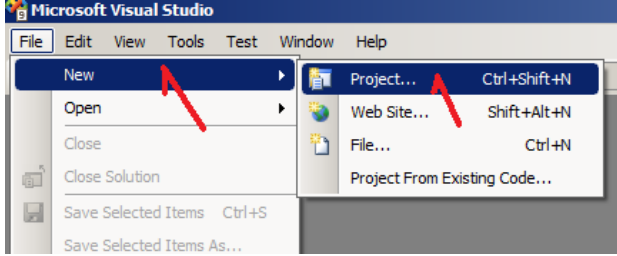


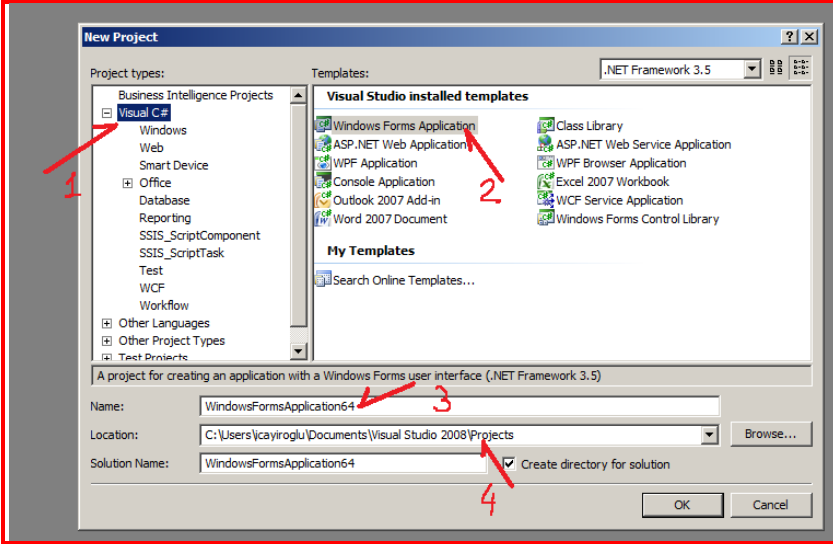
İNTERNET TABANLI PROGRAMLAMA

VISUAL STUDIO'DA YENİ BİR PROJE OLUŞTURMA

Visual Studio (VS) programını çalıştırdığımızda karşımıza boş bir ekran gelir. Yeni bir proje oluştururken File>New>Project yolu kullanılarak yeni bir proje başlatırız.

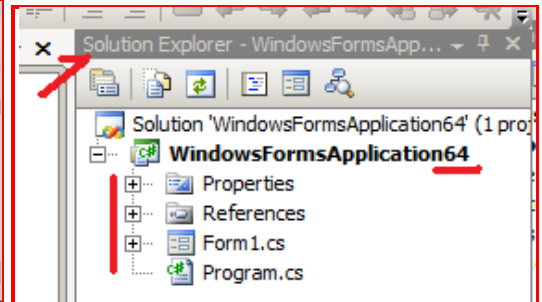
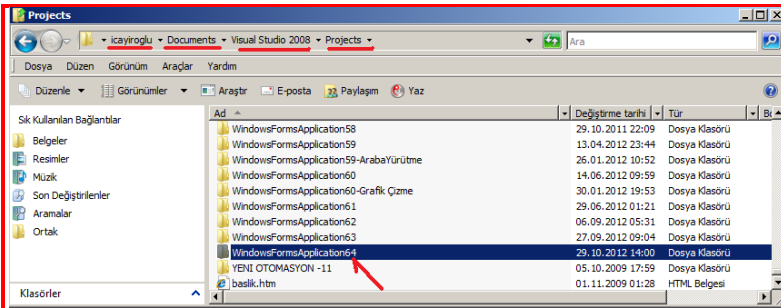


Karşımıza çıkan ekrandan C# dili seçili olmalı (1 nolu). Programımız masaüstü bir program olacağından ve Windows ortamında çalışan bir program olacağından "Windows Forms Application" seçili olmalı. Projenin adı 3 nolu yerde gösterilen addır. Bu projenin bilgisayarımızda nerede kayıtlı olacağını gösteren yer ise 4 nolu yerdir. VS yi kapattıktan sonra hazırladığımız programı başka bir yere taşımak istiyorsak 4 nolu yere gidip orada 3 numara ile gösterilen ismin bulunduğu klasörü alıp kopyalayabiliriz.

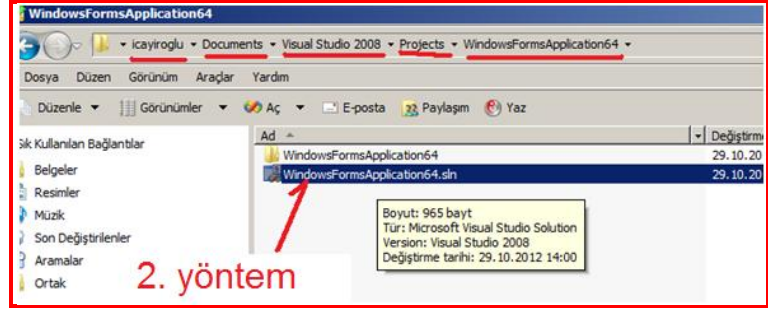
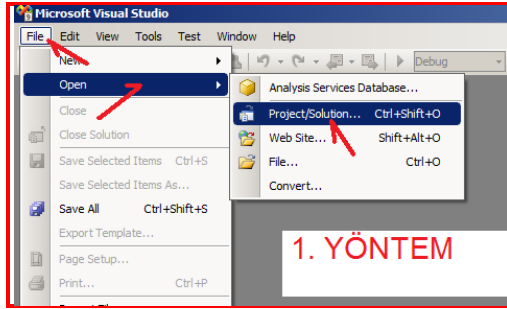


Ok düğmesine bastıktan sonra VS bize projemizin temelinı oluşturan ilk yapıyı hazır olarak verecektir. Bunlarla ilgili ilk kodları projemizin içerisine atar ve bilgisayarımızdaki 4 numara ile gösterilen adresteki yere dosyaları kopyalar.

Şimdi bilgisayarımızdaki bu dosyaları görelim. Göreceğimiz gibi 64 numaralı proje oluşturulmuş durumda. Aynı proje VS içerisinde de şuan açık durumdadır. Biz VS içerisinde projenin dosyaları arasında gezerken VS nin kendi Gezgin (explorer) penceresini kullanırız. Buna "Solution Explorer" penceresi diyoruz.



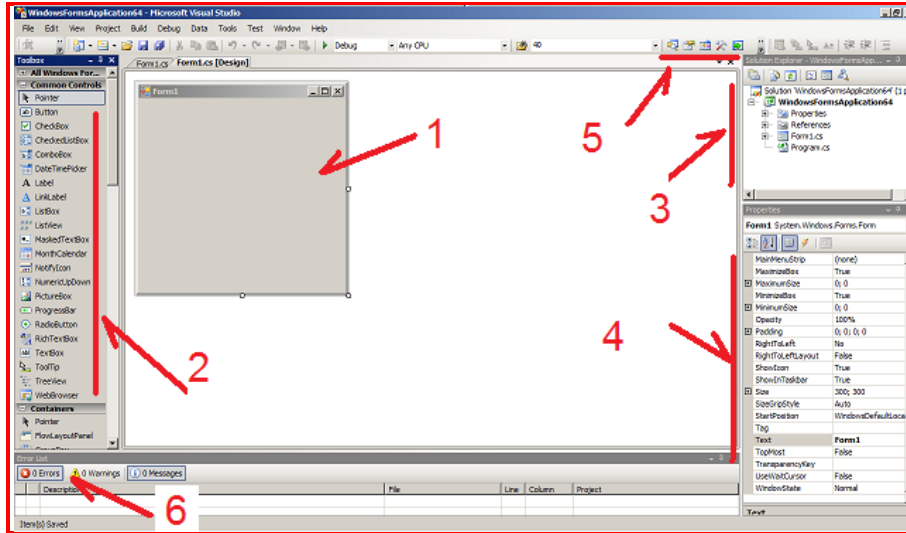
Projemizde epey bir çalışma yaptığımızı düşünelim. Ertesi günü tekrar projemizi açmak istersek iki yolu kullanabiliriz. Ya VS nin içerisinde File>Open>Project yolunu kullanıp buradaki 64 numaralı projeyi açarız. Yada Windows'un kendi gezgin penceresinden gidip 64 nolu projenin ana dosyasına çift tıklayıp VS ile birlikte projenin açılmasını sağlayabiliriz.



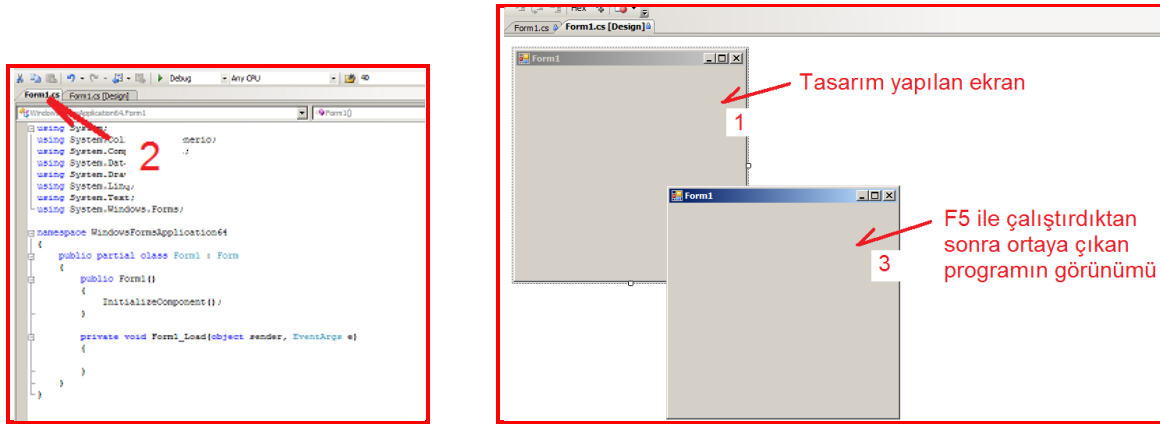
İlk projemizi New deyip oluşturduktan sonra karşımıza aşağıdaki gibi bir ekran gelecektir. Bu ekranda bize ilk olarak hazır bir nesne diyebileceğimiz Form nesnesi verilmiştir. Bu nesne programımızın zeminin oluşturan düz bir çerçevedir (1 nolu). Her nesnenin Özellikleri (properties) ve üzerinde gerçekleştirilebilecek olayları (events) vardır demiştik. Örneğin Form nesnesinin zemin rengini farklı bir renkte yapmak istiyoruz. Bunu nereden ayarlayabiliriz. Sağ taraftaki Properties penceresinden bu Bunu BackColor (arka renk) kısmında değiştirebiliriz. İşte bunun gibi projemize eklediğimiz hazır nesnelerin birçok özelliğini sağ taraftaki Preoperties penceresinde (4 nolu) ayarlayabiliriz.

Dikkat edersek buradaki projemizde sadece bir tane hazır nesne (Form nesnesi) bulunmaktadır. Eğer istersek bu formun üzerine daha birçok hazır nesneleri ekleyebiliriz. Bu iş için sol taraftaki Toolbox (araçkutusu) (2 nolu) penceresini kullanabiliriz. Örneğin formun üzerine TextBox (yazı yazma kutucukları) button (buton) label (etiket) gibi daha bir çok nesneyi sol taraftan sürükleyip formun üzerine getirebiliriz.

Dikkat edersek ekranımızda birçok panel denilen pencereler bulunmaktadır. Bunları tekrar açılsak Toolbox hazır nesnelerin bulunduğu panel, Solution Explorer projemizin içinde bulunan dosyaları, Properties ise seçili olan nesnenin özelliklerini değiştirmeyi sağlayan kısımdır. Kodlarda bir hata var ise bu hatanın nerede olduğunu gösteren 6 nolu kısımda Errors panelimiz bulunmaktadır. İşte bütün bu paneller eğer ekranımızda gözükmeyp kaybolmuş ise 5 numara ile gösterilen yerdeki düğmeler tıklarsak ortaya çıkacaktır.



Projemizi hazırlarken üç tane ekran bizim için önemlidir. Bunlardan birincisi Design penceresi. Bu pencere projemizin nasıl görüneceğini bize gösterir (1 nolu). Diğeri içeresine kod yazdığımız ekrandır. Bu ekranda C# kodlarını yazacağız (2 nolu). Birde Programı F5 ile çalıştırdıktan sonra Design penceresini çalışır halde gördüğümüz ekrandır (3 nolu).



C# da büyük küçük harf ayrımı vardır. Her satırın sonuna mutlaka ; işareti konulmalıdır.

DEĞİŞKENLER VE VERİ TİPLERİ

Verilerin tutulacağı değişkenlerin tanımlanması zorunludur. Değişken tanımlanırken hangi tip veri türü tutulacağı ve hangi aralıkta çalışacağına dikkat edilmelidir. Çalışacağı aralık içerisinde mümkün olduğunca en düşük hafıza tutan veri türünü tercih etmek gerekir. Değişkenlerin tanımlanması karmaşık program yapılarında bilgilerin karışmalarını engellemesi açısından ve en az ram kaynaklarını kullanmaya neden olduğu için kullanımı önemlidir. Değişken türleri aşağıdaki tabloda verilmiştir. Bir değişken hafızada tutmuş olduğu byte sayısı kadar bilgiyi tutabilir. Örneğin $2^8 = 1 \text{ byte} = 256$ kadar olan sayıları tutabilir. Yani 0 ile 256 arası sayıları tutabilir. Bu şekilde tanımlama işaretli (signed) tanımlama olur Eğer negatif bölgeye de geçiş yapılsa bu sayı ikiye bölünür. -128 ile +128 arasında bilgiler tutulmuş olur. Bu tanımlama işaretli tanımlama olur.

Adı		Hafıza (byte)	Sınır Değerleri	İşaretsiz (unsigned) Değerleri		
Tamsayı	sbyte	1	-128 : + 127	byte	1	0 : + 255
	short	2	-32 768 : +32 767	ushort	2	0 : + 65 535
	int	4	-2 147 483 648 : + 2 147 483 648	uint	4	0 : +4 294 967 295
	long	8	-9 223 372 036 854 775 808 : +9 223 372 036 854 775 808	ulong	8	0 : +18 446 744 073 709 551 616
Ondalık	float	4	$\pm 3.6 \times 10^{-38} : \pm 3.6 \times 10^{+38}$ (tek duyarlık)	Yoktur		
	double	8	$\pm 1.8 \times 10^{-308} : \pm 1.8 \times 10^{+308}$ (çift duyarlık)	Yoktur		
	decimal	16	28 digit ondalık sayı tutar.	Yoktur		
metin	char	2	Unicode tek karakteri tutmak içindir.			
	string	2x	Birden fazla karakteri tutmak içindir.			
	bool	1 (bit)	0 : 1 (false – true)			

Değişkenlerin Yaşam Süreleri (Geçerli oldukları aralıklar)

Değişkenlerin tanım aralıkları ve hafızada tuttıkları yerin yanında yaşam süreleri yada geçerli oldukları bölge hakkında da bilgi sahibi olmamız gerekir. Buna göre tanımlanan değişkenler dört farklı şekilde açıklanabilir.

a) Local (yerel) değişkenler: Bu değişkenler sadece tanımlandıkları fonksiyon içinde geçerlidirler. Tanımlandıkları fonksiyon dışından ulaşılmak mümkün değildir. Fonksiyon çağrıldığında hafızada oluşturulurlar, fonksiyondan çıktığında ise tekrar hafızadan silinirler.

b) Global (genel) değişkenler: Tüm fonksiyonların dışında tanımlanırlar. Dolayısı ile tüm fonksiyonlarda geçerli olurlar. Program çalışmaya başladığı anda hafızada yer alırlar ve program çalıştığı sürece hafızada kalırlar. Program sona erdiğinde hafızadan silinirler.

Doğru Değişken Yazımı

string 1isim10; (yanlış)
string isim10; (doğru)
string ad soyad; (yanlış)
string ad_soyad; (doğru)
string AdSoyad;
string true; (yanlış)

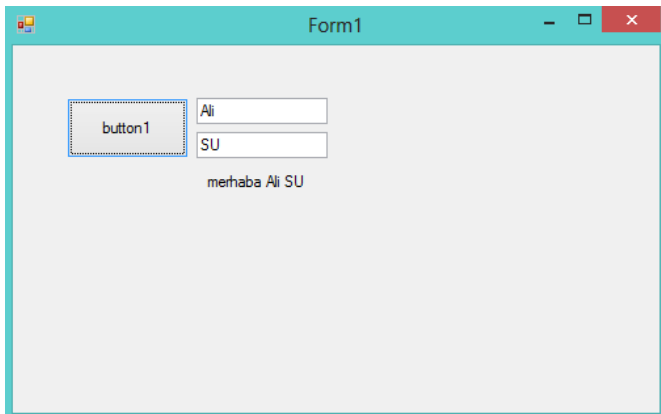
Not: 1 byte = 8 bit = $2^8 = 256$ demektir. Eğer işaretli ise 0-255 arasındaki sayıları tutacak demektir. 2 byte = $(2^8)^2 = 2^{16} = 65536$ demektir. Eğer işaretli ise 0-65536 arasındaki sayıları tutar. Şayet işaretli ise -32000 ile + 32000 küsur sayılar arasındaki rakamları tutar

4 byte = $(2^8)^4 = 4.294.967.296$ sayısına karşılık gelir.

MATEMATİK (Math) KÜTÜPHANESİ

Matematik hesaplamalarında kullanılabilecek önemli fonksiyonlar aşağıda belirtilmiştir.

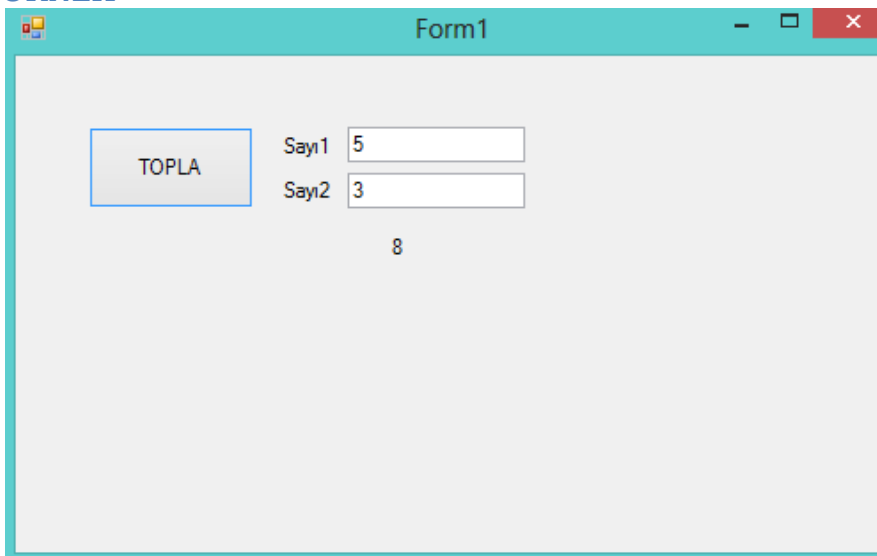
Math.E;	// e sayısını verir
Math.PI;	// pi sayısını verir
Math.Sin(b);	// b sayısının sin değerini alır
Math.Cos(b);	// b sayısının Cos değerini alır
Math.Tan(b);	// b sayısının Tan değerini alır
Math.Exp(b);	// e^b demektir
Math.Pow(b,c);	// b^c demektir
Math.Sqrt(b);	// Karekök değerini alır daha fazla kök için $a^{(2/3)}$, Math.Pow(a,(2/3))
Math.Ceiling(b);	// Ondalık sayıyı üste yuvarlar, b=10.3 , 11 çıkar
Math.Floor(b);	// Ondalık sayıyı aşağıya yuvarlar, b=10.3 , 10 çıkar
Math.Round(b);	// En yakın tamsayıya yuvarlar, b=10.3 , 10 çıkar, b=10.7 den 11 olur. b=10.49864 sayısı , Dikkat b=10.5 sayısını 10 yuvarlar.
Math.Min(b,c);	//b ve c sayısından en küçük sayıyı verir. b=3 , c=4 ise sonuç 3 çıkar
Math.Max(b,c);	//iki sayıdan en büyük olanını döndürür.
Math.Abs(b);	// sayının mutlak değerini alır, yani tüm sayılar pozitif çıkar.
Math.Log10(b);	// b sayısının 10 tabana göre logaritmasını alır. b=100 ise sonuç 2 çıkar $b=10^2 \Rightarrow 2$ çıkar.
Math.Log(b);	// b sayısının ln'ini almaktadır. e tabanına göre logaritmasını alır.
Math.Log(b,c);	//c tabanında b sayısının logaritmasını alır. Örneğin b=8 ve c=2 ise sonuç 3 tür.

ÖRNEK

```
private void button1_Click(object sender, EventArgs e)
{
    string Ad, Soyad;
    int Yas;
    double Ortalama;

    Ad = textBox1.Text;
    Soyad = textBox2.Text;

    label1.Text = "merhaba " + Ad + " " + Soyad;
}
```

ÖRNEK

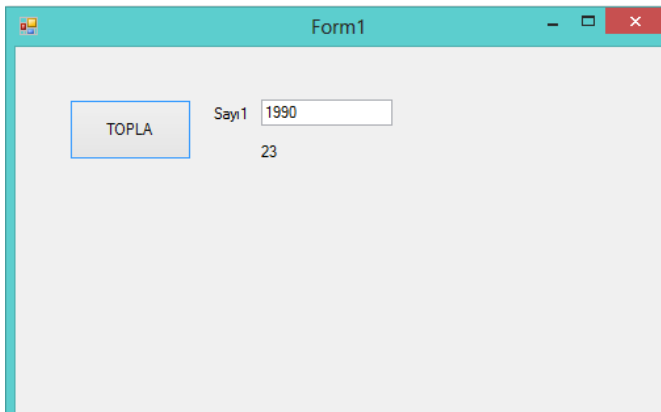
```
private void button1_Click(object sender, EventArgs e)
{
    int Sayı1, Sayı2;

    Sayı1 = Convert.ToInt32(textBox1.Text);
    Sayı2 = Convert.ToInt32(textBox2.Text);

    label1.Text = (Sayı1 + Sayı2).ToString();
}
```

```
}
```

ÖRNEK

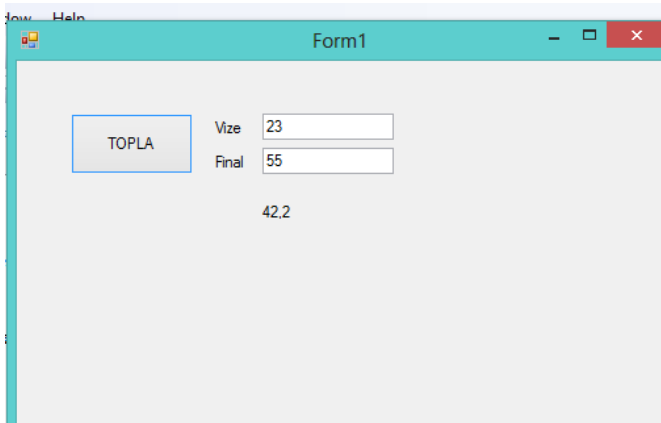


```
private void button1_Click(object sender, EventArgs e)
{
    int DogumTarihi, Yas;

    DogumTarihi = Convert.ToInt32(textBox1.Text);
    Yas = 2013 - DogumTarihi;

    label1.Text = Yas.ToString();
}
```

ÖRNEK



```
private void button1_Click(object sender, EventArgs e)
{
    int Vize, Final;
    double Ortalama;

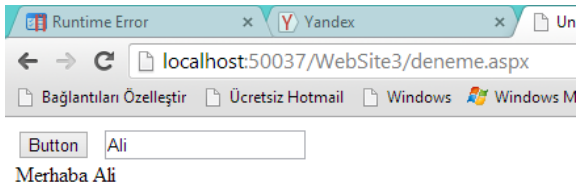
    Vize = Convert.ToInt32(textBox1.Text);
    Final = Convert.ToInt32(textBox2.Text);

    Ortalama = Vize * 0.40 + Final * 0.60;

    label1.Text = Ortalama.ToString();
}
```

}

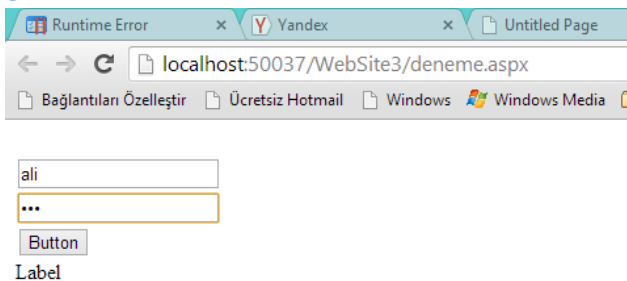
ÖRNEK



```
protected void Button1_Click(object sender, EventArgs e)
{
    string Ad;
    Ad = TextBox1.Text;

    Label1.Text = "Merhaba " + Ad;
}
```

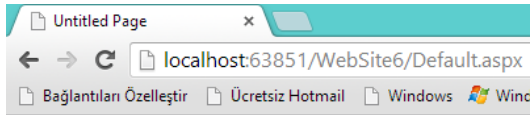
ÖRNEK



```
protected void Button1_Click(object sender, EventArgs e)
{
    string KullaniciAdi, Sifre;

    KullaniciAdi = TextBox1.Text;
    Sifre = TextBox2.Text;

    if (KullaniciAdi == "ali" && Sifre == "123")
    {
        Label1.Text = "Hoş Geldiniz!..";
    }
    else
    {
        Label1.Text = "Şifre yanlıştır!..";
    }
}
```

ÖRNEK

Vize 45

Final 67

58,2

Kaldı

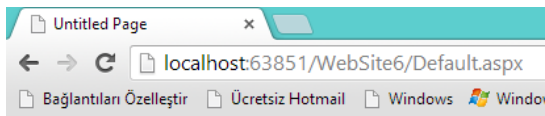
```
protected void Button1_Click(object sender, EventArgs e)
{
    double Vize, Final, Ortalama;

    Vize = Convert.ToDouble( TextBox1.Text);
    Final =Convert.ToDouble( TextBox2.Text);

    Ortalama = Vize * 0.40 + Final * 0.6;

    Label1.Text = Ortalama.ToString();

    if (Ortalama >= 60 && Ortalama <= 100)
    {
        Label2.Text = "Geçti";
    }
    else if (Ortalama >= 0 && Ortalama < 60)
    {
        Label2.Text = "Kaldı";
    }
    else
    {
        Label2.Text = "HATALI NOT";
    }
}
```

ÖRNEK

Button

Ad Oya

Soyad AY

☐ Bay☒ Bayan

Bayan Oya AY Hoşgeldiniz

```
protected void Button1_Click(object sender, EventArgs e)
{
```



```

string Ad, Soyad;
string Cinsiyet =null;

Ad = TextBox1.Text;
Soyad =TextBox2.Text;

if (RadioButton1.Checked ==true)
{
    Cinsiyet = "Bay ";
}
else if (RadioButton2.Checked == true)
{
    Cinsiyet = "Bayan ";
}

Label1.Text  = Cinsiyet + Ad + " " + Soyad + " Hoşgeldiniz";

}

```

ÖRNEK

```

protected void Button1_Click(object sender, EventArgs e)
{
    string Ad, Soyad;
    string Cinsiyet =null;
    string Dersler = null;

    Ad = TextBox1.Text;
    Soyad =TextBox2.Text;

    if (RadioButton1.Checked ==true)
    {
        Cinsiyet = "Bay ";
    }
    else if (RadioButton2.Checked == true)
    {
        Cinsiyet = "Bayan ";
    }
}

```

```

    }

    if (CheckBox1.Checked == true)
    {
        Dersler =Dersler + " Matematik ";
    }
    if (CheckBox2.Checked == true)
    {
        Dersler =Dersler + " Fizik ";
    }
    if (CheckBox3.Checked == true)
    {
        Dersler =Dersler + " Kimya ";
    }

    Label1.Text = Cinsiyet + Ad + " " + Soyad + Dersler + "
Seçtiniz" ;
}

```

ÖRNEK

```

protected void Button1_Click(object sender, EventArgs e)
{
    string Ad, Soyad;
    string Cinsiyet =null;
    string Dersler = null;
    string Sehir = null;

    Ad = TextBox1.Text;
    Soyad =TextBox2.Text;

```

```

        if (RadioButton1.Checked == true)
        {
            Cinsiyet = "Bay ";
        }
        else if (RadioButton2.Checked == true)
        {
            Cinsiyet = "Bayan ";
        }

        if (CheckBox1.Checked == true)
        {
            Dersler = Dersler + " Matematik ";
        }
        if (CheckBox2.Checked == true)
        {
            Dersler = Dersler + " Fizik ";
        }
        if (CheckBox3.Checked == true)
        {
            Dersler = Dersler + " Kimya ";
        }

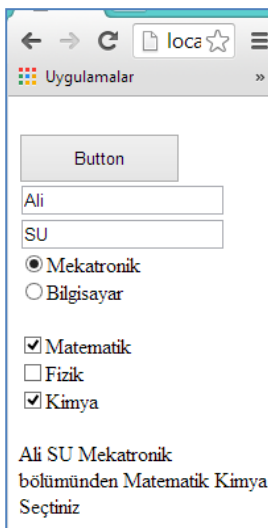
        Sehir = ListBox1.SelectedItem.Text;

        Label1.Text = Cinsiyet + Ad + " " + Soyad + " " + Sehir +
        "'dan " + Dersler + " derslerini alınız.";

    }

```

ÖRNEK



```

protected void Button1_Click(object sender, EventArgs e)
{
    string Ad, Soyad, Bolu=null, Dersler=null;

    Ad = TextBox1.Text;
    Soyad = TextBox2.Text;

```

```

        if (RadioButton1.Checked == true)
            Bolum = "Mekatronik ";
        else if (RadioButton2.Checked == true)
            Bolum = "Bilgisayar ";

        if (CheckBox1.Checked == true)
            Dersler = Dersler + " Matematik ";

        if (CheckBox2.Checked == true)
            Dersler = Dersler + " Fizik ";

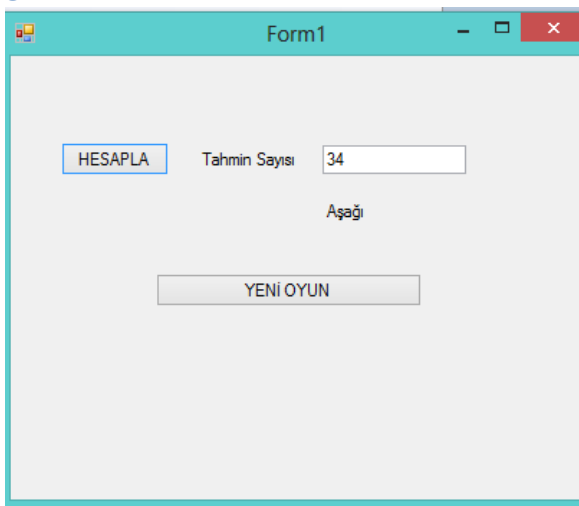
        if (CheckBox3.Checked == true)
            Dersler = Dersler + " Kimya ";

        Label1.Text = Ad + " " + Soyad + " " + Bolum + " bölümünden " + Dersler + " Seçtiniz ";

    }

```

ÖRNEK



```

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

namespace WindowsFormsApplication6
{
    public partial class Form1 : Form
    {
        public Form1()
        {

```

```
        InitializeComponent();
    }

    int RastgeleSayi = 0;
    int sayac = 0;

    private void button1_Click(object sender, EventArgs e)
    {
        sayac = sayac + 1;

        int TahminSayisi = Convert.ToInt32(txtAd.Text);

        if (TahminSayisi > RastgeleSayi)
            label1.Text = " Aşağı";
        else if (TahminSayisi < RastgeleSayi)
            label1.Text = " Yukarı";
        else
            label1.Text = " Tebrikler" + sayac + "hakta
bildiniz";

    }

    private void button2_Click(object sender, EventArgs e)
    {
        Random Rastgele = new Random();
        RastgeleSayi = Rastgele.Next(1, 100);

        label1.Text = "";

    }

    }
}
```