

1-) c sharp'ta protected deklarasyonunu class yapısı içinde tasarlayarak örnek üzerinde gösteriniz. 20p

```
using System;
```

```
namespace aa
```

```
{
```

```
    public class X
```

```
    {
```

```
        protected string protectedOdev;
```

```
        public string publicOdev;
```

```
    }
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        var x = new X();
```

```
        var s = x.publicOdev; // publicOdev public olduğu için kullanabiliyoruz
```

```
        var s2 = x.protectedOdev; // protectedOdev protected olduğu için erişilemiyor
```

```
    }
```

```
}
```

```
}
```

2-) Aşağıdaki kodun sonucunu classın yapısını anlatarak yazınız.(c sharp ta) 20p

```
1 başvuru
class dairebaskanligi
{
    public string dairebaskani = "Dr. Emre Can";
    0 başvuru
    public string dairebaskanikim()
    {
        return dairebaskani;
    }
}
2 başvuru
class genelmudurluk : dairebaskanligi
{
    public string genelmudur = "Dr. Ali Can";
}
```

Burda dikkat etmemiz gereken genelmudurluk classı çağrıldığı halde daire başkanlığı classının da çalışması, daire başkanını Dr Emre Can olarak atamasıdır. Bunun sebebi ise class genelmudurluk : dairebaskanligi yazdığımız kısımda C# da inheritance adını verdiğimiz kalıtım işlemini gerçekleştirmemiz. Kalıtım işlemi kalıtım yapılan classın, kalıtım yapıldığı classtaki özellikleri de kullanabilmesine yarıyor. Bu sayede genelmudurluk içerisinde hem daire başkanlığında olan değerlere ve fonksiyonlara hem de daire başkanlığı içerisindeki erişebiliriz.

3-) c sharp'ta Interface içinde iki tane metod deklare ederek onu kalıtım alan sınıf (class) içinde kodlayınız. Ayrıca uygulamada bu metotları çalıştırınız. 20p
using System;

```
namespace aa
{

    interface A
    {
        void dosyaOku();
        void dosyaYaz();
    }

    class Dosya:A
    {
        void A.dosyaOku()
        {
            Console.WriteLine("Dosya okunuyor..");
        }
        void A.dosyaYaz()
        {
            Console.WriteLine("Dosyaya yaziliyor...");
        }
    }

    public class Program {
        public static void Main()
        {
            A dosya = new Dosya();

            dosya.dosyaOku();
            dosya.dosyaYaz();
        }
    }

}
```


4-) Liste içerisinde yer alan sayıların toplamını bulup messagebox kutusu içine yazdıran programı kullanıcı tanımlı sınıf içinde tanımlanan metotla gerçekleştiriniz.(sınıfın kendisi ile nesnesi modellenecek) 20p

```
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 WindowsFormsApp1
{
```

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

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

```
        toplama a = new toplama();
        int toplam = a.toplamalslemi();
        String b = string.Join(" ", a.numaralar);
```

```
        MessageBox.Show("Listedeki sayılar: " + b + "\nListenin içindeki değerler toplamı: " +
        toplam.ToString());
```

```
    }
}
```

```
public class toplama {
    public List<int> numaralar = new List<int>();
```

```
    public int toplamalslemi()
    {
        numaralar.Add(31);
        numaralar.Add(62);
        numaralar.Add(93);
        numaralar.Add(15);
```

```
int toplam = numaralar.Sum(x => Convert.ToInt32(x));  
return toplam;  
}
```

```
}  
}
```

5-) İçerisinde yapıcı metot, iki tane alan değişkeni ve bir adet kullanıcı tanımlı metot barındıran bir static sınıf tasarlayınız. Uygulama içerisinde (button un click olayı olabilir) bu sınıfa erişerek istediğiniz şekilde işletiniz. 20p

```
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 WindowsFormsApp1
{
    public static class ogrenci
    {
        static String isimSoyisim;
        static int ogrencild;

        static ogrenci()
        {
            isimSoyisim = "Osman Cakici";
            ogrencild = 157842354;
        }

        public static void bilgiler()
        {
            Console.WriteLine("Ogrenci ismi : " + isimSoyisim + " Ogrenci Numarası : " + ogrencild);
            MessageBox.Show("Ogrenci ismi : " + isimSoyisim + " Ogrenci Numarası : " + ogrencild);
        }
    }
}

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

```
}

private void Form1_Load(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
    ogrenci.bilgiler();
}
}
}
```

Buton1'e basıldığında :