

**Soru 1** Aşağıdaki işlemlerde hatanın ne olduğunu ve düzeltmek için ne olması gerektiğini boş bırakılan alana (yanına) yazınız: Atamadan sonra bilgi kaybı olmaması gerekmektedir! (**2 \* 5 = 10 puan**)

**a)**

byte i;

i= 1234; ..... (byte)1234; // Tam sayı literal int varsayılır, int değer daha dar kapsamlı byte tipi değişkene atanamaz. Cast edilerek byte olarak alınması sağlanır.

**b)** int x= 2500000000; ..... long x= 2500000000L; // Tam sayı literal int varsayılır ancak verilen değer int için büyük bir sayıdır, bu sayının long integer olarak alınması için sonuna L eklenir, bu nedenle bu sayı long tanımlanmalıdır. Int' e cast etmek doğru değildir çünkü bilgi kaybı olur.

**c)**

long c1= 5000000000;

int d1= c1; ..... long d1= c1;

**d)** int as= 5.5; ..... double as= 5.5; // Ondalık sayı literaller varsayılan olarak double kabul edilir, double değişkende tutulmalıdır.

**e)** String harf= 'A'; ..... char harf= 'A'; // Tek bir harf karakter tipinde tutulur. Tek tırnak da tek karakter gösterir.

**Soru 2** Soru işareti koyulmuş değişkenlerin son değerleri ne olacaktır yanlarına yazınız. (**20 puan**)

**a) 2 puan**

int s1, s2, s3;

s1= 15; s2= 20;

s3= s1 + s2;

s3= ? 35

**b) 3 puan**

char karakter1= 'A';

karakter1++;

karakter1++;

karakter1= ? C

**c) 3 puan**

boolean b1= false;

b1= b1 || (50 > 49);

b1= ? true

**d) 3 puan**

int x= 55 % 4;

boolean y= (x % 2)==0;

y= ? false

**e) 3 puan**

String a1= "Adana";

String a2= "Türkiye";

String a3= a1+a2;

a3= ? "AdanaTürkiye"

**f) 3 puan**

int z= (int)2.789;

z= ? 2

**g) 3 puan**

double a= 3.1459;

double b= 8/ (int) a;

b= ? 2.0

**Soru 3** Aşağıda verilen dizinin elemanlarının ters dizilmesiyle oluşan yeni bir dizi yaratınız. Eski ve yeni diziyi aşağıdaki gibi ekrana yazdırınız:

Dizi 1:

45.0    80.0    10.0    14.0    15.0    9.0

Yeni Dizi:

9.0    15.0    14.0    10.0    80.0    45.0

```
public static void main(String[] args){
    double[] dizi1= {45,80,10,14,15,9};
    int uzunluk= dizi1.length;

    System.out.println("Dizi 1:");
    for (int i= 0; i<uzunluk; i++){
        System.out.print(dizi1[i]+ " ");
    }
    System.out.println();

    double[] yeniDizi= new double[uzunluk];
    // Yöntem 1: q isimli bir yardımcı sayaç kullanılıyor
    int q= 0;
    for (int k= yeniDizi.length-1; k>=0; k--){
        yeniDizi[k]= dizi1[q];
        q++;
    } // end
    System.out.println("Yeni Dizi:");
    for (int i= 0; i<uzunluk; i++){
        System.out.print(yeniDizi[i]+ " ");
    }
}
```

**Soru 4** Aşağıda verilen matris ile ilgili aşağıdaki kod parçalarını yazınız: (Main metodunun içinde yazıyorsunuz, koddan başlayın)

```
10    15    18    19
33    88    37    30
61    16    90    5
```

a) Yukarıdaki matrisi tutacak "matrisim" isimli değişken tanımlayınız ve başlangıç değeri olarak tek satırda yukarıdaki değerleri atayan kod satırını yazınız.

```
int[][] matrisim= {{10, 15, 18, 19}, {33, 88, 37, 30}, {61, 16, 90, 5}};
```

b) Yarattığınız matrisin en sağdaki sütunundaki elemanlarını ekrana yan yana yazdıran kod parçasını yazınız.

```
for(int satir= 0; satir< matrisim.length; satir++){
    System.out.print(matrisim[satir][3] + " ");
}
System.out.println();
```

c) Main içerisinde aşağıdaki gibi çağırılan metodu yazınız. Bu metod gönderilen matrisin istenen satırının en büyük elemanını bulup gönderiyor.

```
int satir= 2;
int enbuyuk= enBuyukBul(matrisim, satir);
System.out.println("En buyuk eleman: "+ enbuyuk);
```

```
private static int enBuyukBul(int[][] mat, int i) {
    int enb= mat[i][0];
    for (int k=1; k< mat[i].length; k++)
        if (mat[i][k] > enb)
            enb= mat[i][k];
    return enb;
}
```

**Soru 5** Aşağıdaki kod parçasının ekran çıktısının ne olacağını yanına yazınız

```
int[] dizi= {2, 14, 8, 70, 5, 49, 78, 11, 12};
int indis= 0;
int sayac= 1;
boolean diziSonu= false;
```

```
while(!diziSonu){
    System.out.println();
    for (int k=0; k< sayac; k++){
        System.out.print(dizi[indis]+" ");
        indis++;
        if (indis>= dizi.length){
            diziSonu= true;
            break;
        } // end if
    } // end k for
    sayac= sayac+2;
} // end while
```

```
2
14    8    70
5    49    78    11    12
```

**Soru 6** Bir Class tasarımı yapmanız istenmektedir. Senaryo şu şekildedir: Bir işyerinde müdür, takım lideri ve programcı olmak üzere 3 değişik çalışan vardır. Bütün çalışanlar bir nesne olarak bir veri tabanında saklanacaklardır. Çalışanların ortak olarak adsoyad, doğum tarihi ve mezun oldukları bölüm bilgileri tutulmakta, bunlara ek olarak programcıların bildiği dil sayısı; takım liderinin ise liderlik ettiği takımın ismi bilgisi; müdürün ise maaş bilgisi tutulmak istenmektedir.

a) Buna göre ortak özelliklerin mutlaka bütün çalışanlarda tutulacağı bir Class yapısı tasarlayınız. Base class, abstract class, türeyen Class kullanmak gerekir mi, her bir class içinde hangi bilgiler tutulmalı... Tasarladığınızı classların tamamının kodunu yazınız.

```
// Calisan Base class olmalı
// Bu class' tan direkt obje yaratılmayacağı için (her çalışan ya müdür ya takım lideri ya da programcı olmalı) abstract olmalı
public abstract class Calisan {
    public String adSoyad;
```

```

        public String dogumTarihi; // int olarak yıl tutanlar da olabilir
        public String mezuniyetBolumu;
    }
    public class Mudur extends Calisan {
        public long maas;
    }
    public final class TakimLideri extends Calisan {
        public String takimIsim;
    }
    public class Programci extends Calisan {
        public int programlamaDilleri;
    }
}

```

**b)** İşyeri için yaratılan toplam çalışan nesnesi sayısı tutulmak isteniyor. Hangi class içinde nasıl tutulmalı?

```

public abstract class Calisan {

    public static int calisanSayisi= 0;

    public String adSoyad;
    public String dogumTarihi; // int olarak yıl tutanlar da olabilir
    public String mezuniyetBolumu;

    // Base class constructor türeyen classlardan bir object yaratılınca otomatikman çalışır:
    public Calisan(){
        calisanSayisi++;
    }
}

```