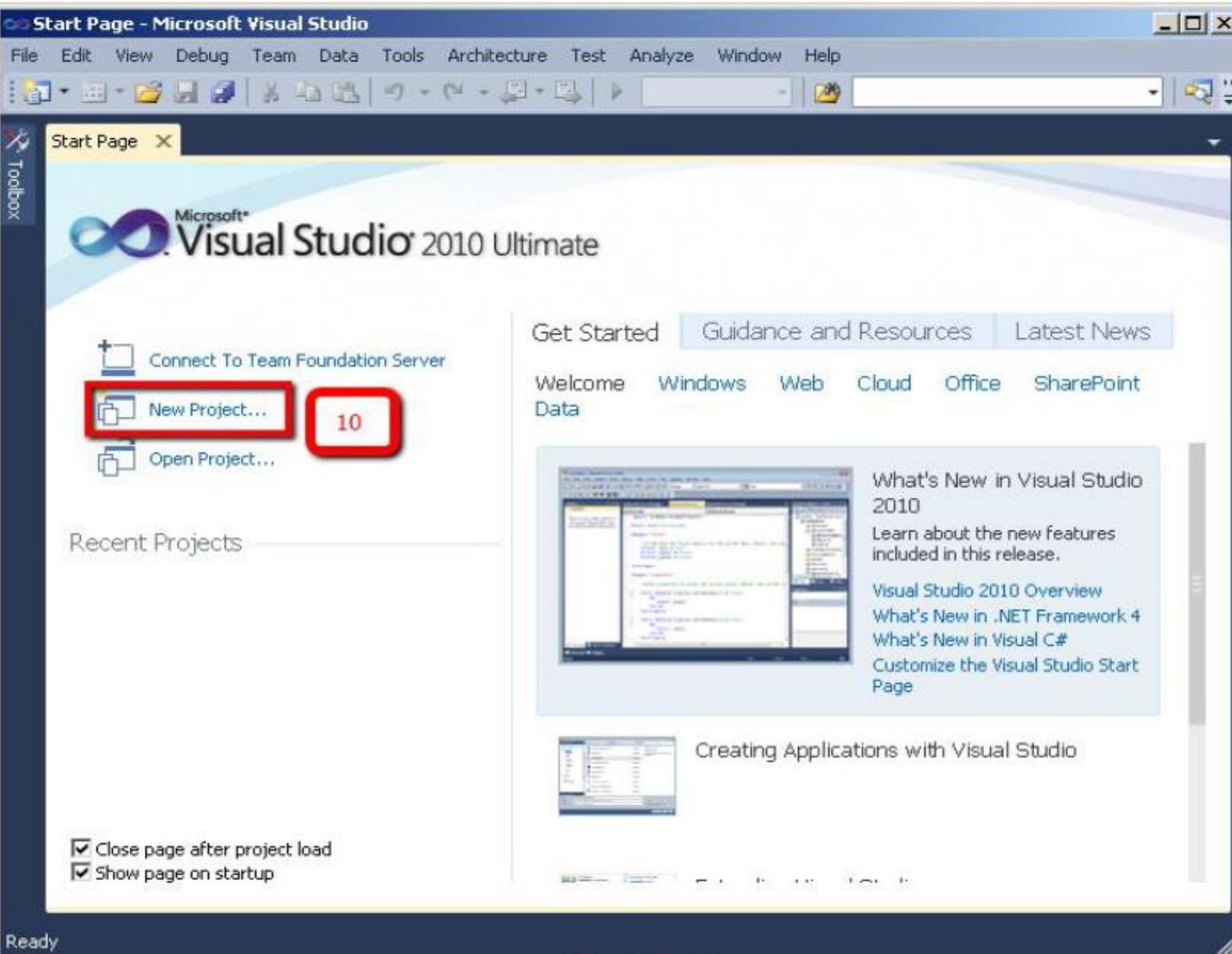
Customizes the environment to maximize code editor screen space and improve the visibility of commands specific to C#. Increases productivity with keyboard shortcuts that are designed to be easy to learn and use.

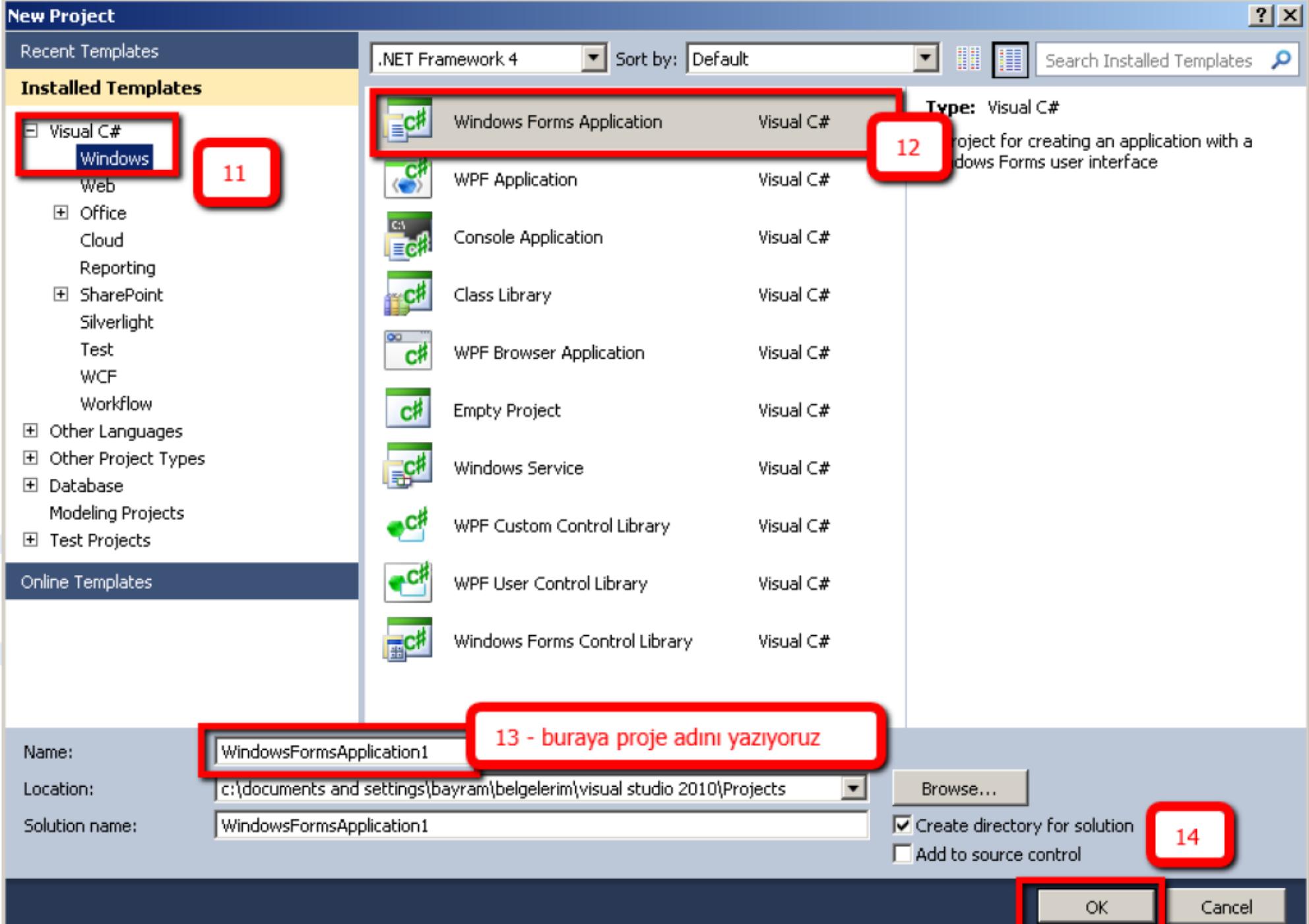
8

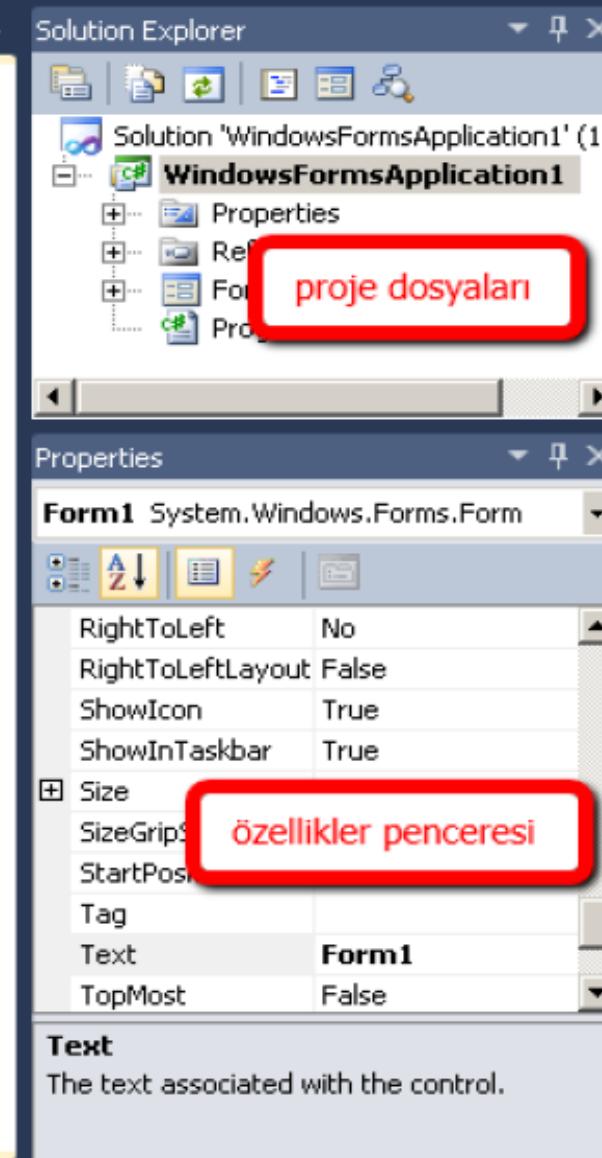
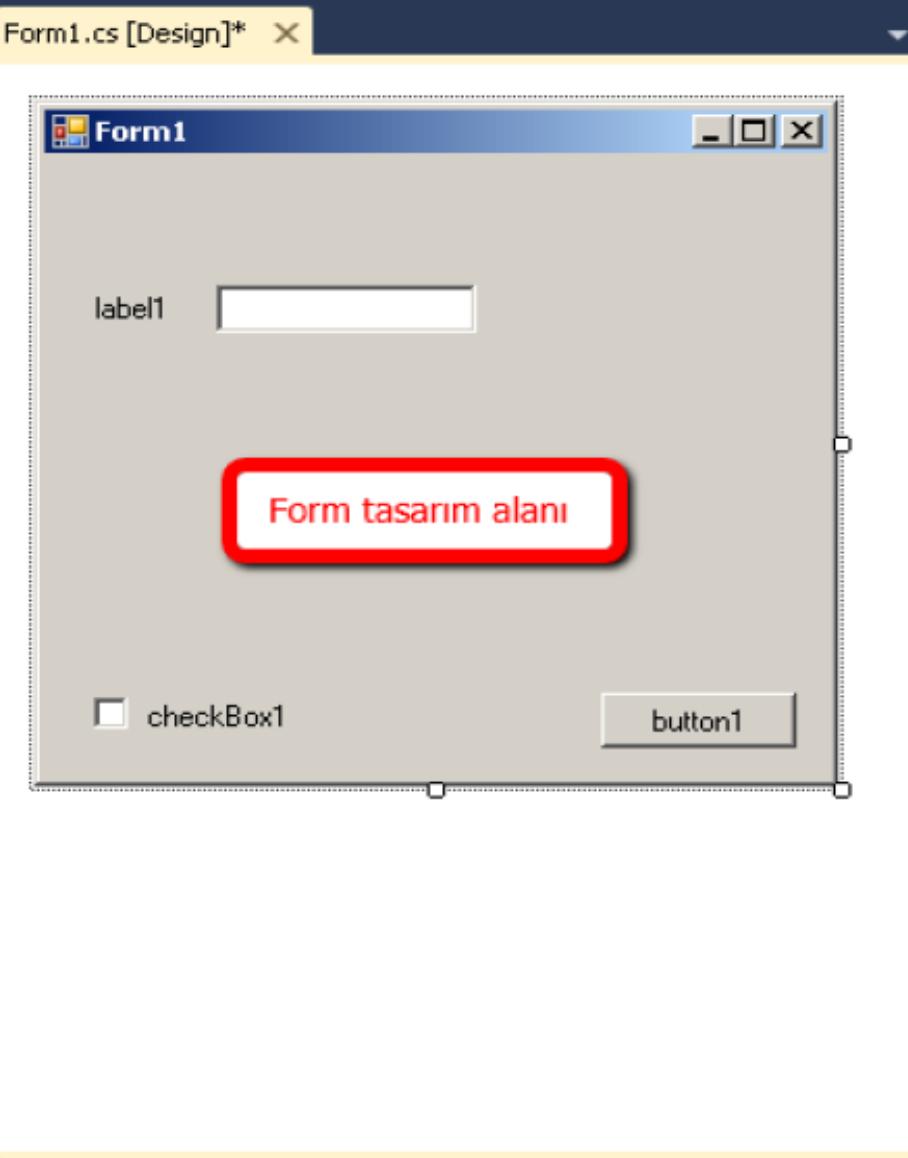
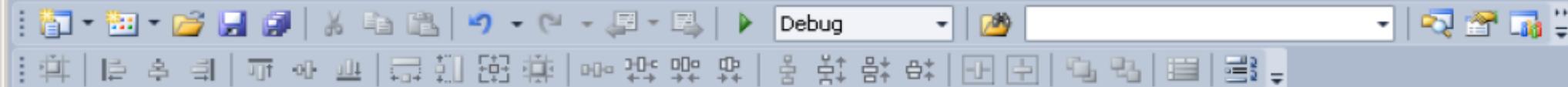
9

Start Visual Studio

Exit Visual Studio

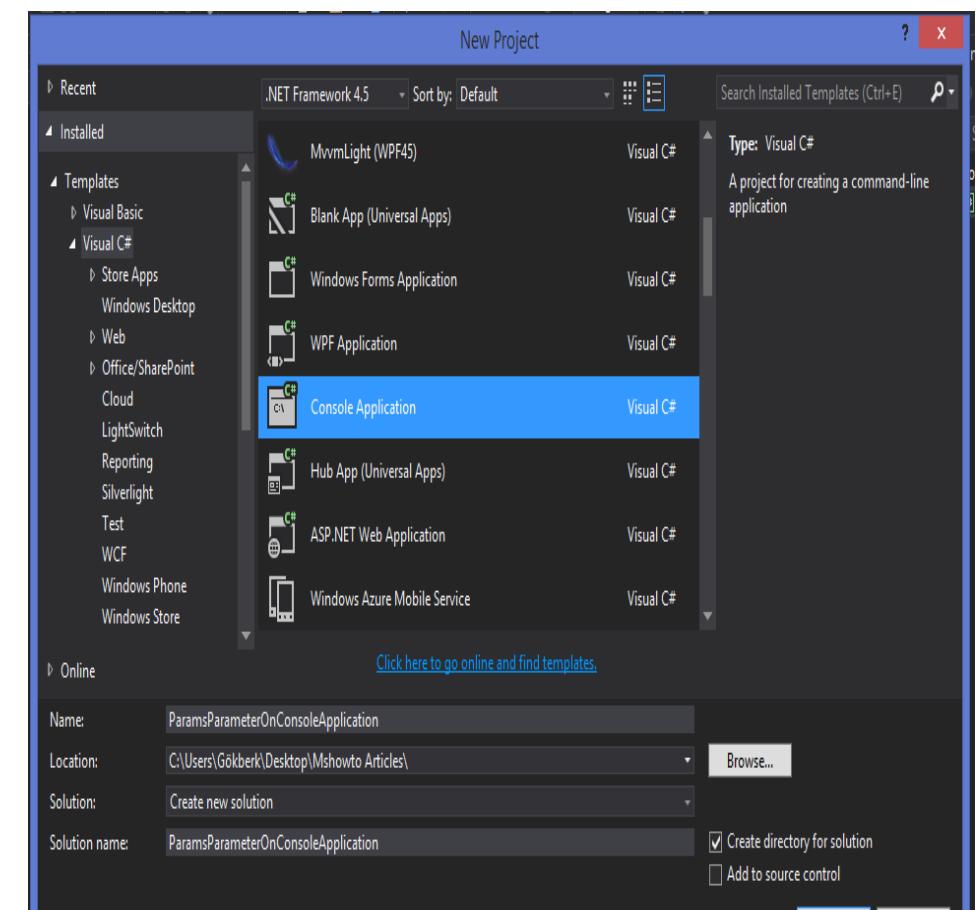






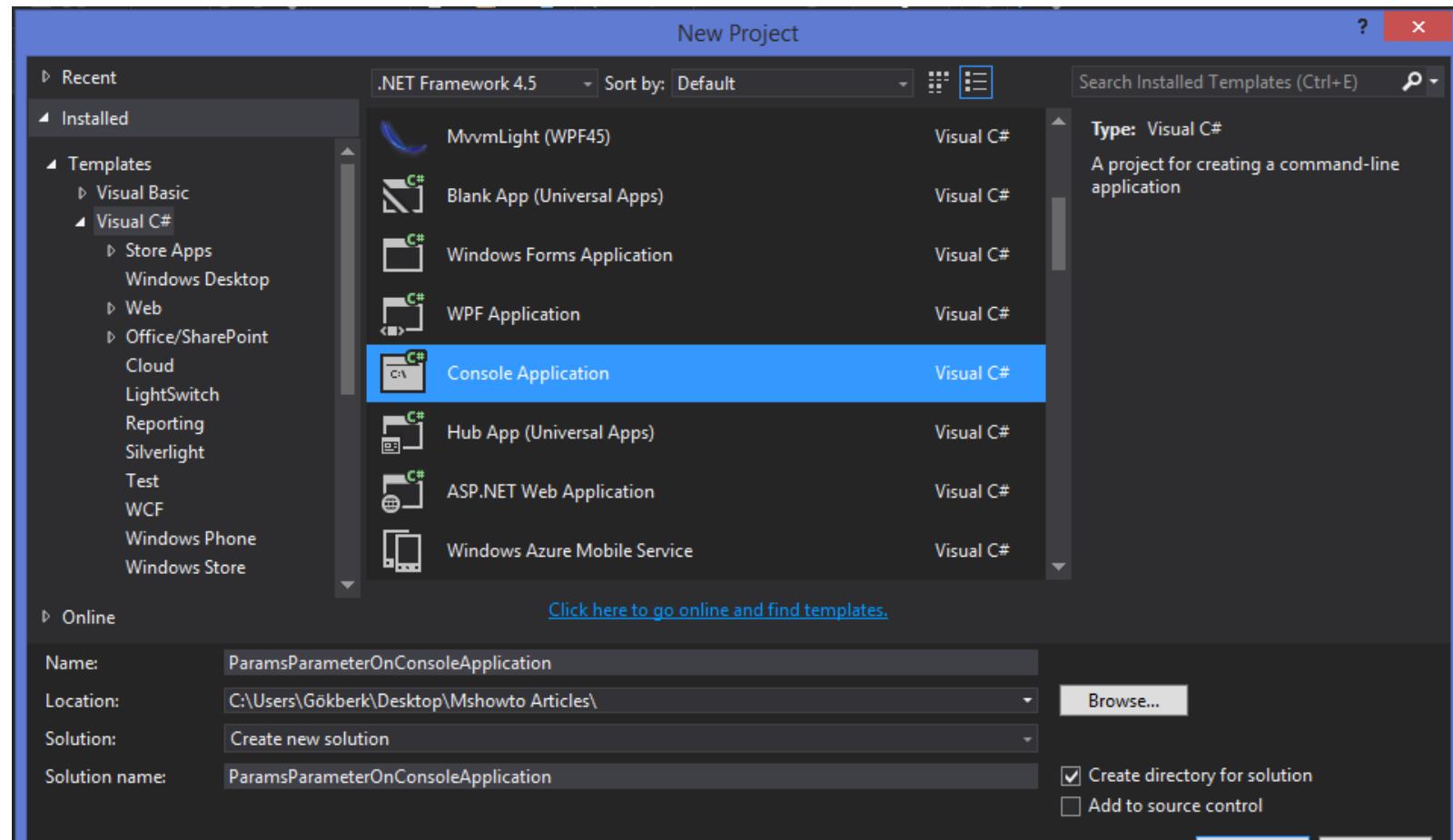
New Project Ekranı Templates

Templates	Açıklama
Windows Form Application	Standart Windows tabanlı uygulamaları oluşturmak için kullanılır.
WPF Application	Yeni nesil multimedia destekli Windows form uygulamaları için kullanılır
Console Application	Komut satırında çalışacak uygulamalar için kullanılır
Class Library	Diger uygulamaların işletim sistemi ile ilgili irtibatının sağlanmasında oluşturulacak DLL oluşturmak için kullanılır.
ASP.NET Web Application	Web uygulamaları
Silverlight Application	Grafiksel olarak çok zengin içerikli web uygulamaları



Konsol Uygulaması? // Windows Form Application ?

Program giriş ve çıkışları bir konsol (komut) penceresi üzerinden gerçekleşir.



Ancak bu ders
haricinde diğer
derslerimizde
doğrudan **Windows**
Forms Application
kullanacağımız. İhtiyaç
duyulan bazı
örneklerde **Console**
Application
kullanabiliriz.



FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW

Start Debug Form1.cs* Form1.cs [Design]*

WindowsFormsApplication2.Form1

```
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 WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}
```

namespace:

- birbiri ile ilişkili olan class yapılarını kullanma imkanı sağlar.
- kısacası class silsilesi diyebiliriz.

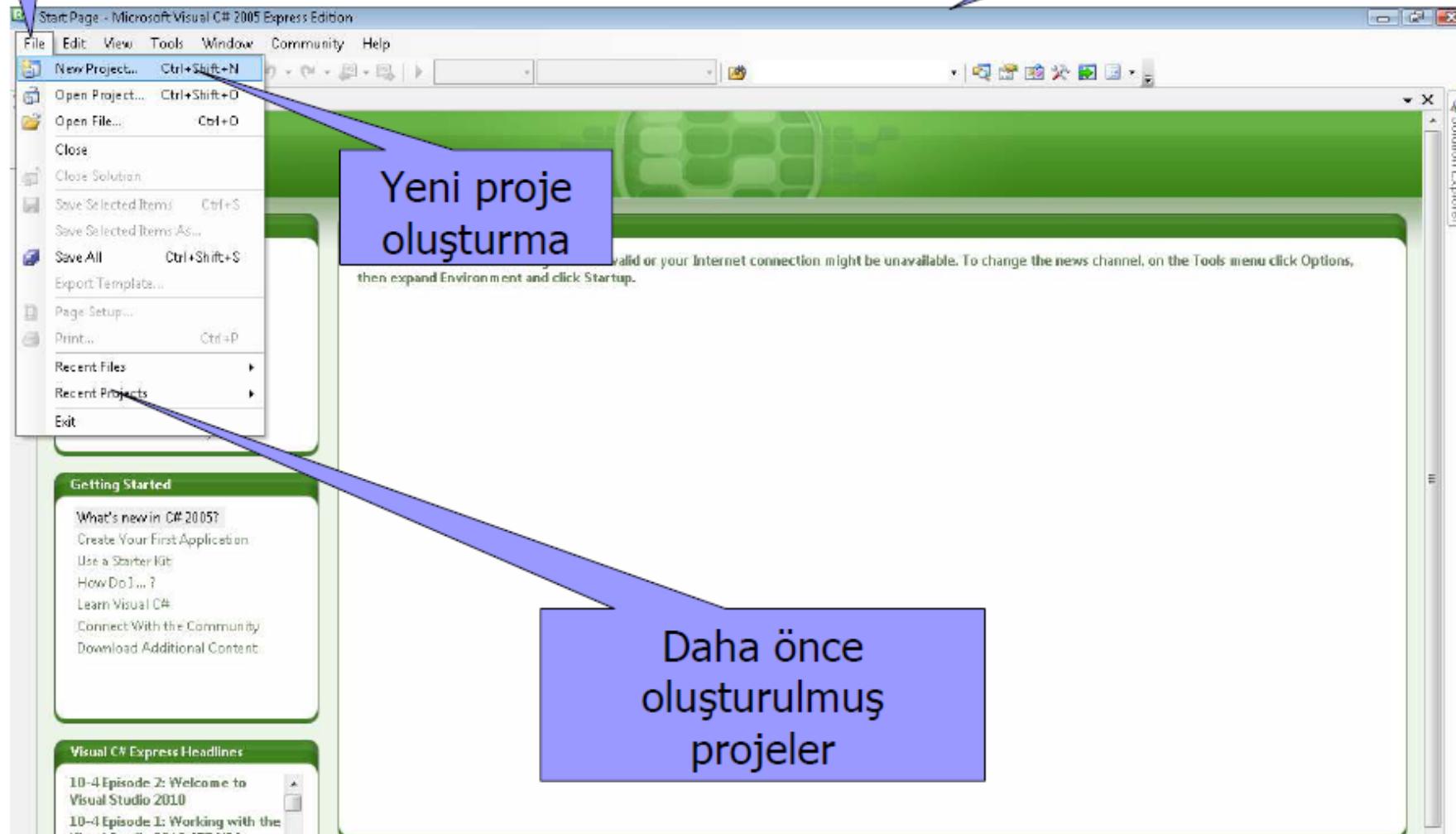
class neydi?
methodları ve özellikleri
tanımladığımız küçük
kütüphanelerdi...

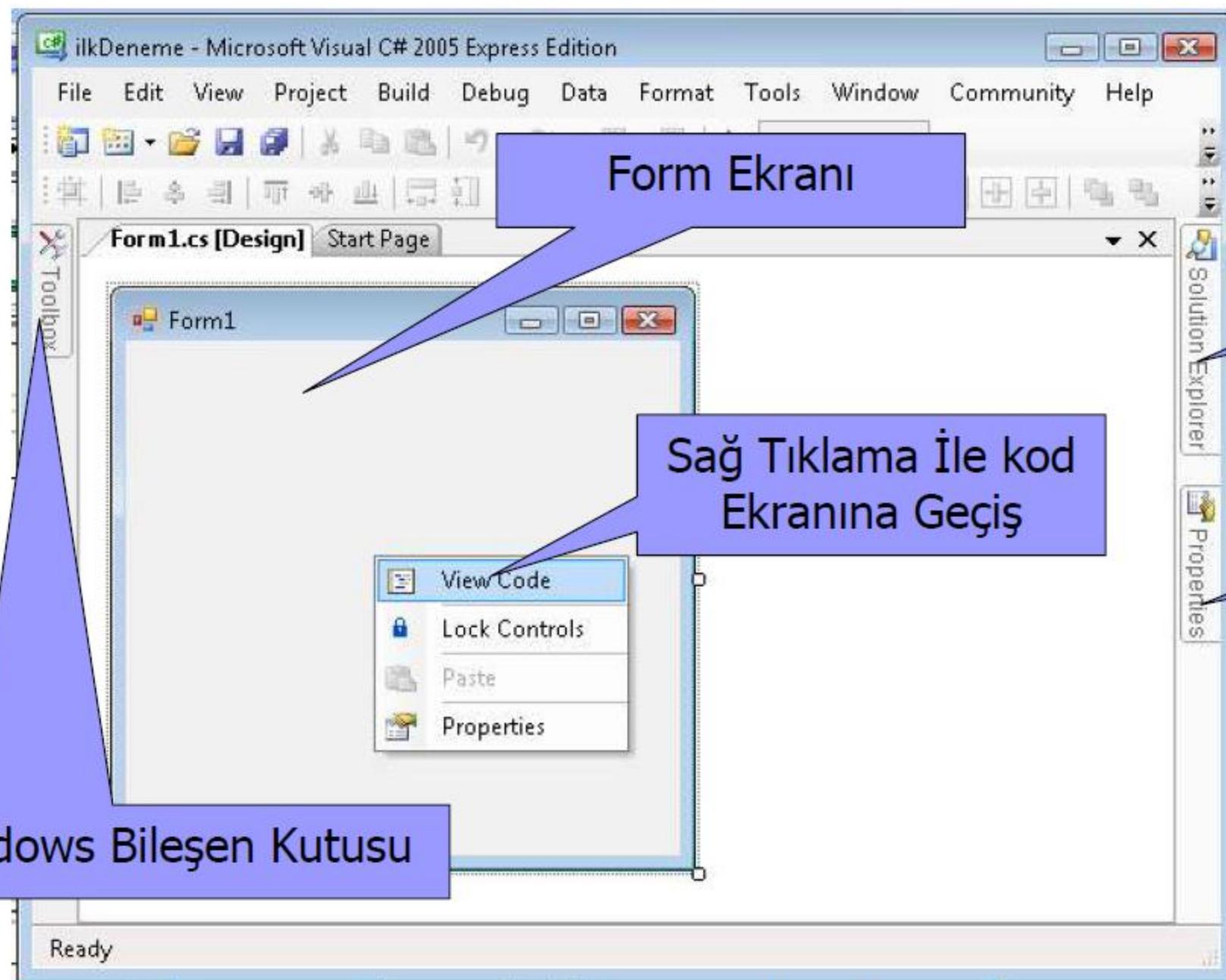
File Menüsü

Program penceresi

Yeni proje
oluşturma

Daha önce
oluşturulmuş
projeler





Proje Gezgini

Özellik Ekranı

Referanslar (Kütüphaneler)

```
Form1.cs  Form1.cs [Design]  Start Page
X
Toolbox
ilkDeneme.Form1

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

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



Ad Alanı

```
Form1.cs  Form1.cs [Design]  Start Page
ilkDeneme.Form1
Form

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

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

Kodlar
Küme
parantezleri
arasına
yazılır

The diagram illustrates the use of curly braces in C# code. A red curly brace groups the entire 'using' block at the top. Another red curly brace groups the 'namespace' declaration and the class definition. A third red curly brace groups the constructor and the call to 'InitializeComponent()'. A fourth red curly brace groups the closing brace of the class definition. A fifth red curly brace groups the closing brace of the namespace declaration. A sixth red curly brace groups the closing brace of the 'using' block.

```
Form1.cs Form1.cs [Design] Start Page
ilkDeneme.Form1 Form1()
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace ilkDeneme
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

Her parantezin
bir kapanışı
olmalıdır.

Form1.cs Form1.cs [Design] Start Page

ilkDeneme.Form1 Form10

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

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

Windows Form
Sınıfı

Solution Explorer Properties

```
Box
using System.Drawing;
using System.Text;
using System.Windows.Forms;

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

        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}
```

Form1 Yükleniği anda çalışacak kodlar

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
```

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

        private void Form1_Load(object sender, EventArgs e)
        {
            // ilk Kodumuzu Yazıyoruz...
            MessageBox.Show("C# Programında Hoşgeldiniz");
        }
    }
}
```

Ekrana Mesaj Kutusu

Açıklama satısı

";" ifade sonu

Kaydetme ve Hangi Dosya Ne işe Yarar?

Tip	Açıklama
Solution Files (.sln)	.sln uzantısı, bir ve birden fazla projenin birbiri ile ilişkili bilgilerini ve global bilgileri tutar. Aynı zamanda projenizi çalıştırmak için kullanabilirsiniz.
Solution Files (.suo)	Solution User Options, breakpoints ve task items gibi bilgileri tutar. Gizli bir dosyadır, .sln ile aynı klasördedir.
Project Files (.csproj)	Projeye ait tüm bilgilerin tutulduğu XML tabanlı bir dosyadır. .sln 'de tüm projenin bilgileri tutulurken, .csproj' de ise sadece tek bir projeninkisi tutulur.
Local Project Items (.cs)	kısaca class, form ve user control bilgilerinin tutulduğu yerdir.
Web Project Items (.aspx, .asmx, .asax)	Web projeleri .cs uzantılarında içerebilir ancak varsayılan dosya uzantısı .aspx'tir. özel işlemler için .asmx, .asax kullanılabilir. (.asmx ve .asax Merak Ediniz, Araştırınız :)
.config	Veritabanı projelerinde veritabanı yolunu tutar.
.xmal	WPF ve Microsoft silverlight uygulamalarında kullanıcı arayüzü elementleri için kullanılır.

Windows Form Application Örnek 2:

- Yeni bir Form açalım ve forma iki tane Buton ekleyelim.
- Button1'in üzerine çift tıklayarak aşağıdaki kodu yazalım

Button1

```
this.Text = "Merhaba";
this.BackColor = Color.Blue;
button2.Visible = true;
button1.Visible = false;
```

- Button2'ye ise;

```
this.Text = "Hoşgeldiniz";
this.BackColor = Color.Red;
button1.Visible = true;
button2.Visible = false;
```

Welcome1.cs // Console Application Örneği

```
1 // welcome1.cs
2 // Text-printing application.
3 using System;
4
5 class Welcome1
6 {
7     // Main method begins execution of C# application
8     static void Main( )
9     {
10         Console.WriteLine( "Welcome to C# Programming!" );
11     } // end method Main
12 } // end class Welcome1
```

Welcome to C# Programming!

Not: Ekranda gözükmesi ve kapanmaması için **ctrl+f5** ya da **Console.ReadKey()**;

Açıklama Satırları

1 // Welcome1.cs

2 // Text-printing application.

- **Açıklayıcı satırlar (Comments) // ile başlar.**
 - Programın işleyişine katkıda bulunmazlar.
 - Kodun anlaşılmasını kolaylaştırırlar.
- **Birkaç satırda açıklama yazmak için: /* . . . */**
/* Birkaç satırda
açıklama yazmak için
kullanabilirsiniz */

3 **using** System;

- **using**

- Önceden tanımlanmış (pre-defined) sınıfların kullanılabilmesine imkan verir.
- Derleyiciye programcının kullanmak istediği sınıfı (class) bulma noktasında yardımcı olur.
- Sınıflar isim uzayları (namespace) altında organize edilirler
 - .NET'in isim uzayları Framework Class Library (FCL) olarak adlandırılır.

-
- **Boş satır**
 - Programın daha kolay okunabilmesini ve anlaşılabilmesini sağlar.
 - Boş satırlar derleyici tarafından dikkate alınmazlar.

 - 5 **class** Welcome1

 - **Welcome1** adındaki sınıfın (**class**) deklarasyonunu gerçekleştirir.
 - Her bir C# programı en azından bir adet kullanıcı tanımlı sınıfa (**user-defined class**) MUHAKKAK sahip olmalıdır.
 - **class** anahtar kelimesini müteakip sınıfın adı yazılmalıdır.
 - Sınıflar adlandırılırken her bir kelimenin ilk harfi büyük olmalı ve kelimeler bitişik yazılmalıdır. (**Pascal notasyonu**)
 - OrnekSinifAdi

C# Keywords

abstract	as	Base	bool	break
byte	case	Catch	char	checked
class	const	Continue	decimal	default
delegate	do	Double	else	enum
event	explicit	Extern	false	finally
fixed	float	For	foreach	goto
if	implicit	In	int	interface
internal	is	Lock	long	namespace
new	null	Object	operator	out
override	params	private	protected	public
readonly	ref	return	sbyte	sealed
short	sizeof	stackalloc	static	string
struct	switch	this	throw	true
try	typeof	uint	ulong	unchecked
unsafe	ushort	using	virtual	void
volatile	while			

Hatırlatma

.cs uzantılı program dosyanızın adı ile program içerisinde kullandığınız sınıfınızın adının aynı olmasına dikkat ediniz.

```
6  {
```

```
12 } // end class Welcome1
```

Her sınıfın gövdesi { ile başlar ve } ile biter.

Hatırlatma

Açıığınız her parantezi **{ hemen kapatırsanız }** ve daha sonra parantezler arasında yapmak istediğiniz komutları eklerseniz hata yapma oranınız azalır. Açılan her parantezin muhakkak kapatılması gerektiğini, aksi takdirde programınızın hata vereceğini unutmayınız.

- Bütün C# uygulamaları çalışmaya **Main** fonksiyonunu işletecek başlar.
- **Main** fonksiyonu olmayan bir C# uygulaması düşünülemez.
- **Main** kelimesinin ilk harfi olan **M** daima büyük harfle yazılmalıdır. (**C#’ın case-sensitive olduğunu unutmayın.**)
- **Main** kelimesinden sonra kullanılan normal parantezler bir fonksiyonun (metodun) varlığına işaret eder.
- Metodlar belirli işler yaparlar ve genelde geriye değer döndürürler.
 - **void** anahtar kelimesi **Main** fonksiyonunun geriye hiçbir değer döndürmediğini belirtir.

10

```
Console.WriteLine( "Welcome to C# Programming!" );
```

- Bu satır bir C# deyimidir (statement) ve her C# deyimi MUHAKKAK bir noktalı virgül ; ile sonlanmalıdır.
- **Console.WriteLine** metodu konsol penceresine (command window) bilgi yansıtmak için kullanılır. Bu örneğimizde ekrana string dediğimiz ve çift tırnak içerisinde tanımladığımız karakter dizisini basmak amacıyla kullanılmıştır.

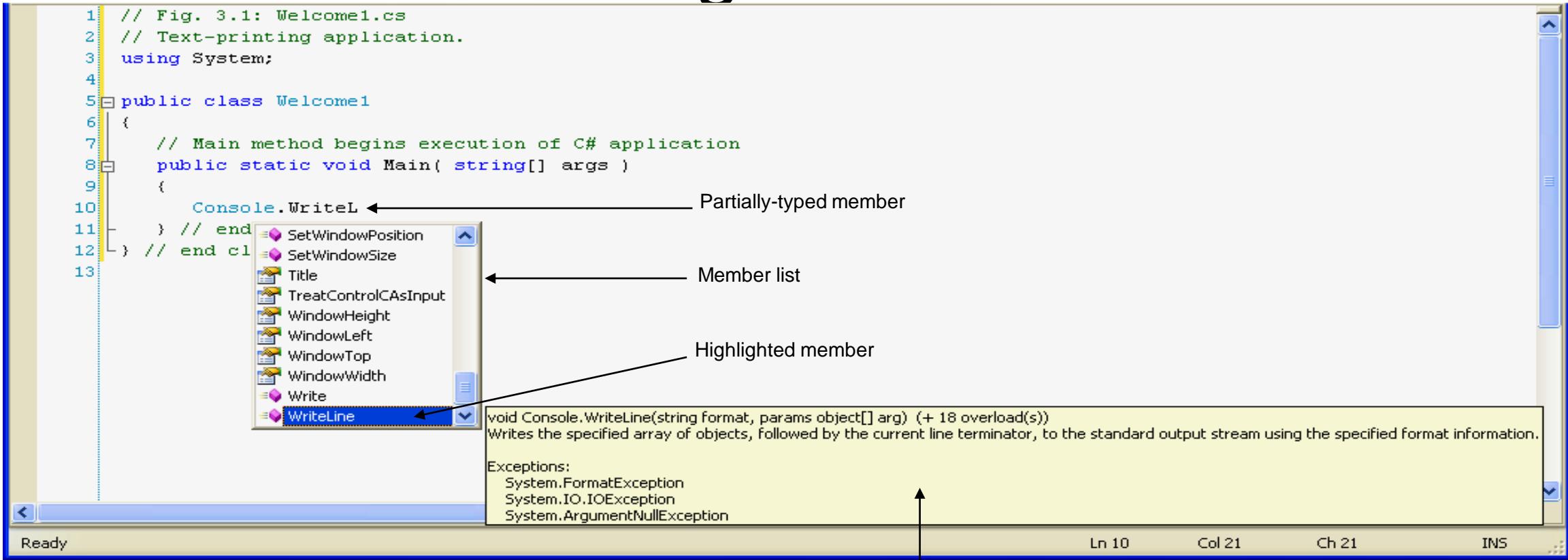
Hatırlatma

Herhangi bir C# deyiminin sonuna noktalı virgül eklemeyi unutmak programımızın hata vermesine neden olur.

Hatırlatma-Syntax Error

Derleyicinin vereceği bir syntax hataları hata mesajıyla belirtilen satırda olmayabilir. Bu tür durumlarda hata satırından önceki satırları kontrol etmek faydalıdır.

intelliSense özelliği



Visual C# Express 2013 Programının intelliSense özelliği.
(Migros.Temizlik Ürünleri.Sıvı Sabunlar.Duru)

Down arrow Parameter Info window

```
public class Welcome1
{
    // Main method begins execution of C# application
    public static void Main( string[] args )
    {
        Console.WriteLine()
    } // e
}
```

Up arrow

1 of 19 void **Console.WriteLine()**
Writes the current line terminator to the standard output stream.

Parameter Info penceresi.

Welcome2.cs

```
1 // Welcome2.cs
2 // Printing one line of text with multiple statements.
3 using System;
4
5 class Welcome2
6 {
7     // Main method begins execution of C# application
8     static void Main( )
9     {
10         Console.Write( "Welcome to " );
11         Console.WriteLine( "C# Programming!" );
12     } // end method Main
13 } // end class Welcome2
```

Welcome to C# Programming!

Cıktı sonrası imleç yerinde kalır.

Cıktı sonrası imleç bir alt satırın başına gider.

-
- **Kaçış (Escape) Karakterleri**
 - Bir string içerisinde yani çift tırnaklar arasında kullanılırlar.
 - Backslash (\) karakterini takip ederler.
 - Özel bir karakterin çıktısının alınacağına işaret ederler.
 - **Yeni Satır (Newline) Karakteri (\n)**
 - **Console.Write** ve **Console.WriteLine** metodları bu kaçış karakterini imleci bir alt satıra götürmek şeklinde yorumlarlar.

Welcome3.cs

```
1 // Welcome3.cs
2 // Printing multiple lines with a single statement.
3 using System;
4
5 class Welcome3
6 {
7     // Main method begins execution of C# application
8     static void Main()
9     {
10         Console.WriteLine("Welcome\n to\n C#\n Programming!");
11     } // end method Main
12 } // end class Welcome3
```

Welcome
to
C#
Programming!

\n escape sırasının çıktıyı etkilediğine dikkat ediniz.

Bazi Yaygın Escape Sıraları

Escape sequence	Description
\n	Newline. Positions the screen cursor at the beginning of the next line.
\t	Horizontal tab. Moves the screen cursor to the next tab stop.
\r	Carriage return. Positions the screen cursor at the beginning of the current line—does not advance the cursor to the next line. Any characters output after the carriage return overwrite the characters previously output on that line.
\\"	Backslash. Used to place a backslash character in a string.
\"	Double quote. Used to place a double-quote character ("") in a string. For example, <pre>Console.WriteLine("in quotes");</pre> displays "in quotes"

Welcome4.cs

```
1 // Welcome4.cs
2 // Printing multiple lines of text with string formatting.
3 using System;
4
5 class Welcome4
6 {
7     // Main method begins execution of C# application
8     static void Main( )
9     {
10         Console.WriteLine( "{0}\n{1}", "welcome to", "C# Programming!" );
11     } // end method Main
12 } // end class Welcome4
```

Welcome to
C# Programming!

Console.WriteLine
formatlanmış veriyi ekrana
basar.

```
2 // Displaying the sum of two numbers input from the keyboard.
```

```
3 using System; ←  
4  
5 class Addition  
6 {
```

```
7     // Main method begins execution of C# application
```

```
8     static void Main( )
```

```
9     {
```

```
10    int number1; // declare first number to add
```

```
11    int number2; // declare second number to add
```

```
12    int sum; // declare sum of number1 and number2
```

```
14    Console.WriteLine( "Enter first integer: " ); // prompt user
```

```
15    // read first number from user
```

```
16    number1 = Convert.ToInt32( Console.ReadLine() ); ←
```

Addition.cs

number1, number2 ve sum adlarında ve integer veri tipinde 3 değişken deklare et.

```
18    Console.WriteLine( "Enter second integer: " ); // prompt user
```

```
19    // read second number from user
```

```
20    number2 = Convert.ToInt32( Console.ReadLine() ); ←
```

Bir string olan kullanıcı girişini integer veri tipine çevir ve number1 adlı değişkene ata.

```
22    sum = number1 + number2; // add numbers
```

Bir string olan kullanıcı girişini integer veri tipine çevir ve number2 adlı değişkene ata.

```
24    Console.WriteLine( "Sum is {0}", sum ); // display sum
```

number1 ve number2 değişkenlerini topla ve sonucu sum adlı değişkene ata.

```
25 } // end method Main
```

```
26 } // end class Addition
```

Enter first integer: 45

Enter second integer: 72

Sum is 117

Formatlı çıktı olarak toplamı ekranda görüntüle.

```
5 class Addition
```

```
6 {
```

Addition adlı sınıfı deklare eder. Dosya adı ile sınıf adının aynı oluşuna dikkat ediniz.

```
8 static void Main( )
```

```
9 {
```

Main fonksiyonu tanımlanır. (MUHAKKAK)

Değişkenler bellekte bulunan ve değer saklama amacıyla kullanılan ünitelerdir. Her değişken kullanılmadan önce **muhakkak** tanımlanmalıdır ve bir isimle bir veri tipine sahip olmalıdır. Bir değişken deklarasyonu muhakkak bir noktalı virgül ile sonlanmalıdır. Değişkenlere kullanım maksatlarına uygun isimler verilmelidir. Değişkenlere ilk değer atama işlemi (initialization) deklarasyon yanında da yapılabilir , daha sonra da.

```
10 int number1; // first number to add  
11 int number2; // second number to add  
12 int sum;      // sum
```

- **int** veri tipinde (data type) **number1**, **number2** ve **sum** adlarında 3 değişken deklare edilir.
 - **int** veri tipi tamsayı değerleri saklamak için kullanılır. (45 veya -67.)
 - **float**, **double** ve **decimal** veri tipleri ondalıklı sayı saklamak için kullanılırlar. (12.345)
 - **char** veri tipi tek bir karakter saklamak için kullanılır.
 - **int**, **float**, **double**, **decimal** ve **char** basit veri tipleridir. (built-in)
- Değişkenlerin kullanım amaçlarını belirtmek için açıklayıcı satırlar (comments ler) eklenebilir.

```
int number1, // first number to add  
      number2, // second number to add  
      sum;      // sum
```

- Aynı veri tipine sahip birden fazla değişken birbirlerinden virgülle ayrılarak topluca deklare edilebilir.

Hatırlatma

Her bir değişkeni farklı bir satırda deklare etmek açıklayıcı satır eklenmesi işini kolaylaştırır.

Değişken adları oluşturulurken genel olarak deve notasyonundan (camel notation) faydalанılır. Bu notasyon değişken adlarının ilk kelimesinin ilk karakterini küçük, diğer kelimelerin ilk karakterlerini büyük yapmak şeklinde işletilir.

enKucukSayı

Bazı Veri Tipleri (Saklama Kaplarını Düşünün)

.NET YAPISI	C#	BOYUT(byte)	KAPASİTE
System.Byte	byte	1	0-255
System.Sbyte	sbyte	1	-128 ile 127
System.Int16	short	2	-32768 ile 32767
System.Uint16	ushort	2	0 ile 65535
System.Int32	int	4	-2147483648 ile 2147483647
System.Uint32	uint	4	0 ile 4294967295
System.Int64	long	8	???
System.Uint64	ulong	8	???
System.Single	float	4	7 digits
System.Double	double	8	15 digits
System.Decimal	decimal	8	29 digits (Bankacılık İşlemleri)
System.Char	char	2	Unicode karakterler
System.String	string		Metin Dizileri
System.Boolean	bool	1	true veya false

Değişken Deklarasyonu ve Değer Atama Örnekleri

- `bool buyukMu;`
- `bool test=false;`
- `char kullaniciGiris='y';`
- `ushort kardesSayisi;`
- `int sicaklik;`
- `float ortalama=2.345;`
- `double piSayisi=3.1418;`
- `decimal hesapBakiyesi=1100.57;`

ÖDEV

UYGULAMA

- Kullanıcıdan klavye yoluyla alacağı 3 tamsayının en büyüğünü ve en küçüğünü bularak ekrana yansıtacak bir C# konsol uygulaması yazınız. (if şartlı deyiminin nasıl kullanıldığını araştırınız.)