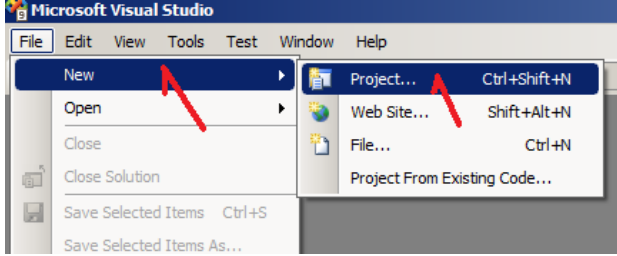


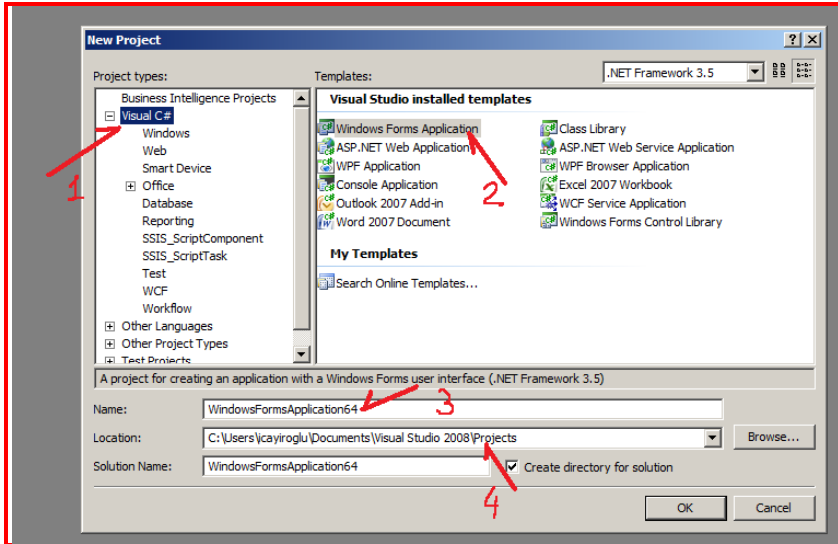
BİLGİSAYAR PROGRAMLAMA-II

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.

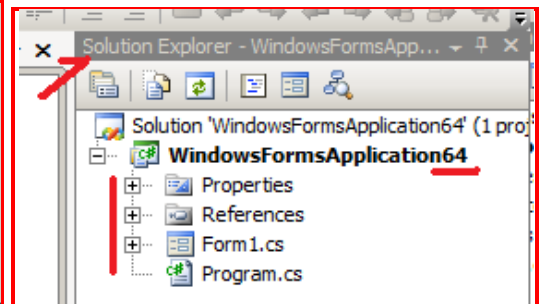
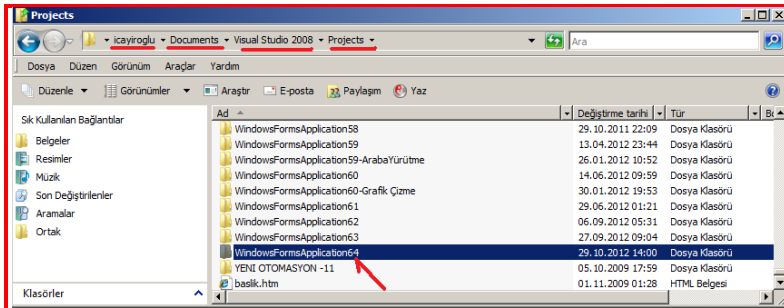


Karşımıza çıkan ekrandan C# dili seçili olmalı (1 nolu). Programımız masaüstü bir program olacağından ve Windows ortamında çalışan bir program olacağından "Windows Forms Application" seçili olmalı. Projenin adı 3 nolu yerde gösterilen addır. Bu projenin bilgisayarımızda nerede kayıtlı olacağını gösteren 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.

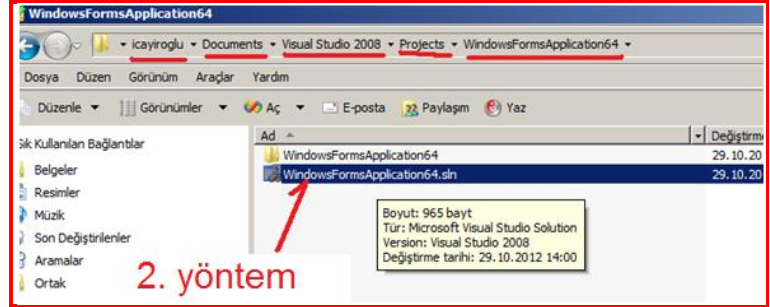
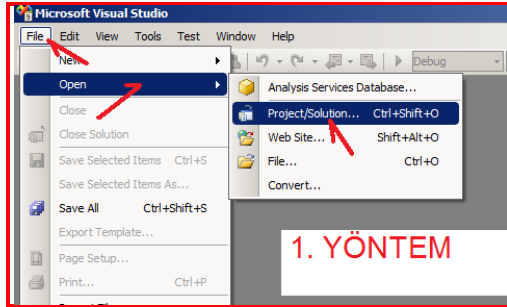


Ok düğmesine bastıktan sonra VS bize projemizin temelini oluşturan ilk yapıyı hazır olarak verecektir. Bunlarla ilgili ilk kodları projemizin içerisine atar ve bilgisayarımızdaki 4 numara ile gösterilen adresteki yere dosyaları kopyalar.

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.



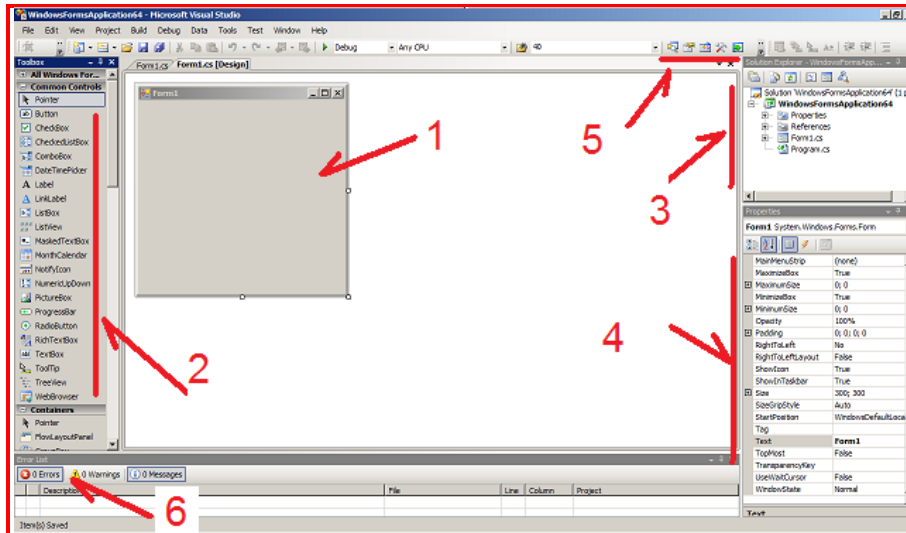
Projemizde epey bir çalışma yaptığımızı düşünelim. Ertesi günü tekrar projemizi açmak istersek iki yolu kullanabiliriz. Ya VS nin içerisinde File>Open>Project yolunu kullanıp buradaki 64 numaralı projeyi açarız. Yada Windows'un kendi gezgin penceresinden gidip 64 nolu projenin ana dosyasına çift tıklayıp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



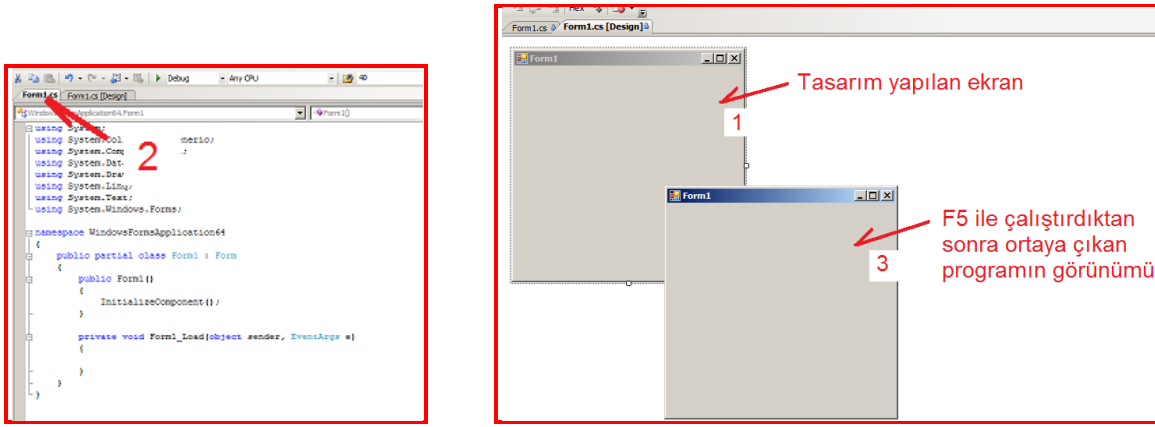
İlk projemizi New deyip oluşturduktan sonra karşımıza aşağıdaki gibi bir ekran gelecektir. Bu ekranda bize ilk olarak hazır bir nesne diyebileceğimiz Form nesnesi verilmiştir. Bu nesne programımızın zeminin oluşturduğu bir çerçevedir (1 nolu). Her nesnenin Özellikleri (properties) ve üzerinde gerçekleştirilebilecek 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 Properties penceresinde (4 nolu) ayarlayabiliriz.

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

Dikkat edersek ekranımızda birçok panel denilen pencereler bulunmaktadır. Bunları tekrar açıklarsak 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 gözükmemişse 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 içerisine 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 tipinde tek bir karakteri tutmak içindir. Tek tırnak içine yazılmalıdır (char sube='A') gibi			
	string	2x	Birden fazla karakteri tutmak içindir.			
	bool	1 (bit)	0 : 1 (false – true)			

Not: Ondalık sayıları tanımlarken üç tane tip tanımlıyoruz. Float daha küçük sayılar (10^{+38} gibi yine de büyük sayı). Bu sayıları tanımlarken sonun f harfi konulmalı. Yoksa double kabul eder. Double daha büyük sayıdır. (10^{+308} gibi çok büyük sayılar). Decimal ise daha hassas yani virgülden sonra hane sayısı 28 digite kadar giden bir sayı atarken kullanılır. Bunu tanımlarken sonuna m harfi konulmalıdır. Bir tanımlayı yaptığımızda üzerine gelip sağ tuşa tıklarsak değişkenin alabileceği üst ve alt sınırları sınıf kodları içinde görebiliriz.

```

int A = 10;
float B = 10.51f;
double C = 12.67;
decimal D = 13.78m;

public struct Single : IComparable, IFormattable, IConvertible
{
    public const Single MinValue = -3.40282347E+38F;
    public const Single Epsilon = 1.401298E-45F;
    public const Single MaxValue = 3.40282347E+38F;
    public const Single PositiveInfinity = 1F / 0F;
}

// Represents the largest possible value of System.Decimal. This field is
// and read-only.
public const Decimal MaxValue = 79228162514264337593543950335M;
public const Decimal MinValue = -79228162514264337593543950335M;

```

Aynı şekilde küçükten büyüğe doğru tam sayı tanımlamalarını da yazarsak

```

byte E = 123; //8 bit = 1 byte
short F = 12345; //16 bit = 2 byte
int G = 1234567890; //32 bit = 4 byte
long H = 1234567890123456789; //64 bit = 8 byte

```

Tanımlamaların başına “u” harfini (unsigned=işaretsiz) getirsek eksi bölgede sayılar artı bölge üzerine eklenir ve 0 dan itibaren iki kat daha büyük sayıları tutabilir (detaylar için yukarıdaki tabloyu inceleyin).

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 küsür sayılar arasındaki rakamları tutar

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

Nullable (null değer alabilir): String ve Char tipi değişkenler içerisine null (boş) alabilir. Yani başlangıçta herhangi bir şey atamazsak çalışır. Fakat sayısal ifadeler için başlangıçta içerisine 0 değerini atayabiliriz. Fakat yinede 0 da bir sayı olduğu için bunu atamak istemeyebiliriz. Bu durumda null değerini atmamız gerekir ama sayısal değişkenlerde bu hata verir. Bu değişkenlerin null değerini alabilmesi için nullable yapılması gerekir. Tanımın sonuna bir ? soru işareti konulmalı.

```

byte E = null;           byte? E = null;
short F = null;         short? F = null;
int G = null;           int? G = null;
long H = null;          long? H = null;

float B = null;         float? B = null;
double C = null;       double? C = null;
decimal D = null;      decimal? D = null;

char L = null;          char? L = null;

string K = null;        string K = null;

```

Bool gibi bir değişkeni cinsiyet için kullandığımızda true/false ifadelerini, erkek/kadın şeklinde değerlendirebiliriz. Eğer nullable yaparsak bu sefer cinsiyetini belirtmek istemiyor şeklinde de kullanabiliriz. Yani üç tane seçenek haline gelmiş olur.

MATEMATİK (Math) KÜTÜPHANESİ

Matematik hesaplamalarında kullanılanılabilecek 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);           // eb demektir
Math.Pow(b,c);         // bc 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=102 => 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

The screenshot shows a Windows application window titled "Form1". Inside the window, there is a button labeled "button1". To the right of the button are two text input fields. The first input field contains the text "Ali" and the second contains "SU". Below these input fields, there is a label that displays the text "merhaba Ali SU".

```
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;
}

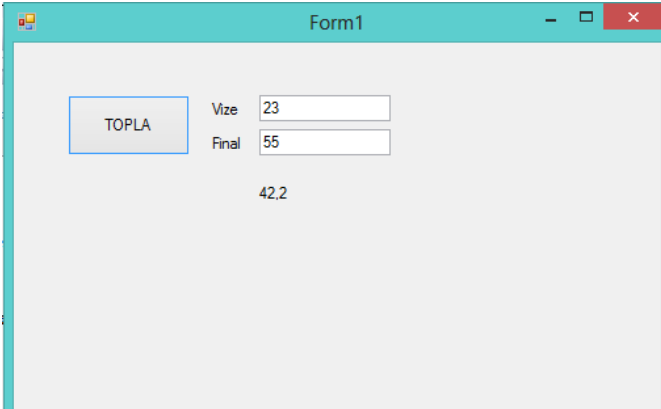
```

```

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

```

ÖRNEK



```

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

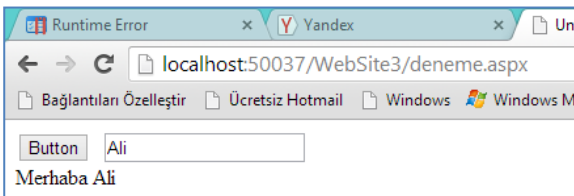
    Vize = Convert.ToInt32(textBox1.Text);
    Final = Convert.ToInt32(textBox2.Text);

    Ortalama = Vize * 0.40 + Final * 0.60;

    label1.Text = Ortalama.ToString();
}

```

ÖRNEK



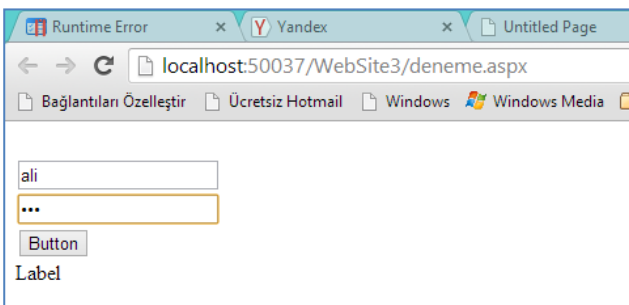
```

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

    Label1.Text = "Merhaba " + Ad;
}

```

ÖRNEK

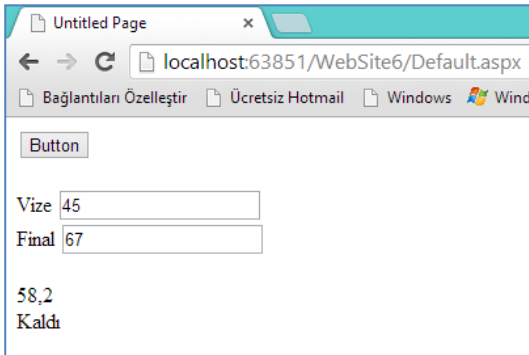


```
protected void Button1_Click(object sender, EventArgs e)
{
    string KullaniciAdi, Sifre;

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

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

ÖRNEK



```
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

Button

Ad Ali

Soyad SU

Bay

Bayan

Matematik

Fizik

Kimya

Bay Ali SU Matematik Kimya Seçtiniz

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

```

string Dersler = null;

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

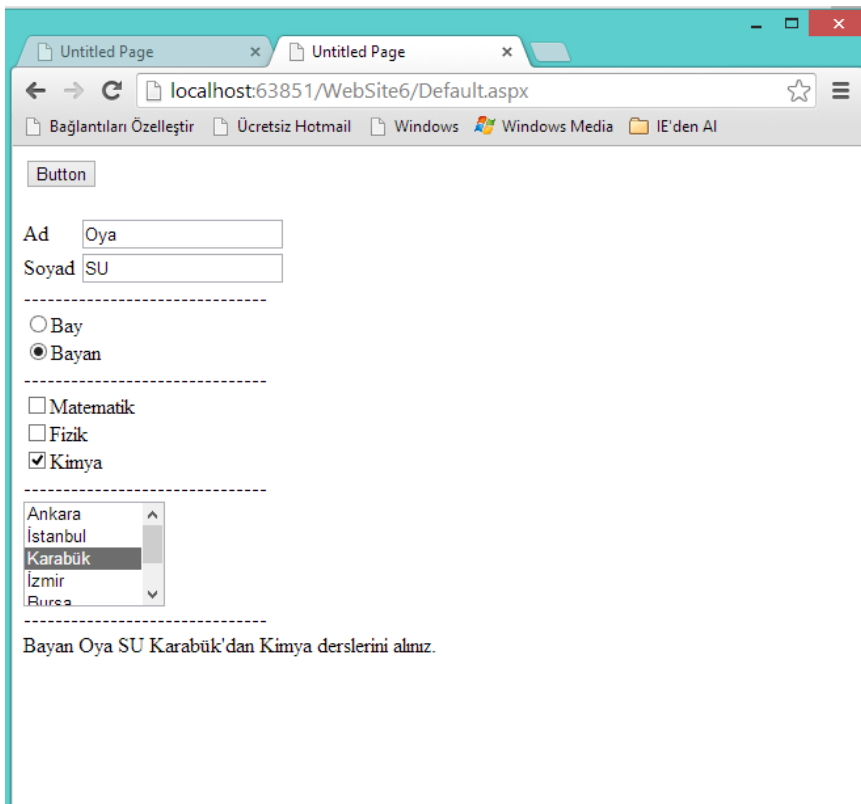
if (RadioButton1.Checked ==true)
{
    Cinsiyet = "Bay ";
}
else if (RadioButton2.Checked == true)
{
    Cinsiyet = "Bayan ";
}

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 ";
}

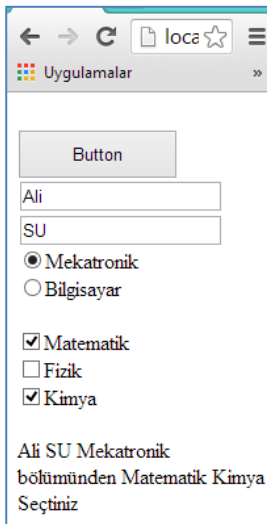
if (CheckBox1.Checked == true)
{
    Dersler =Dersler + " Matematik ";
}
if (CheckBox2.Checked == true)
{
    Dersler =Dersler + " Fizik ";
}
if (CheckBox3.Checked == true)
{
    Dersler =Dersler + " Kimya ";
}

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, Bolun=null, Dersler=null;

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

    if (RadioButton1.Checked == true)
        Bolun = "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

```
string Cinsiyet = null;
```

```
if (radioButton1.Checked == true)
    Cinsiyet = "Bay ";
else if (radioButton2.Checked == true)
    Cinsiyet = "Bayan ";
```

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

```
int SiraNo = listBox1.SelectedIndex;
string Bolum = listBox1.Items[SiraNo].ToString();
```

```
label4.Text = Cinsiyet + " " + Ad + " " + Soyad + " " + Bolum + " e kaydoldunuz!";
```

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 = "";
        }
    }
}

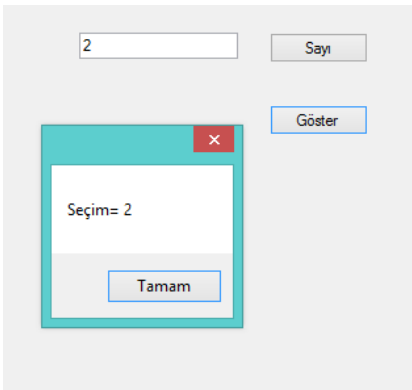
```

Switch Case Yapısı

Bu yapı If yapısı gibi koşullu durumlar için kullanılabilir. Burada koşulu sağlarken sadece eşit olma durumuna bakar. Büyük yada Küçük gibi koşulları kabul etmez. Değişken olarak kullanılacak ifadenin sadece integer olması gerekir. Yani blok yapıda kullanılacak değişkenini içeriği 1,2,3... gibi integer sayılar olmalıdır. Şu şekilde düşünebiliriz. Değişkenimizin içindeki değer 1 eşitse şu işlemi yap, 2 ye eşitse şu işlemi yap, gibi durumlar için kullanılır. Yapısı aşağıdaki şekildedir.

```
switch (Sayi)
{
    case 1:
        ....
        break;
    case 2:
        ....
        break;
    case 3:
        ....
        break;
    default:
        ....
        break;
}
```

Örnek:



```
int Sayi = 0;
```

```
private void button1_Click(object sender, EventArgs e)
{
    Sayi = Convert.ToInt32(textBox1.Text);
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    switch (Sayi)
    {
        case 1:
            MessageBox.Show("Seçim= " + Sayi.ToString());
            break;
        case 2:
            MessageBox.Show("Seçim= " + Sayi.ToString());
            break;
    }
}
```

```

    case 3:
        MessageBox.Show("Seçim= " + Sayi.ToString());
        break;
    default:
        MessageBox.Show("Geçersiz bir değer girdiniz" + Sayi.ToString());
        break;
    }
}

```

Get ve Set Metodları

set ve get metodunun görevi kısaca budur.. private tanımlanmış değişkenlere ulaşmayı sağlar..

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace set_get
{
    //dortgen clasının boy ve en özelliklerini tanımladık
    class dortgen {
        //private: oluşturulan değişkenlere sadece o class içinde ulaşılabilir.
        private int mBoy;
        private int mEn;

        //değişkene değer atama işlemleri
        //meodumuzu oluşturulımkı ve private nesnelermize değer atayabilelim
        public int En {
            get {
                return mEn;
            }

            //önce set metodu çalışacak. sonra get metodu ile değişken geri döndürülecek.
            set {
                mEn = value;
            }
        }

        //boy için metot oluşturulım
        public int boy{
            get {
                return mBoy;
            }
            set {
                mBoy = value;
            }
        }
    }

    class Program
    {
        static void Main(string[] args)

```

```

{
//nesnemizi oluşturalım
dortgen d1 = new dortgen();
//ve artık private olan değişkenlerimize değer atamaları yapabiliriz.. ve onlara istediğimiz gibi set
ve get metodu ile ulaşabiliriz.
d1.En = 52;
d1.boy = 152;

//ekrana yazdıralım
Console.WriteLine("En : " + d1.En);
Console.WriteLine("Boy : " + d1.boy);
Console.ReadLine();
}
}
}

```

Ödev 1: Pi sayısının hesaplanması

Pi Sayısını Hesaplayan Programı Yazın.

Pi Sayısı Nedir?

Bu sayı aslında bir orandır ve dairenin çevresinin çapına bölümünden elde edilir. Çapı 1 olan bir dairenin çevresi pi sayısına eşittir. Pi sayısının nasıl ve kim tarafından bulunduğu kesin olarak bilinmemektedir. Bunun sebebi pi sayısının farklı devirlerde farklı milletler tarafından kullanılmasıdır.

Pi sayısı tarih boyunca kaç alınmıştır. M.Ö. 2000 yılı dolaylarında Babiller 3,125; Antik Mısırlılar ise 256/ 81 yani yaklaşık 3,1605' i kullanmaktaydı. Eski Yunanda 3,162 sayısı kullanıldı. Archimedes ise (M.Ö. 287- 212) 3.10/71 ve 3.1/7 sayısını Pi sayısı olarak kullanmıştır. M.S. 500 yılı civarında Pi sayısı için 3, 1415929 olarak kullanılıyordu. 1424 yılında İran' da virgülden sonraki 16 basamağı doğru olarak biliniyordu. 1596 yılında Alman Ludolph van Ceulen, Pi' nin virgülden sonraki 20 basamağını hesapladı. O tarihten sonra Pi sayısının virgülden sonraki milyarlarca basamağı hesaplanmıştır.

Pi sayısını nasıl hesaplanacağına yönelik bir yöntem geliştirin ve bunu programla hesaplayın. Bakalım siz pi sayısını kaç haneye kadar bulabileceksiniz. Yöntem olarak örneğin daireyi belli açılarda dilime bölün. Her dilimi ikiz kenar üçgen olarak varsayın. Üçgenin dik yüksekliği yarıçapı, kısa kenarı daire üzerindeki bir uzunluğu ifade etsin. Buna göre ne kadar küçük dilime bölerseniz çevre ile çap oranını hesaplayabilirsiniz. Buda zaten Pi sayıdır.

3.14

159265358979323846264338327
950288419716939937510582097
49445923078164062862089986
28034825342117067982148086513282
30664709384460955058223172535940812848
1117450284102701938521105559644622948954
9303819644288109756659334461284756482337867831652712
01909145648566923460348610454326648213393607260249
1412737245870066063155881748815209209628292540917153
6436789259036001133053054882046652138414695194151160
9433057270365759591953092186117381932611793105118548074462379962749
567351885752724891227938183011949129833673362440656643086021394946
39522473719070217986094370277053921717629317675238467481846766940
513200056812714526356082778577134275778960917363717872146844090122
4953430146549585371050792279689258923542019956112129021960864034
41815981362977477130996051870721134999999837297804995105973173281609631859
502445945534690830264252230825344685035261931188171010003137838752886587532083814206
17177669147303598253490428755468731159562863882353787593751957781857780532
1712268066130019278766111959092164201989380952572010654858632788659...

Ödev 2: "e" Sayısının hesaplanması

e Sayısını hesaplayan Programı yazın. Yazdığınız program 10 dakika çalışarak, bulduğunuz en hassas e sayısını tespit edin. Bu sayıyı n=kaç için buldunuz gösteriniz.

e Sayısı Nedir?

Jakob Bernoulli, e sabitini birleşik faiz problemini incelerken keşfetmiştir. Daha sonra bu sayıyı euler ilan etmiş ve isminin baş harfi ile anılmaktadır.

Basit bir örnekle anlatılabilir. Elinde 1 lirası olan bir yatırımcı, parasını yılda %100 faiz veren bir bankaya yatırırsa, bir sene sonra 2 lirası olacaktır. Diğer yandan bu yıllık faiz %50 – %50 şeklinde yılda iki kez işlerse, yatırımcının yıl sonundaki parası $(1 + \frac{1}{2})^2 = 2,25$ lira olacaktır. Benzer şekilde eğer faiz yılda dört kez %25 oranında işlerse, yatırımcının yıl sonundaki parası $(1 + \frac{1}{4})^4 = 2,44141...$ lira olacak, faiz her ay %8,333... oranında işlerse yıl sonundaki para $(1 + \frac{1}{12})^{12} = 2,6130...$ lira olacaktır. Faizin işleme süresini daha da kısaltırsak, her hafta işleyen faiz (yılda 52 kez) yıl sonunda 2,6925... lira, her gün işleyen faiz (yılda 360 kez) yıl sonunda 2,71453... lira verecektir. Faizin işleme süresi kısaltıkça, yıl sonundaki para 2 ve 3 arasında belli bir değere yakınsamaktadır. e sayısının yakınsaması n bağlı olarak aşağıdaki şekilde olacaktır.

n	$(1+1/n)^n$
1	2
2	2,25
3	2,37037
4	2,44141
5	2,48832
10	2,59374
50	2,69159
100	2,70481
1000	2,71692
1.000.000	2.718280469219

Ödev 3: Mükemmel sayının hesaplanması

Mükemmel sayıları hesaplayan programı yazın. Programınız sistem saatini okusun. 10 dakika hesap yaptıktan sonra kaç tane mükemmel sayı bulursa onları Listbox'a yazsın.

Mükemmel Sayı nedir? 6,28,496 gibi kendisi hariç bütün pozitif çarpanlarının toplamı kendisine eşit olan sayılara denir. Mükemmel sayılar sonsuz sayıdadır. Genel formülleri henüz bulunamamıştır. Ancak n=çift sayı olmak üzere $2^n(2^{n+1}-1)$ formülünün mükemmel sayı olduğu görülebilir. Bu formül tüm mükemmel sayılar için ortak formül değildir. Günümüze kadar 44 tane mükemmel sayı bulunmuştur. Hepsi ya 6 ile yada 8 ile bitmektedir. İlk 4 mükemmel sayı şunlardır {6,28,496,8128,...}. Geri kalanı siz bulmaya çalışın. Kaç 10 dakika hesaplama ile kaç tane bulabileceksiniz araştırın.

$$6 \left| \begin{array}{l} 6 \\ 3 \\ 2 \\ 1 \end{array} \right. \} = 6$$

$$28 \left| \begin{array}{l} 28 \\ 14 \\ 7 \\ 4 \\ 2 \\ 1 \end{array} \right. \} = 28$$