



JAVA QUESTION BANK

2022



TechPro Education LLC.

JAVA Questions Bank

Copyright © TechPro Education LLC, 2022

All publication rights of this work belong to TechPro Education LLC.

All right reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system without permission in writing from TeachPro Education.

ISBN : 979-8-9862229-2-9

www.techproeducation.com

Published: September 2022

PREFACE

What is JAVA?

Java is a versatile and powerful programming language that enables developers to create robust, high-performance applications. Java is platform-independent, meaning that programs written in Java can run on any operating system. Java programming language is a high-level, object-oriented, and general-purpose computer programming language. Using java you can write computer applications, games, and utility software. Programming and coding with Java is an excellent way to develop your skills in coding and programming. Java is a versatile language that can be used for developing a wide range of applications. In addition, Java's Just-In-Time compiler makes it possible to run Java programs at high speed. Java is an excellent choice for developing a wide variety of applications.

Why should I learn JAVA?

Java is easy to learn and use, making it an ideal choice for beginners. Programming and coding with Java will allow you to develop applications for a variety of platforms, including Windows, Mac, Linux, and Android.

As you become more proficient in coding with Java, you will be able to develop increasingly complex applications. In addition, you will be able to take advantage of the many features that Java offers, such as its object-oriented programming model and its rich set of libraries.

With its versatility, ease of use, and rich set of features, Java is an ideal choice for anyone interested in coding and programming.

About the book

Mastering Java can be challenging, especially for beginners. This question bank will help you hone your Java skills, whether you're just starting or looking to take your knowledge to the next level.

ABOUT THE AUTHOR

Mr. Alptekin has more than 10 years of teaching experience and touched many people's lives by teaching development courses.

Mr. Alptekin is a very passionate instructor. He is currently teaching development classes such as APEX, SOQL, JAVA at TechPro Educations for future developers.

Mr. Alptekin holds Bachelor's in Mathematics and an Associate degree in Computer Science. He has Oracle JAVA, AWS, and Scrum Mater certified. He is also a high-rated Udemy instructor. Mr. Alptekin is one of the TechPro Education co-founders as well.

CONTENTS

PART 1 - TESTS

Introduction to Java	9
Variables	12
Variables Wrappers	16
Increment Decrement Concatenation	19
Scanner	23
Type Casting	26
If Else - If Elseif	30
NestedIf Ternary	34
Nested Ternary Switch	39
String Methods	46
String Methods For Loop	50
For Loops	54
While Loops	59
Do While Loop	64
Instance Local Variables Method Call	69
Constructors	74
Static Keyword	79
Static Block Arrays	88
Arrays	94
Multi Dimensional Arrays	99
Array Lists	104
For Each Loops	110
Date Time	116
Varargs Access Modifiers	122
String Builder	128
Encapsulation	134
Inheritance	142
Inheritance Constructor Call	146
Overriding	155
Overloading-Overriding	168
Exceptions - 1 - 1	176
Exceptions - 2	185

CONTENTS

Abstract Classes	195
Interface - 1	203
Interface - 2	210
Iterators.....	219
Collections 1	226
Collections 2	232

PART 1 - APEX QUESTIONS

Variables	239
Type Casting	241
Wrapper Classes.....	242
String Manipulations	243
Date Class.....	249
Date - Time Class	252
If Statement	254
Ternary Statement.....	261
Switch-On Statement.....	264
Loops	274
Arrays.....	299
Lists.....	307
Sets.....	316
Maps.....	319
Answer Keys.....	323



TESTS

1. Java'da objelerle ilgili aşağıdakilerden hangisi yanlıştır?

- A) Objelerin özellikleri ve işlevleri olur.
- B) Özellikler Fields, işlevler method olarak adlandırılır.
- C) Objeler Class'lar kullanılarak oluşturulur.
- D) Objeler Class'lar için birer kalıptır.

2. Java'da Class'lar için aşağıdakilerden hangisi yanlıştır?

- A) Class isimleri büyük harf ile baslar.
- B) Class'lar kalıp gibi kullanılarak objeler üretilir.
- C) Class'larda da Fieldlar ve Methodlar vardır.
- D) Class isimlerinden sonra () kullanılır.

3. Java'da Access Modifier'larla ilgili aşağıdakilerden hangisi yanlıştır?

- A) Giriş düzenleyici demektir. Oluşturulan bir Class'a kimin ulaşip kimin ulaşamayacağını düzenlemeye yarar.
- B) Java'da 4 tane Access Modifier vardır.
- C) Bir Class'ın Access Modifieri "public" olursa, o Class'a herkes ulaşabilir demektir.
- D) Class oluştururken Access Modifierlar Class isminden hemen sonra yazılır.

4. Aşağıdakilerden hangisi bir Class oluşturmak için doğru yazılan bir kod'dur?

- A) `public class MyFirstClass{ }`
- B) `class public MyFirstClass{ }`
- C) `MyFirstClass public class{ }`
- D) `public class myFirstClass{ }`

5. `public int myFirstMethod(){ }`

Yukarıda yazılan kod ile ilgili aşağıdakilerden hangisi yanlıştır?

- A) Bir method oluşturmak için yazılmıştır.
- B) "int" return type olarak adlandırılır ve method'un ürettiği sonuçun tipinin tamsayı olduğunu ifade eder.
- C) Method ismi küçük harfle başlatılarak hata yapılmıştır.
- D) "{" ile başlayıp "}" ile biten kısım Method Body olarak isimlendirilir.

6. **Aşağıdakilerden hangisi main method oluşturmak için yazılan koddur?**

- A) `public static void main(String[] args) { }`
- B) `public static void Main(String[] args) { }`
- C) `public void static main(String[] args) { }`
- D) `public static void main(String() args) { }`

7. `public static void main(String[] args) { }`

Yukarıda verilen kod ile ilgili aşağıdakilerden hangisi yanlıştır?

- A) Java programlarının başlangıç noktasıdır.
- B) Main Method olarak adlandırılır.
- C) Bir Java programında main method olmazsa o program çalıştırılmaz.
- D) Hiçbiri.

8. Java'da bir method 1.2 ile 4.7'i topluyor.

Aşağıdakilerden hangileri doğrudur?

- A) Bu method 5.9 return eder
- B) Bu method'un return type'i ondalıklı sayıdır.
- C) Bu methodun ismi küçük harfle başlamalıdır.
- D) Bu method'un access modifier'i public olmak zorundadır.

9. **Aşağıdaki eşleştirmelerden hangisi yanlıştır?**

- A) fields = variables
- B) functions = methods
- C) entry point = main method
- D) void = access modifier

10. **Aşağıdakilerden hangileri doğrudur?**

- A) OOP Object Oriented Programming demektir
- B) Integrate parçaları bir araya getirmek demektir.
- C) Java Code'larını 0 ve 1 lere çevirmek Compile demektir.
- D) 0 elektrik var, 1 elektrik yok demektir.

1. Hangileri doğrudur?

- A) `System.out.print("Ali");`
`System.out.println("Can");`
Yukarıdaki ifade console'a AliCan yazdırır
- B) `System.out.println("Ali");`
`System.out.print("Can");`
Yukarıdaki ifade console'a AliCan yazdırır
- C) `System.out.print("Ali");`
`System.out.println("Can");`
Yukarıdaki ifade console'a
Ali
Can
yazdırır
- D) `System.out.println("Ali");`
`System.out.println("Can");`
Yukarıdaki ifade console'a
Ali
Can
yazdırır

- ## 2.
- 1) "class" kelimesi
 - 2) Class ismi
 - 3) Access Modifier
 - 4) { }

Class oluşturabilmek için yukarıdakilerin doğru sıralaması aşağıdakilerden hangisi gibi olmalıdır?

- A) 3 - 1 - 2 - 4
B) 3 - 2 - 1 - 4
C) 2 - 3 - 1 - 4
D) 4 - 2 - 1 - 3

3. 1) ()
2) Access Modifier
3) { }
4) Return Type
5) Method Ismi

Bir metod oluşturmak için yukarıdakilerin doğru sıralanışı hangisi gibi olmalıdır?

- A) 2 - 4 - 5 - 1 - 3
B) 2 - 5 - 4 - 1 - 3
C) 2 - 1 - 3 - 4 - 5
D) 3 - 4 - 5 - 1 - 2

4. Java'da toplama işlemi yapan bir method yazdık. Bu method 3 ile 5 tam sayılarını toplayarak bize sonucunu veriyor.

"Return Type" aşağıdakilerden hangisidir?

- A) 8
B) Tamsayı
C) 3 ile 5
D) Toplama

5. Ekran "Java Kolaydır" yazdırmak için aşağıdakilerden hangileri doğrudur?

- A) `System.out.println(Java Kolaydır);`
B) `System.out.println("Java Kolaydır")`
C) `System.out.println("Java Kolaydır");`
D) `System.out.print("Java Kolaydır");`

6. `int fiyat = 123;`

Aşağıdakilerden hangisi doğrudur?

- A) `System.out.print("fiyat");`
Ekran 123 yazdırır.
- B) `System.out.print(fiyat);`
Ekran 123 yazdırır.
- C) `System.out.print('fiyat');`
Ekran fiyat yazdırır.
- D) `System.out.print('fiyat');`
Ekran 123 yazdırır.

7. **Hangisi yanlıştır?**

- A) 8 byte 1 bittir.
- B) Memory'nin en küçük parçası bittir.
- C) Her boolean memory'de 1 bit yer kaplar.
- D) boolean'lar için iki farklı değer vardır.

8. **Java kodlarının arasına yorum cümlesi veya cümleleri eklemek isteyen biri aşağıdakilerden hangilerini yapabilir?**

- A) `// Ben bir yorum cümlesiyim Java beni okumaz`
- B) `/ Ben bir yorum cümlesiyim Java beni okumaz /`
- C) `/*`
Ben bir paragraftaki yorum cümlesiyim Java beni okumaz
Ben bir paragraftaki yorum cümlesiyim Java beni okumaz
Ben bir paragraftaki yorum cümlesiyim Java beni okumaz
`/*`
- D) `/*`
Ben bir paragraftaki yorum cümlesiyim Java beni okumaz
Ben bir paragraftaki yorum cümlesiyim Java beni okumaz
Ben bir paragraftaki yorum cümlesiyim Java beni okumaz
`*/`

9. Aşağıdaki kodlardan hangileri ekrana

Ali Can

216 1234567

Imza

yazdırır?

- A) `System.out.println("Ali Can");`
`System.out.println("216 1234567");`
`System.out.println("Imza");`
- B) `System.out.println("Ali Can");`
`System.out.println("216 1234567");`
`System.out.print("Imza");`
- C) `System.out.print("Ali");`
`System.out.println(" Can");`
`System.out.println("216 1234567");`
`System.out.println("Imza");`
- D) `System.out.print("Ali Can");`
`System.out.print("216 1234567");`
`System.out.print("Imza");`

10. `int x = 12;`

`int y = 13;`

Aşağıdakilerden hangileri yukarıdaki ile aynı anlamdadır?

- A) `int x;` `int y;`
`x = 12;` `y = 13;`
- B) `int x = 12, y = 13;`
- C) `int x = 12, int y = 13;`
- D) `int (x = 12, y = 13);`

1. Aşağıdakilerden hangisi yanlıştır?

- A) primitive data type'lar Java tarafından üretilmiştir. Bilgisayar programcıları yeni primitive data type'lar üretemezler.
- B) Bir ilçenin nüfusunu gösteren bir variable için "byte" data type kullanılabilir.
- C) Türkiye'de bir öğretmenin aylık maaşını gösteren bir variable için "short" data type kullanılabilir.
- D) Java bütün ondalık sayıları otomatik olarak "double" olarak kabul eder, eğer siz ondalık sayıyı "float" yapmak isterseniz sonuna "f" veya "F" koymalısınız.

2. long data type'ların alabileceği maximum değeri bulmak için aşağıdaki kodlardan hangisi yazılmalıdır?

- A) `System.out.println(Long.Max_Value);`
- B) `System.out.println(Long.MAX-VALUE);`
- C) `System.out.println(LONG.MAX_VALUE);`
- D) `System.out.println(Long.MAX_VALUE);`

3. Aşağıdakilerden hangileri doğrudur?

- A) `int harf = 'a';`
`System.out.println(harf);`
Ekrana a yazdırır.
- B) `char harf = 'b';`
`System.out.println(harf);`
Ekrana b yazdırır.
- C) `System.out.println('a' + 'b');`
Ekрана bir sayı yazdırır.
- D) `int harf = 'a';`
`System.out.println(harf);`
Kodu yanlış yazılmıştır, int data type'ına char atanamaz.

4. Aşağıdakilerden hangisi "non-primitive data type" lar için yanlıştır?

- A) String bir "primitive data type" dir.
- B) Bilgisayar programcılar istedikleri kadar "non-primitive data type" üretebilirler.
- C) "non-primitive data type" larin memory'de kapladıkları yer hep aynı büyüklüktedir.
- D) "non-primitive data type" lar memory'de değer yanında method da içerirler.

5. Hangileri doğrudur?

- A) Primitive'ler sadece value içerir, non-primitive'ler value and methodlar içerir.
- B) Primitive'ler küçük harf ile, non-primitive'ler büyük harf ile başlar.
- C) Primitive'leri Java oluşturdu biz primitive oluşturamayız.
Non-primitive'leri biz de oluşturabiliriz, Java da oluşturabilir.
- D) Primitive'lerin büyüklükleri data type'ing göre değişir, non-primitive'lerin hepsi aynı büyüklüktedir.

6. Aşağıdakilerden hangileri doğrudur?

- A) `System.out.println('b' > 'B');`
Console'a true yazdırır.
- B) `System.out.println('At');`
Console'a At yazdırır.
- C) `System.out.println('a' + 0);`
Compile Time Error verir.
- D) `char ch = "";` //Tek tirnak arasında hiçbirşey yok
`System.out.println(ch);`
Compile Time Error verir.

7. Aşağıdakilerden hangileri Compile Time Error verir?

- A) float f1 = 1.34;
- B) byte b1 = 723;
- C) String s1 = ""; //Cift tirnak arasında hiçbirsey yok
- D) double d1 = 5;

8. byte b1 = 10;

byte b2 = 20;

byte b3 = b1 * b2;

System.out.println(b3);

Yukarıdaki kod ile ilgili aşağıdakilerden hangisi doğrudur?

- A) Ekrana b3 yazdırır.
- B) Ekrana 200 yazdırır.
- C) Compile Time Error verir.
- D) Hiçbiri

9. Aşağıdakilerden hangileri Wrapper Class'lar için doğrudur?

- A) Her primitive data type için bir tane Wrapper Class vardır.
- B) Wrapper Class'lar non-primitive'dir.
- C) Wrapper Class'lar içlerinde variable değeri ve faydalı methodlar içerirler.
- D) Boolean, Char, Byte, Short, Int, Long, Float ve Double wrapper class'lardır.

10. String s1 = "Java Kolaydır";

System.out.println(s1.toUpperCase());

Yukarıdaki kodun ekran çıktısı nedir?

- A) Java Kolaydır
- B) Java Kolaydır
- C) JAVA Kolaydır
- D) Compile Time Error verir

1. `System.out.println(2+3x(4+5));`

Ekran çıktısı nedir?

- A) 29
- B) 45
- C) 19
- D) 23

2. `int sayı1 = 5;`

`int sayı2 = 2;`

`int sonuç = sayı1 / sayı2;`

`System.out.println(sonuç);`

Yukarıdaki kod çalıştırıldığında ekran çıktısı ne olur?

- A) 2.5
- B) 2
- C) 1
- D) Hiçbiri

3. `int sayı1 = 5;`

`int sayı2 = 2;`

`double sonuç = sayı1 / sayı2;`

`System.out.println(sonuç);`

Yukarıdaki kod çalıştırıldığında ekran çıktısı ne olur?

- A) 2.5
- B) 2.0
- C) 2
- D) Hiçbiri

4. `double sayı1 = 5;`
`double sayı2 = 2;`
`double sonuç = sayı1 / sayı2;`
`System.out.println(sonuç);`

Yukarıdaki kod çalıştırıldığında ekran çıktısı ne olur?

- A) 2.5
- B) 2.0
- C) 2
- D) Hiçbiri

5. `int num = 12;`

num isimli variable'in değerini 1 artırmak için aşağıdakilerden hangisi yanlıştır?

- A) `num = num + 1;`
- B) `num+=1;`
- C) `num++;`
- D) `num+=1;`

6. `int num = 12;` **num isimli variable'in değerini 2 azaltmak için aşağıdakilerden hangileri yapılabilir?**

- A) `num = num - 2;`
- B) `num-=2;`
- C) `num--;`
- D) `num-2=num;`

7. `int x = 12;`
`int y = 15;`
`x*=2;`
`y/=3;`
`x++;`
`y--;`
`System.out.println(x + y);`

Yukarıdaki programın ekran çıktısı nedir?

- A) 24
- B) 27
- C) 29
- D) 3

8. **Aşağıdakilerden hangisinin ekran çıktısı "true" değildir?**

- A) `System.out.println(2+3 == 5);`
- B) `System.out.println(2+3 >= 5);`
- C) `System.out.println(2+3 <= 5);`
- D) `System.out.println(2+4 != 5);`

9. `int g = 2`
`int h = 4;`
`String i = "A";`

Aşağıdakilerden hangileri doğrudur?

- A) `System.out.println((g*h) + [(h-g) + i] + (g-h));`
Console'a 82A-2 yazdırır
- B) `System.out.println((g*h) + (h-g) + i + (g-h));`
Console'a 10A-2 yazdırır
- C) `System.out.println((g*h) + (h-g) + (h/g) + i);`
Console'a 12A yazdırır

10. `int num1 = 4;`

`int num2 = 5;`

`String str1 = "Ali";`

Hangileri doğrudur?

- A) `System.out.println(str1 + num1 + num2);`
Console'a Ali45 yazdırır
- B) `System.out.println(num1 + str1 + num2);`
Console'a 4Ali5 yazdırır
- C) `System.out.println(num1 + num2 + str1);`
Console'a 45Ali yazdırır
- D) `System.out.println(str1 + [num1 + num2]);`
Console'a Ali9 yazdırır

1. `Scanner scan = new Scanner(System.in);`

Yukarıdaki kod parçası ile ilgili aşağıdakilerden hangisi yanlıştır?

- A) Scanner class'i kalıp olarak kullanılarak scan isimli bir object oluşturulmuştur.
- B) new kelimesi object oluştururken kullanılan bir keyword'dur.
- C) `Scanner scan = new Scanner(System.in);`
kodu yazıldıktan sonra kırmızı bir alt çizgi alırız, bu hata mesajıdır. Bu hatayı düzeltmek için Scanner class'ini java.util kütüphanesinden import etmemiz gerekir.
- D) Hiçbiri

2. `Scanner scan = new Scanner(System.in);`
`System.out.println("Lutfen bir sayı giriniz");`
`double sayı = scan.nextDouble();`
`System.out.println(sayı*sayı);`

Yukarıdaki program ile ilgili aşağıdakilerden hangileri yanlıştır?

Program kullanıcıdan sayı alır ve bu sayının karesini ekrana yazdırır.
Program ekrana sayı*sayı yazdırır.
Program hatalı yazılmıştır.
Kullanıcı tamsayı girerse program çalışmaz.

3. **Kullanıcının girdiği, sadece true veya false değerlerini içeren data'yi yazdığımız programdaki bir variable'a atayabilmek için hangi method kullanılmalıdır?**

- A) `nextBoolean()`
- B) `next()`
- C) `nextInt()`
- D) Hiçbiri

4. Kullanıcıdan bir byte data type'ında bir sayı alıp, o sayının 5 katını ekrana yazdırmak için aşağıdaki kodlardan hangileri yazılabilir?

- A)

```
Scanner scan = new Scanner(System.in);
System.out.println("Lutfen -128 ile 127 arasında bir sayı giriniz");
Byte sayı = scan.nextByte();
System.out.println(5*sayı);
```
- B)

```
Scanner scan = new Scanner(System.in);
System.out.println("Lutfen -128 ile 127 arasında bir sayı giriniz");
byte sayı = scan.nextByte();
System.out.println(sayı + sayı + sayı + sayı + sayı);
```
- C)

```
Scanner scan = new Scanner(System.in);
System.out.println("Lutfen -128 ile 127 arasında bir sayı giriniz");
int sayı = scan.nextInt();
System.out.println(5*sayı);
```
- D)

```
Scanner scan = new Scanner(System.in);
System.out.println("Lutfen -128 ile 127 arasında bir sayı giriniz");
double sayı = scan.nextDouble();
System.out.println(sayı + sayı + sayı + sayı + sayı);
```

- 5.**

```
Scanner scan = new Scanner(System.in);
System.out.println("Kare'nin bir kenar uzunluğunu giriniz");
double kenar = scan.nextDouble();
double alan = kenar*kenar;
System.out.println("Alan: " + alan);
```

Yukarıda verilen kod ile ilgili aşağıdakilerden hangisi doğrudur?

- A) Kullanıcının girdiği her değer için Ekrana Alan: alan yazdırır.
- B) Kullanıcı 5 girerse Ekrana Alan: 25 yazdırır.
- C) Kullanıcı 7 girerse Ekrana Alan: 49.0 yazdırır.
- D) Program çalışmaz

6. Hangisi yanlıştır?

- A) `char harf = scan.next().charAt(0);`
Kullanıcıdan char type'ında bir data almak için yukarıdaki kod kullanılır.
- B) `String isim = scan.next();`
Kullanıcıdan tek kelimelik bir data almak için yukarıdaki kod kullanılır.
- C) `String isim = scan.nextLine();`
Kullanıcının girdiği String type'ındaki tüm data'yi almak için yukarıdaki kod kullanılır.
- D) `char harf = scan.nextChar();`
Kullanıcıdan char type'ında bir data almak için yukarıdaki kod kullanılır.

7. Aşağıdakilerden hangileri doğrudur?

- A) `Scanner scan = new Scanner(System.in);`
`System.out.println("Ad ve soyadınızı giriniz");`
`String isim = scan.nextLine();`
`System.out.println(isim);`
Kullanıcı Ali Can girerse ekrana Ali Can yazar
- B) `Scanner scan = new Scanner(System.in);`
`System.out.println("Ad ve soyadınızı giriniz");`
`String isim = scan.next();`
`System.out.println(isim);`
Kullanıcı Ali Can girerse ekrana Ali yazar
- C) `Scanner scan = new Scanner(System.in);`
`System.out.println("Ad ve soyadınızı giriniz");`
`String isim = scan.next();`
`System.out.println(isim);`
Kullanıcı 12,345 girerse ekrana 12,345 yazar
- D) `Scanner scan = new Scanner(System.in);`
`System.out.println("Ad ve soyadınızı giriniz");`
`String isim = scan.nextLine();`
`System.out.println(isim);`
Kullanıcı true false girerse ekrana true false yazar

1. `System.out.println(2+3x(4+5));`

Ekran çıktısı nedir?

- A) 29
- B) 45
- C) 19
- D) 23

2. `Scanner scan = new Scanner(System.in);`
`System.out.println("Adınızı giriniz");`
`String isim = scan.next();`
`System.out.println(isim);`

Yukarıdaki kod çalıştırıldığında kullanıcı Ali Can girerse ekran çıktısı ne olur?

- A) Ali Can
- B) Ali
- C) Can
- D) Hiçbiri

3. **next() ile nextLine() arasındaki farklar aşağıdakilerden hangileridir?**

- A) next() ilk space karakterine kadar olan String'i alır, sonrasını almaz.
- B) nextLine() kullanıcının girdiği tüm String'i alır.
- C) next() char data type'lar için, nextLine() String data type'lar için kullanılır.
- D) next() ve nextLine() aynı işi yapar.

4. `int sayı1 = 5;`
`int sayı2 = 2;`
`int sonuç = sayı1 / sayı2;`
`System.out.println(sonuç);`

Yukarıdaki kod çalıştırıldığında ekran çıktısı ne olur?

- A) 2.5
- B) 2
- C) 1
- D) Hiçbiri

5. `int sayı1 = 5;`
`int sayı2 = 2;`
`double sonuç = sayı1 / sayı2;`
`System.out.println(sonuç);`

Yukarıdaki kod çalıştırıldığında ekran çıktısı ne olur?

- A) 2.5
- B) 2.0
- C) 2
- D) Hiçbiri

6. `double sayı1 = 5;`
`double sayı2 = 2;`
`double sonuç = sayı1 / sayı2;`
`System.out.println(sonuç);`

Yukarıdaki kod çalıştırıldığında ekran çıktısı ne olur?

- A) 2.5
- B) 2.0
- C) 2
- D) Hiçbiri

7. "Auto Widening" nedir?

- A) Küçük data tipini büyük data tipine atarsak Java otomatik olarak küçük data tipini büyük data tipine çevirir. Bu işleme "Auto Widening" denir.
- B) Büyük data tipini küçük data tipine atarsak Java otomatik olarak büyük data tipini küçük data tipine çeviremez. Bunu biz yapmalıyız. Bizim yaptığımız bu işleme "Auto Widening" denir.
- C) boolean data type'indeki bir variable'in char data type'indeki bir variable'a atanmasına "Auto Widening" denir.

8. Aşağıdakilerden hangisi "Auto Widening"e örnektir?

- A) `byte sayı1 = 12;`
`int sayı2 = sayı1;`
- B) `long sayı1 = 234;`
`short sayı2 = sayı1;`
- C) `int sayı1 = 57;`
`int sayı2 = sayı1;`
- D) `short sayı1 = 1232;`
`byte sayı2 = (byte)sayı1;`

9. Aşağıdakilerden hangisi "Explicit Narrowing"e örnektir?

- A) `int sayı1 = 987;`
`int sayı2 = sayı1;`
- B) `short sayı1 = 127;`
`byte sayı2 = (byte)sayı1;`
- C) `long sayı1 = 732;`
`short sayı2 = sayı1;`
- D) `byte sayı1 = 17;`
`int sayı2 = sayı1;`

10. `double sayı1 = 18.934;`
`int sayı2 = (int)sayı1;`
`System.out.println(sayı2);`

Yukarıdaki kod çalıştırıldığında ekran çıktısı aşağıdakilerden hangisidir?

- A) 18.934
- B) 18
- C) Hiçbiri
- D) 19

1. `if (5 > 7 && 10 > 6) {
 System.out.println("Elma");
} else {
 System.out.println("Armut");
}`

Ekran çıktısı nedir?

- A) Elma
- B) Armut
- C) Hiçbiri

2. `if (5 < 7 && 10 > 6) {
 System.out.println("Elma");
} else {
 System.out.println("Armut");
}`

Ekran çıktısı nedir?

- A) Elma
- B) Armut
- C) Hiçbiri

3. `if (5 > 7 || 10 < 6) {
 System.out.println("Elma");
} else {
 System.out.println("Armut");
}`

Ekran çıktısı nedir?

- A) Elma
- B) Armut
- C) Hiçbiri

```
4. If(yas > 65){  
    System.out.println("Emekli");  
}  
If(yas<70){  
    System.out.println("Dede");  
}
```

Yaşı 67 olan birisi için ekran çıktısı aşağıdakilerden hangisidir?

- A) Emekli
Dede
- B) EmekliDede
- C) Emekli
- D) Dede

```
5. if(sayı != 10){  
    System.out.println("Elma");  
}else{  
    System.out.println("Armut");  
}
```

9 için ekran çıktısı nedir?

- A) Elma
Armut
- B) Elma
- C) Armut
- D) Hiçbiri

If Else - If Elself

6. `if(sayı<100){
 System.out.println("Elma");
}
if(sayı<90){
 System.out.println("Armut");
}`

80 için ekran çıktısı nedir?

- A) Elma
Armut
- B) Elma
- C) Armut
- D) Hiçbiri

7. `if(sayı<100){
 System.out.println("Elma");
}else{
 System.out.println("Armut");
}`

100 için ekran çıktısı nedir?

- A) Armut
- B) Elma
- C) Elma
Armut
- D) Hiçbiri

8. **Aşağıdakilerden hangisi "OR" işlemi için yanlıştır?**

- A) Sembolü ||
- B) Şartlardan birinin doğru olması sonucun doğru olması için yeterlidir.
- C) Sonucun false olması için şartların ikisi de false olmalıdır.
- D) Hiçbiri

9. Aşağıdakilerden hangisi "AND" işlemi yanlıştır?

- A) Sembolu ||
- B) Sonucun false olması için bütün şartların false olması gerekir
- C) Sonucun true olması için şartlardan birinin true olması yeterlidir
- D) Hepsi

10. Scanner scan = new Scanner(System.in);
System.out.println("Bir pozitif integer giriniz");
int a = scan.nextInt();

if(a>9 && a<100) {
 System.out.println("Elma");
}
if(a>99 && a<1000) {
 System.out.println("Armut");
}
if(!(a>99 && a<1000) && !(a>9 && a<100)) {
 System.out.println("Karpuz");
}

Yukarıdaki koda göre aşağıdakilerden hangileri doğrudur?

- A) Kullanıcı 55 girerse console'a Elma yazdırır.
- B) Kullanıcı 555 girerse console'a Armut yazdırır.
- C) Kullanıcı 5555 girerse console'a Karpuz yazdırır.
- D) Kullanıcı Karpuz girerse console'a Karpuz yazdırır

1. `int y = 20;`
`System.out.println((y >= 20) ? (2 + y) : (y-3));`

Ekran çıktısı nedir?

- A) 20
- B) 22
- C) 17
- D) Kod çalışmaz

2. `int y = 8;`
`int x = (y % 2 == 1) ? (3 * y) : (y/4);`
`System.out.println(x);`

Ekran çıktısı nedir?

- A) 2
- B) 24
- C) 8
- D) Hiçbiri

3. `int y = 12;`
`if(y%2==0){`
 `System.out.println(3*y);`
`}else{`
 `System.out.println(5+y);`
`}`

Aşağıdakilerden hangilerinin ekran çıktısı yukarıdaki ile aynıdır?

- A) `int y = 12;`
 `int x = y%2==0 ? 3*y : 5+y;`
 `System.out.println(x);`
- B) `int y = 12;`
 `int x = y%2>0 ? 5+y : 3*y;`
 `System.out.println(x);`
- C) `int y = 9;`
 `int x = y%3!=1 ? 4*y : 8+y;`
 `System.out.println(x);`

Nested Ternary

4. `int y = 2;`
`int z = 0;`
`z >= 0 ? y++ : z--;`
`System.out.println(y + "," + z);`

Ekran çıktısı nedir?

- A) 3,0
- B) 2,0
- C) 3,-1
- D) 2,-1

5. `if(cinsiyet.equals("Erkek")) {`
 `if(yas<18) {`
 `System.out.println("Elma");`
 `}else {`
 `System.out.println("Armut");`
 `}`
`}else if(cinsiyet.equals("Kadin")) {`
 `if(yas<18) {`
 `System.out.println("Portakal");`
 `}else {`
 `System.out.println("Erik");`
 `}`
`}else {`
 `System.out.println("Domates");`
`}`

Aşağıdakilerden hangisi yanlıştır?

- A) Kullanıcı Erkek ve 18 girerse output Armut olur
- B) Kullanıcı Erkek ve 12 girerse output Elma olur
- C) Kullanıcı Kadın ve 32 girerse output Erik olur
- D) Kullanıcı kadın ve 13 girerse output Portakal olur

Nested Ternary

6. `Scanner scan = new Scanner(System.in);`
`System.out.println("Enter an integer");`
`int a = scan.nextInt();`

```
if(a%2==0) {
    if(a%3==0) {
        System.out.println("Wooooow!");
    }else {
        System.out.println("Hmmmm!");
    }
}else{
    if(a%3==0) {
        System.out.println("Normal!");
    }else {
        System.out.println("Yazik!");
    }
}
```

Yukarıdaki kod ile ilgili aşağıdakilerden hangisi yanlıştır?

- A) Kullanıcı 6 girerse output Wooooow! olur
- B) Kullanıcı 10 girerse output Hmmmm! olur
- C) Kullanıcı 9 girerse output Normal! olur
- D) Kullanıcı 15 girerse output Yazik! olur

7. `Scanner scan = new Scanner(System.in);`
`System.out.println("Bir tamsayı giriniz");`
`int n = scan.nextInt();`
`int result = n >= 0 ? n : -1 * n;`
`System.out.println(result);`

Aşağıdakilerden hangisi yanlıştır?

- A) `n=2` ise console'a 2 yazdırır.
- B) `n=-2` ise console'a 2 yazdırır.
- C) `n=0` ise console'a 0 yazdırır.
- D) Hiçbiri

Nested If Ternary

8. Scanner scan = new Scanner(System.in);
System.out.println("Enter password");
String password = scan.nextLine();
char initial = password.charAt(0);

```
if(initial>='A' && initial<='Z') {  
    if(initial=='A') {  
        System.out.println("Gecerli");  
    }else {  
        System.out.println("Gecersiz");  
    }  
}else if(initial>='a' && initial<='z') {  
    if(initial=='z') {  
        System.out.println("Gecerli");  
    }else {  
        System.out.println("Gecersiz");  
    }  
}else {  
    System.out.println("Bu ne?");  
}
```

Aşağıdakilerden hangisi yanlıştır?

- A) Kullanıcı Ali girerse output Geçerli olur
- B) Kullanıcı cem girerse output Geçersiz olur
- C) Kullanıcı 123 girerse output Bu ne? olur
- D) Kullanıcı B123 girerse output Geçerli olur

Nested If Ternary

9. `Scanner scan = new Scanner(System.in);`
`System.out.println("Ürün miktarını sayı olarak giriniz");`
`int miktar = scan.nextInt();`
`System.out.println("Ürün birim fiyatını giriniz");`
`double fiyat = scan.nextDouble();`
`double result = miktar > 1000 ? fiyat * 0.9 * miktar : fiyat * miktar;`
`System.out.println(result);`

Aşağıdakilerden hangileri doğrudur?

- A) miktar = 2000 ve fiyat = 3 ise console'a 5400.0 yazdırır.
- B) miktar = 200 ve fiyat = 3 ise console'a 600.0 yazdırır.
- C) miktar = 1000 ve fiyat = 3 ise console'a 2700.0 yazdırır.
- D) miktar = 900 ve fiyat = 3 ise console'a 2460.0 yazdırır.

10. `Scanner scan = new Scanner(System.in);`
`System.out.println("İki tamsayı giriniz");`
`int n1 = scan.nextInt();`
`int n2 = scan.nextInt();`
`int result = n1 > n2 ? n2 : n1;`
`System.out.println(result);`

Aşağıdakilerden hangileri doğrudur?

- A) n1 = 12 ve n2 = 13 ise console'a 12 yazdırır
- B) n1 = 12 ve n2 = -15 ise console'a -15 yazdırır
- C) n1 = 12 ve n2 = 12 ise console'a 12 yazdırır
- D) n1 = -13 ve n2 = 0 ise console'a -13 yazdırır

```
1. switch(m) {  
    case 'A':  
        System.out.println("First letter");  
        break;  
    case 'B':  
        System.out.println("Second letter");  
    case 'C':  
        System.out.println("Third letter");  
        break;  
    case 'D':  
        System.out.println("Fourth letter");  
    default:  
        System.out.println("What kind of letter is this?");  
}
```

m'nin hangi değerleri için console'a What kind of letter is this? yazdırır?

- A) B
- B) C
- C) D
- D) E

Nested Ternary Switch

```
2. switch(dayNum) {  
    case 1:  
        System.out.println("Sunday");  
        break;  
    case 2:  
        System.out.println("Monday");  
        break;  
    case 3:  
        System.out.println("Tuesday");  
        break;  
    case 4:  
        System.out.println("Wednesday");  
        break;  
    case 5:  
        System.out.println("Thursday");  
        break;  
    case 6:  
        System.out.println("Friday");  
        break;  
    case 7:  
        System.out.println("Saturday");  
        break;  
    default:  
        System.out.println("Enter a valid day number");  
}
```

- 1) Kullanıcı herhangi bir gün sayısı girdiğinde o günün ve sonraki günlerin isimlerini ekranda görmek istiyor.
- 2) default mesajını geçerli gün sayısı girdiğinde görmek istemiyor.

Buna göre yukarıdaki kod'dan kaç tane "break;" kodunu silmelidir?

- A) 5
- B) 6
- C) 7
- D) 4

Nested Ternary Switch

3. `int x = (y > 5) ? (y < 10 ? 2+y : 3*y) : (y > 10 ? 2*y : 3+y);`

Hangileri doğrudur?

- A) If $y = 10$ then the value of x is 30
- B) If $y = 4$ then the value of x is 8
- C) If $y = 11$ then the value of x is 13
- D) If $y = 1$ then the value of x is 4

4. `String str = (y > 'a' && y < 'z') ? (y < 'f' ? "Good" : "Bad") : (y > 'K' ? "Big" : "Small");`

Hangileri doğrudur?

- A) If $y = 'c'$ then the value of str is Good
- B) If $y = 'h'$ then the value of str is Bad
- C) If $y = 'K'$ then the value of str is Small
- D) If $y = 'P'$ then the value of str is Big

5. `boolean result = (y%2==0 || y<13) ? (true) : (y<0 ? true : false);`
`System.out.println(result);`

Hangisi yanlıştır?

- A) if $y = 12$ then
prints true on the console
- B) if $y = -12$ then
prints true on the console
- C) if $y = 15$ then
prints false on the console
- D) if $y = -17$ then
prints false on the console

Nested Ternary Switch

6. `String result = (a%2!=0 && a>29) ? ("Good") : ("Bad");`
`System.out.println(result);`

Hangileri doğrudur?

- A) If a = 33 then the output is Good
- B) If a = 29 then the output is Bad
- C) If a = 12 then the output is Bad
- D) If a = 44 then the output is Good

7. `int i = 1;`
`int j = 0;`
`switch (i) {`
 `case 2 :`
 `j += 6;`
 `case 4 :`
 `j += 1;`
 `case 1 :`
 `j += 4;`
 `default :`
 `j += 2;`
`}`
`System.out.println("j = " + j);`

Ekran çıktısı nedir?

- A) j = 6
- B) j = 4
- C) j = 2
- D) j = 0



Nested Ternary Switch

```
8. char ch = 'a';
switch (ch){
    case 'a':
    case 'A':
        System.out.print(ch);
        break;
    case 'b':
    case 'B':
        System.out.print(ch);
        break;
    case 'c':
    case 'C':
        System.out.print(ch);
        break;
    case 'd':
    case 'D':
        System.out.print(ch);
}
```

Ekran çıktısı nedir?

- A) a
- B) a
a
- C) a
A
- D) a
a
A
A

Nested Ternary Switch

9. `int number = 3;`
`if(number == 1) {`
 `System.out.println("One");`
`}`
`if(number == 2){`
 `System.out.println("Two");`
`}`
`if(number == 3){`
 `System.out.println("Three");`
`}`

Aşağıdakilerden hangisi yukarıdaki program ile aynı çıktıyı verir?

A) `int number = 3;`
`switch(number){`
 `case 1:`
 `System.out.println("One");`
 `break;`
 `case 2:`
 `System.out.println("Two");`
 `break;`
 `case 3:`
 `System.out.println("Three");`
 `break;`
`}`

B) `int number = 3;`
`switch(number){`
 `case 1:`
 `System.out.println("One");`
 `break;`
 `case 2:`
 `System.out.println("Two");`
 `break;`
 `case 3:`
 `System.out.println("Three");`
 `default:`
 `System.out.println("Done");`
`}`

Nested Ternary Switch

```
C) int number = 3;
   switch(number){
       case 3:
           System.out.println("One");
           break;
       case 2:
           System.out.println("Two");
           break;
       case 1:
           System.out.println("Three");
           break;
       default:
           System.out.println("Done");
   }
```

```
10. int variable = 1;
    switch (variable){
        case 1 :
            System.out.print("P");
        case 2 :
        case 3 :
            System.out.print("Q");
            break;
        case 4 :
            System.out.print("R");
        default :
            System.out.print("S");
    }
```

Ekran çıktısı nedir?

- A) P
- B) S
- C) PQ
- D) PQR

1. `Scanner scan = new Scanner(System.in);`
`System.out.println("İsminizi giriniz");`
`String isim = scan.nextLine();`

Kullanıcıdan ismini aldıktan sonra, o ismin tüm harflerini büyük harfe çevirmek için Aşağıdaki kodlardan hangisi yazılmalıdır?

- A) `isim = isim.toLowerCase();`
- B) `isim = isim.toLowerCase;`
- C) `isim = isim.toUpperCase();`
- D) `isim = isim.toUpperCase;`

2. `String str = "sarıkanarya";`

Yukarıda verilen String'e göre aşağıdakilerden hangisi yanlıştır?

- A) `System.out.println(str.charAt(4));`
Ekran çıktısı k olur.
- B) `System.out.println(str.charAt(0));`
Yukarıdaki kod ilk harfi ekrana yazdırır
- C) `System.out.println(str.length());`
Ekran çıktısı 10 olur.
- D) `System.out.println(str.charAt(str.length()-1));`
Yukarıdaki kod son harfi ekrana yazdırır

3. `String str = "Ayhan Beyhan";`

Aşağıdakilerden hangisi yanlıştır?

- A) `System.out.println(str.toLowerCase());`
Ekran ayhan beyhan yazdırır.
- B) `System.out.println(str.replace("h", ""));`
Ekran Ayan Beyan yazdırır.
- C) `System.out.println(str.replace("han", "ol"));`
Compile Time Error verir çünkü 3 harf yerine 2 harf konulamaz
- D) `System.out.println(str.length());`
Ekran 12 yazdırır.

4. String str = "Java ah java";

Yukarıda verilen String'e göre aşağıdakilerden hangisi yanlıştır?

- A) System.out.println(str.contains("J"));
Ekranı true yazdırır
- B) System.out.println(str.contains("x"));
Ekranı -1 yazdırır
- C) System.out.println(str.contains("ja"));
Ekranı true yazdırır

5. String str1 = "Ali";
String str2 = "ali";

Yukarıda verilen String'lere göre aşağıdakilerden hangisi yanlıştır?

- A) System.out.println(str1.equals(str2));
Ekranı true yazdırır
- B) System.out.println(str1.equalsIgnoreCase(str2));
Ekranı true yazdırır
- C) System.out.println(str2.equals(str1));
Ekranı false yazdırır
- D) System.out.println(str2.equalsIgnoreCase(str1));
Ekranı true yazdırır

6. **Aşağıdakilerden hangisi yanlıştır?**

- A) !(false) = true
- B) !(true) = true
- C) !(4<5) = false
- D) !(2>3) = false

7. String str1 = "ali";
String str2 = "Ali";

Yukarıda verilen String'lere göre aşağıdakilerden hangileri yanlıştır?

- A) System.out.println(str1.equalsIgnoreCase(str2));
Ekrana true yazdırır
- B) System.out.println(str1==str1);
Ekrana true yazdırır
- C) System.out.println(str1.equals(str2));
Ekrana true yazdırır
- D) System.out.println((str1+str2).equals(str2+str1));
Ekrana true yazdırır

8. String str = "şeyhan yavuz";

Yukarıdaki Stringde "s" ve "y" harflerini büyük harf yapmak için hangileri doğrudur?

- A) System.out.println(str.replace("s", "S"));
System.out.println(str.replace("y", "Y"));
- B) System.out.println(str.replaceAll("s", "S"));
System.out.println(str.replaceAll("y", "Y"));
- C) System.out.println(str.replace('s', 'S'));
System.out.println(str.replace('y', 'Y'));
- D) System.out.println(str.replaceAll('s', 'S'));
System.out.println(str.replaceAll('y', 'Y'));

9. String str = "Java Kolaydır";

Aşağıdakilerden hangileri doğrudur?

- A) System.out.println(str.endsWith("r"));
Ekranan true yazdırır
- B) System.out.println(str.endsWith("dir"));
Ekranan true yazdırır
- C) System.out.println(str.endsWith(""));
Ekranan true yazdırır
- D) System.out.println(str.endsWith("Java Kolaydır"));
Ekranan true yazdırır

10. String b = "Ali 12 ?_";

Aşağıdakilerden hangisi yanlıştır?

- A) System.out.println(b.replaceAll("\\d", "*"));
Ekranan Ali ** ?_ yazdırır
- B) System.out.println(b.replaceAll("\\D", "*"));
Ekranan ****12*** yazdırır
- C) System.out.println(b.replaceAll("\\S", "*"));
Ekranan *** ** ** yazdırır
- D) System.out.println(b.replaceAll("\\w", "*"));
Ekranan Ali*12**_ yazdırır

1. String str= "Canan Can";

Aşağıdakilerden hangileri ekrana true yazdırır?

- A) System.out.println(str.startsWith("c"));
- B) System.out.println(str.endsWith("n"));
- C) System.out.println(str.startsWith("a",3));
- D) System.out.println(str.endsWith("Can"));

2. String str = "Java code yaz";

Aşağıdakilerden hangileri doğrudur?

- A) System.out.println(str.substring(5));
Ekranaya ode yaz yazdırır.
- B) System.out.println(str.substring(12));
Ekranaya z yazdırır.
- C) System.out.println(str.substring(5, 9));
Ekranaya code yazdırır.
- D) System.out.println(str.substring(8, 8));
Ekranada hiçbirşey göremezsiniz.

3. String text = "Can, Cananlara gitmek için can atıyor";
System.out.println(text.replace("Can", "Türk"));

Yukarıdaki kodun ekran çıktısı nedir?

- A) Türk, Türkanlara gitmek için turk atıyor
- B) Türk, Türkanlara gitmek için can atıyor
- C) Hiçbiri
- D) Türk, Cananlara gitmek için can atıyor

String Methods For Loop

4. String str = "Ayhan BEYHAN";

Yukarıdaki String'i AYHAN beyhan yapmak için hangileri yapılabilir?

- A) str.substring(0,5).toUpperCase();
str.substring(6).toLowerCase();
- B) str.replace("Ayhan", "AYHAN");
str.replace("BEYHAN", "beyhan");
- C) str.replace('Ayhan', 'AYHAN');
str.replace('BEYHAN', 'beyhan');
- D) str.replace(str.substring(0), "AYHAN beyhan");

5. for(int i=1; i<6; i++) {
 System.out.print(i + "-");
}

Yukarıdaki programın ekran çıktısı nedir?

- A) 1-2-3-4-5-
- B) 5-4-3-2-1-
- C) 1-2-3-4-5-6-
- D) 1-
2-
3-
4-
5-

String Methods For Loop

6. Aşağıdakilerden hangileri aynı ekran çıktısını verir?

- A)

```
for(int i=1; i<10; i+=2){  
    System.out.println(i);  
}
```
- B)

```
for(int i=1; i<10; i=i+2){  
    System.out.println(i);  
}
```
- C)

```
for(int i=9; i>=1; i-=2){  
    System.out.println(i);  
}
```
- D)

```
for(int i=1; i<=9; i++){  
    System.out.println(i);  
    i++;  
}
```

7. Hangisi doğrudur?

- A)

```
String str = "Java";  
System.out.println(str.replace("", "/"));
```

Ekrana J/a/v/a yazdırır
- B)

```
String str = "Java";  
System.out.println(str.replace("", "/"));
```

Ekrana /J/a/v/a yazdırır
- C)

```
String str = "Java";  
System.out.println(str.replace("", "/"));
```

Ekrana /J/a/v/a/ yazdırır
- D)

```
String str = "Java";  
System.out.println(str.replace("", "/"));
```

Ekrana Java yazdırır

8. String s = "Java is Java";

Hangisi doğrudur?

- A) System.out.println(s.indexOf('a'));
Ekran 2 yazdırır
- B) System.out.println(s.indexOf("Java"));
Ekran 0 yazdırır
- C) System.out.println(s.indexOf('a', 4));
Ekran 10 yazdırır
- D) System.out.println(s.indexOf("va", 2));
Ekran 10 yazdırır

9. String s = "Java is Java";

Hangileri doğrudur?

- A) System.out.println(s.lastIndexOf('v'));
Ekran 11 yazdırır
- B) System.out.println(s.lastIndexOf("va"));
Ekran 10 yazdırır
- C) System.out.println(s.toUpperCase().lastIndexOf("V"));
Ekran -1 yazdırır
- D) System.out.println(s.toLowerCase().lastIndexOf("j"));
Ekran 8 yazdırır

10. String s = "Learn";

String t = "Java";

Hangisi yanlıştır?

- A) System.out.println((s.concat(t)).length());
Ekran 9 yazdırır
- B) System.out.println("" + s.length() + t.length());
Ekran 9 yazdırır
- C) System.out.println(s.concat(t.length()));
Ekran Learn4 yazdırır
- D) System.out.println(s.length() > s.charAt(2));

1. `int sum =0;`
`for(int i=1; i<6; i++) {`
`sum = sum + i;`
`}`
`System.out.println("Toplam: " + sum);`

Yukarıdaki programın ekran çıktısı nedir?

- A) Toplam: 10
- B) Toplam: 15
- C) Toplam: 6
- D) Hiçbiri

2. `int sum=0;`
`for(int i=10; i>0; i=i-5) {`
`sum=sum+i;`
`}`
`System.out.println(sum);`

Yukarıdaki programın ekran çıktısı nedir?

- A) 10
- B) 15
- C) 0
- D) Hiçbiri

3. `for (int j = 0; j <=10; ?){`
`System.out.print(j + " ");`
`}`
? yerine ne yazılırsa output 0 3 6 9 olur?

- A) `j = j + 3`
- B) `j = j - 3`
- C) `j++`
- D) `j--`

For Loops

4. `int product = 1;`
`for(int i=5; i>=0; i--){`
`product = product * i;`
`}`

`System.out.println(product);`

Yukarıdaki programın ekran çıktısı nedir?

- A) 0
- B) 5
- C) 24
- D) 120

5. `for (int i = 1; i <= 6; i++) {`
`System.out.print(i + " ");`
`i++;`
`}`

Output nedir?

- A) 1 3 5
- B) 2 4 6
- C) 1 2 3 4 5 6
- D) Compile Time Error

6. `for (int j = 10; j >0; j++){`
`System.out.print("Hello");`
`}`

Ekrana kac kere Hello yazdırır?

- A) 10
- B) 11
- C) 0
- D) Sonsuz kere

For Loops

7. `for (char i = 'f'; i > 'a'; i--){
 System.out.print(i + " ");
 i--;
}`

Output nedir?

- A) f d b
- B) f e d b c
- C) e c
- D) Run Time Error

8. **Aşağıdakilerden hangileri ekrana 10 kez "Welcome to Java" yazar?**

- A) `for (int count = 1; count <= 10; count++) {
 System.out.println("Welcome to Java");
}`
- B) `for (int count = 1; count < 10; count++) {
 System.out.println("Welcome to Java");
}`
- C) `for (int count = 10; count >= 1; count--) {
 System.out.println("Welcome to Java");
}`
- D) `for (int count = 1; count < 11; count++) {
 System.out.println("Welcome to Java");
}`

For Loops

9.

```
for(int i=1; i<=5; i++) {  
    for(int j=(5-i); j>1; j--) {  
        System.out.print(i+j);  
    }  
    System.out.println();  
}
```

Output nedir?

A) 543
54
5

B) 432
43
4

C) 5
4
3

D) 5432
543
54

For Loops

```
10. Scanner scan = new Scanner(System.in);  
    System.out.println("Başlangıç değerini giriniz");  
    int bas = scan.nextInt();  
    System.out.println("Bitiş değerini giriniz");  
    int bit = scan.nextInt();  
  
    int carpim = 1;  
    int toplam = 0;  
  
    if(bas<=bit) {  
        for(int i=bas; i<=bit; i++) {  
            toplam = toplam + i;  
        }  
    } else {  
        for(int i=bas; i>=bit; i--) {  
            carpim = carpim * i;  
        }  
    }  
    System.out.print(carpim);  
    System.out.println(toplam);  
    scan.close();  
}
```

Yukarıdaki kod ile ilgili aşağıdakilerden hangileri doğrudur?

- A) Başlangıç değeri bitiş değerinden küçük ve eşit ise, program başlangıç değerinden bitiş değerine kadar tüm tamsayıları toplar.
- B) Başlangıç değeri bitiş değerinden büyük ise, program başlangıç değerinden bitiş değerine kadar tüm tamsayıları çarpar.
- C) Başlangıç değeri 3, bitiş değeri 5 ise, program ekrana 112 yazdırır.
- D) Başlangıç değeri 5, bitiş değeri 3 ise, program ekrana 600 yazdırır.

1. `int num = 1;`
`while(num<3) {`
 `System.out.print(num + " ");`
 `num++;`
`}`

Yukarıdaki programın ekran çıktısı nedir?

- A) 1 2 3
- B) 1 2
- C) 1
2
- D) 1

2. `int num = 1;`
`while(num<1) {`
 `System.out.print(num + " ");`
 `num++;`
`}`

Yukarıdaki programın ekran çıktısı nedir?

- A) 1
- B) Program çalışmaz
- C) Program çalışır ama ekranda hiçbirşey görülmez

3. `int num = 1;`
`while(num<=1) {`
 `System.out.print(num + " ");`
 `num--;`
`}`

Yukarıdaki programın ekran çıktısı nedir?

- A) 1
- B) Program çalışmaz
- C) 1 0 -1 -2 ... şeklinde negative sonsuza kadar yazar

While Loops

4. `int num = 1;`
`while(num=1) {`
 `System.out.print(num + " ");`
 `num++;`
`}`

Yukarıdaki programın ekran çıktısı nedir?

- A) 1
- B) Program çalışmaz
- C) Sonsuz defa "1 " yazar
- D) Hiçbiri

5. `char ch = 'a';`
`while(ch<'d') {`
 `System.out.print(ch + " ");`
 `ch++;`
`}`

Aşağıdakilerden hangilerinin ekran çıktısı yukarıdaki ile aynıdır?

- A) `for(char i='a'; i<'d'; i++) {`
 `System.out.print(i + " ");`
`}`
- B) `System.out.print('a ' + 'b ' + 'c ');`
Not: a, b ve c harflerinden sonra tek tırnağın içinde bir karakter bosluk vardır.
- C) `System.out.print('a ' + " " + 'b ' + " " + 'c ' + " ");`
Not: Cift tırnakların içinde bir karakter bosluk vardır.
- D) `System.out.print('a' + 'b' + 'c');`

While Loops

6.

```
for(int i=1; i<10; i=i+1) {  
    if(i%3==0) {  
        System.out.print(i + " ");  
    }  
}
```

Aşağıdakilerden hangilerinin ekran çıktısı yukarıdaki ile aynıdır?

- A)

```
int num = 3;  
while(num<10) {  
    System.out.print(num + " ");  
    num=num+3;  
}
```
- B)

```
int num = 1;  
while(num<10) {  
    if(num%3==0){  
        System.out.print(num + " ");  
    }  
    num=num+1;  
}
```
- C)

```
int num = 9;  
while(num>1) {  
    System.out.print(num + " ");  
    num=num-3;  
}
```

While Loops

7. `int sum = 0;`
`int num = 1;`
`while(num<5) {`
 `sum = sum + num;`
 `num++;`
`}`
`System.out.println(sum);`

Yukarıdaki programın ekran çıktısı nedir?

- A) Program çalışmaz
- B) 10
- C) 15
- D) Program çalışır ama ekranda hiçbirşey görülmez

8. `int sum = 5;`
`int num = 1;`
`while(num<5) {`
 `sum = sum + num;`
 `num++;`
`}`
`System.out.println(sum);`

Yukarıdaki programın ekran çıktısı nedir?

- A) Program çalışmaz
- B) 10
- C) 15
- D) Program çalışır ama ekranda hiçbirşey görülmez.

While Loops

9. `int product = 1;`
`int num = 1;`
`while(num<5) {`
 `product = product * num;`
 `num++;`
`}`
`System.out.println(product);`

Yukarıdaki programın ekran çıktısı nedir?

- A) 10
- B) 12
- C) 120
- D) 24

10. `int product = 1;`
`int num = 1;`
`while(num<5) {`
 `product = product * num;`
 `product++;`
`}`
`System.out.println(product);`

Yukarıdaki programın ekran çıktısı nedir?

- A) 120
- B) 24
- C) Program çalışmaz
- D) Sonsuz döngü oluşur

1. `int num = 1;`
`do {`
 `System.out.print(num + " ");`
 `num++;`
`}while(num<4);`

Yukarıdaki programın ekran çıktısı nedir?

- A) 1 2 3
- B) 1
2
3
- C) 1 2 3 4
- D) 1
2
3
4



2. `int num = 1;`
`do {`
 `System.out.print(num + " ");`
 `num++;`
`}while(num<1);`

Yukarıdaki programın ekran çıktısı nedir?

- A) Program çalışır ama ekranda hiç bir şey görünmez
- B) 1
- C) Program çalışmaz
- D) 1 2

Do While Loop

3. `int num = 1;`
`do {`
 `System.out.print(num + " ");`
 `num++;`
`}while(num<2);`

Aşağıdakilerden hangisinin ekran çıktısı yukarıdaki programın ekran çıktısı ile aynıdır?

- A) `int num=1;`
 `while(num<2) {`
 `System.out.print(num + " ");`
 `num+=1;`
 `}`
- B) `int num=1;`
 `while(num<=1) {`
 `System.out.print(num + " ");`
 `num++;`
 `}`
- C) `for(int num=1; num<2; num++){`
 `System.out.print(num + " ");`
 `}`
- D) Hepsi

4. **Aşağıdakilerden hangileri while loop ile do-while loop'un farklarındandır?**

- A) while loop önce condition'i kontrol eder sonra kodu çalıştırır.
do-while loop önce kodu çalıştırır sonra condition'i kontrol eder.
- B) while loop'da body içindeki kod bazı durumlarda hiç çalışmayabilir.
do-while loop'da body içindeki kod en az bir kere çalışır.
- C) while loop sonsuz döngüye girebilir.
do-while loop sonsuz döngüye girmez.

Do While Loop

5. `int numA=4;`
`while(numA<3) {`
`System.out.print(numA);`
`numA++;`
`}`

`int numB=4;`
`do {`
`System.out.print(numB);`
`numB++;`
`}while(numB<3);`

Yukarıdaki while loop ile do-while loop'ların ekran çıktıları ile ilgili aşağıdakilerden hangisi doğrudur?

- A) while loop ekrana hiçbirşey yazdıramaz.
do-while loop ekrana 4 yazdırır.
- B) while loop ekrana hiçbirşey yazdıramaz.
do-while loop ekrana hiçbirşey yazdıramaz.
- C) while loop ekrana 4 yazdırır.
do-while loop ekrana 45 yazdırır.
- D) Hiçbiri

6. `int num = 2;`
`do {`
`System.out.print(num + " ");`
`num++;`
`}while(num>2);`

Yukarıdaki program ile ilgili aşağıdakilerden hangisi doğrudur?

- A) Program çalışmaz.
- B) Sonsuz döngü oluşur.
- C) Ekrana 2 yazdırır.
- D) Program çalışır ama ekrana hiçbirşey yazdırmaz.

Do While Loop

7. `int num=0;`
`do {`
 `System.out.print(num + " ");`
 `num+=2;`
`}while(num<8);`

Yukarıdaki programın ekran çıktısı nedir?

- A) 2 4 6 8
- B) 0 2 4 6 8
- C) 0 2 4 6
- D) 2 4

8. `int num=0;`
`do {`
 `if(num%2==0){`
 `System.out.print(num + " ");`
 `}`
 `num++;`
`}while(num<8);`

Yukarıdaki programın ekran çıktısı nedir?

- A) 2 4 6 8
- B) 0 2 4 6 8
- C) 0 2 4 6
- D) 2 4

9. `String str = "AliCan";`
`System.out.println(str.substring(2,5));`

Yukarıdaki programın ekran çıktısı nedir?

- A) iCan
- B) iCa
- C) liCa
- D) liCan

Do While Loop

10. Kullanıcıdan passwordunu alıp, password'un "AliCan" olup olmadığını kontrol eden, password doğru ise ekrana "Doğru", password yanlış ise ekrana "Yanlış" ve "Passwordunuzu giriniz" yazdıran program aşağıdakilerden hangileri gibi olabilir?

A)

```
Scanner scan = new Scanner(System.in);
String password = "";
do {
    System.out.println("Passwordunuzu giriniz");
    password = scan.nextLine();

    if(password.equals("AliCan")) {
        System.out.println("Doğru");
    }else {
        System.out.println("Yanlış");
    }

}while(!(password.equals("AliCan")));
```

B)

```
Scanner scan = new Scanner(System.in);
String password = "";
do {
    System.out.println("Passwordunuzu giriniz");
    password = scan.nextLine();

    if(!(password.equals("AliCan"))) {
        System.out.println("Yanlış");
    }
}while(!(password.equals("AliCan")));
System.out.println("Doğru");
```

C)

```
Scanner scan = new Scanner(System.in);
System.out.println("Passwordunuzu giriniz");
String password = scan.nextLine();
while(!(password.equals("AliCan"))) {
    System.out.println("Yanlış");
    System.out.println("Passwordunuzu giriniz");
    password = scan.nextLine();
}
System.out.println("Doğru");
```

1. **Not:** Eğer aynı isimli farklı parametrelili method'lar oluşturursanız buna method overloading denir.

Aşağıdakilerden hangisi method overloading yapmak için doğru bir yol değildir?

- A) Parametrelerin sayısını değiştirmek
- B) Parametrelerin data type'ini değiştirmek
- C) Parametrelerin data type'lari farklı ise yerlerini değiştirmek
- D) Method body'i değiştirmek.

2. `public static void multiply(int num1, int num2) {
 System.out.println(num1* num2);
}
public static void multiply(int num1) {
 System.out.println(num1*num1);
}
public static void multiply(double num1, double num2) {
 System.out.println(num1* num2);
}`

Hangisi yanlıştır?

- A) `public static void main(String[] args) {
 multiply(3, 5);
}`
Ekran 15 yazar.
- B) `public static void main(String[] args) {
 multiply(5);
}`
Ekran 25 yazar
- C) `public static void main(String[] args) {
 multiply(2.5);
}`
Ekran 4 yazar
- D) `public static void main(String[] args) {
 multiply(2.5, 2.5);
}`
Ekran 6.25 yazar

3.

```
public static void add(int num1, int num2) {  
    System.out.println(2 + num1 + num2);  
}  
public static void add(double num1, double num2) {  
    System.out.println(3 + num1 + num2);  
}
```

Hangisi yanlıştır?

- A)

```
public static void main(String[] args) {  
    add(3, 5);  
}
```


Ekрана 10 yazар
- B)

```
public static void main(String[] args) {  
    add(3.1, 5.2);  
}
```


Ekрана 11.3 yazар
- C)

```
public static void main(String[] args) {  
    add(3.1, 5);  
}
```
- D) Compile Time Error verir

4.

```
public static void main(String[] args) {  
    updateWord("john", "black");  
}  
public static void updateWord(String s, String t) {  
    s = s.substring(0,1).toUpperCase() + t.substring(2);  
    System.out.println(s);  
}
```

Output nedir?

- A) Jack
- B) Joack
- C) Jck
- D) Compile Time Error

5.

```
private static void getData(){
    System.out.println("Method 1");
}
public static void getData(){
    System.out.println("Method 2");
}
public static void main(String[] args){
    getData();
}
```

Hangisi doğrudur?

- A) Ekrana 0 yazar
- B) Ekrana 1 yazar
- C) Ekrana hiçbirşey yazmaz
- D) Compile Time Error verir

6.

```
public static void main(String[] args) {
    add(3.2, 4.8); //Output A
    add(5, 7.3); //Output B
    add(5, 7); //Output C
    add(5.3, 7); //Output D
}

public static void add(double n1, double n2) {
    System.out.println(n1 + n2);
}

public static void add(int n1, double n2) {
    System.out.println(n1 + n2);
}

public static void add(int n1, int n2) {
    System.out.println(n1 + n2);
}
```

Hangisi yanlıştır?

- A) Output A = 8.0
- B) Output B = 12.3
- C) Output C = 12
- D) Output D Compile Time Error verir.

7. Hangileri doğrudur?

A)

```
public static void function(float i, int j){  
    System.out.println("Good");  
}  
public static void function(double i, double j){  
    System.out.println("Bad");  
}  
public static void main(String[] args){  
    function(1.2, 3);  
}
```

Ekrana Bad yazar

B)

```
public static void function(float i, int j){  
    System.out.println("Good");  
}  
public static void function(double i, double j){  
    System.out.println("Bad");  
}  
public static void main(String[] args){  
    function(2, 3);  
}
```

Ekrana Good yazar

C)

```
public static void function(float i, int j){  
    System.out.println("Good");  
}  
public static void function(double i, double j){  
    System.out.println("Bad");  
}  
public static void main(String[] args){  
    function(2.1F, 3.2);  
}
```

Ekrana Bad yazar


```
8. public class MyClass{
    int num1;
    String name = "Ali";
    public static void main(String args){
        add();
        product(5);
    }
    public static void add(){
        num1++;
        int num2 = 6;
        char letter;
        System.out.println("Do addition");
    }
    public void product(int num3){
        name = "Veli";
        num2++;
        System.out.println(num3 * num3);
    }
}
```

Aşağıdakilerden hangileri doğrudur?

- A) num1 ve name instance variable'dir.
- B) num2 ve letter local variable'dir.
- C) System.out.println(num1) ekrana hiçbirşey yazdırmaz
- D) num2++; kodu Compile Time Error verir.
- E) product(5); kodu Compile Time Error verir.

1. Aşağıdakilerden hangisi Constructor ile Method'un farklarından değildir?

- A) Constructor'larda return type yoktur, Method'larda return type olmalıdır
- B) Constructor'ların ismi class ile aynı olmalıdır, Method'ların ismi class'dan farklı olabilir.
- C) Constructor'ların ismi büyük harf ile başlar, Method'ların ismi küçük harfle başlar.
- D) Bir class'da sadece bir tane Constructor olabilir, fakat birden fazla Method olabilir.

2.

```
public class MyClass {  
    int price = 20000;  
    int year = 2020;  
    String make = "Honda";  
    public static void main(String[] args) {  
        MyClass hondaAraba = new MyClass();  
        System.out.println(hondaAraba.price);  
        System.out.println(hondaAraba.year - 3);  
        System.out.println(hondaAraba.make.replace("H", "K"));  
    }  
}
```

Yukarıdaki programın ekran çıktısı nedir?

- A) 20000
2020
Honda
- B) 20000
2020
Konda
- C) 20000
2017
Konda
- D) 2017
Konda

3. Aşağıdakilerden hangileri Constructor için doğrudur?

- A) Her class oluşturulduğunda Java otomatik olarak bir Constructor oluşturur, bu Constructor'a default constructor denir.
- B) Default Constructor'ların parametresi olmaz.
- C) Biz bir class'da Constructor oluşturduğumuzda, Java default constructor'i iptal eder.
- D) Constructor'lar bir class'dan yeni bir obje oluşturmak için kullanılır.

4. public class MyClass {

String isim = "Ali Can";

int yas = 33;

int kilo = 85;

MyClass(){

}

MyClass(String isim, int yas){

this.isim = isim;

this.yas = yas;

}

MyClass(String isim){

this.isim = isim;

}

}

Yukarıdaki class'da 3 farklı constructor verilmiştir.

Aşağıdakilerden hangisi yanlıştır?

- A) MyClass insan01 = new MyClass();
insan01 objesinin ismi Ali Can olur.
- B) MyClass insan02 = new MyClass("Veli Han");
insan02 objesinin ismi Veli Han, yasi 33, kilosu 85 olur.
- C) MyClass insan03 = new MyClass(44, "Ayse Tan");
insan03 objesinin ismi Ayse Tan, yasi 44, kilosu 85 olur.

5. Aşağıdakilerden hangileri default constructor'a sahiptir?

- A) `public class Bird { }`
- B) `public class Bird { bird() { } }`
- C) `public class Bird { Bird(String name) { } }`
- D) `public class Bird { Bird() { } }`
- E) `public class Bird { void Bird() { } }`

6. `public class Car {
 int price = 50;

 public Car(int price) {
 this.price = price;
 }
 public static void main(String[] args) {

 }
}`

Hangisi ekrana 50 yazar?

- A) `Car c1 = new Car(50);
System.out.println(c1.price);`
- B) `Car c1 = new Car();
System.out.println(c1.price);`
- C) `Car c1 = new Car(50);
System.out.println(c1);`
- D) `Car c1 = new Car();
System.out.println(c1);`

7.

```
public class BirdSeed {  
    private int numberBags;  
    boolean call;  
  
    public BirdSeed() {  
        this.numberBags = 11;  
        call = false;  
    }  
    public BirdSeed(int numberBags) {  
        this.numberBags = numberBags;  
        this.call = true;  
    }  
    public static void main(String[] args) {  
    }  
}
```

Aşağıdakilerden hangisi yanlıştır?

- A)

```
BirdSeed seed = new BirdSeed();  
System.out.println(seed.numberBags);
```


Output 11 dir
- B)

```
BirdSeed seed = new BirdSeed(7);  
System.out.println(seed.numberBags);
```


Output 7 dir
- C)

```
BirdSeed seed = new BirdSeed();  
System.out.println(seed.call);
```


Output false dir
- D)

```
BirdSeed seed = new BirdSeed(7);  
System.out.println(seed.call);
```


Gives Compile Time Error.

8.

```
public class Order {  
    String value = "a";  
    Order() {  
        value = value + "b";  
    }  
    Order(String c) {  
        value = value + c;  
    }  
    public static void main(String[] args) {  
    }  
}
```

Hangileri doğrudur?

- A)

```
Order order = new Order("f");  
System.out.println(order.value);
```


Output af dir
- B)

```
Order order = new Order();  
System.out.println(order.value);
```


Output ab dir
- C) Gives Compile Time Error
- D)

```
Order order1 = new Order("r");  
Order order2 = new Order();  
System.out.println(order1.value + order2.value);
```


Output arab dir

```
1. public class Try {  
    static int a = 10;  
    public static void main(String[] args) {  
        Try t1 = new Try();  
        Try t2 = new Try();  
  
        t1.a = a + 2;  
        t2.a = a + 3;  
        System.out.println(t1.a);  
        System.out.println(t2.a);  
    }  
}
```

Output nedir?

- A) 15
15
- B) 12
13
- C) 12
15
- D) 10
10



Static Keyword

```
2. public class Try {  
    int a = 10;  
    public static void main(String[] args) {  
        Try t1 = new Try();  
        Try t2 = new Try();  
  
        t1.a = 12;  
        t2.a = 13;  
        System.out.println(t1.a);  
        System.out.println(t2.a);  
    }  
}
```

Output nedir?

- A) 12
13
- B) 15
15
- C) 13
12
- D) 10
10

3. Hangileri doğrudur?

- A) Static variable'lara sadece classin ismi kullanılarak da ulaşılabilir.
- B) Static olmayan variable'lara ulaşabilmek için object oluşturmak zorundayız.
- C) Static variable'ların diğer adı class variable'dir. Instance variable'ların diğer adı object variable'dir.
- D) Static variable'lar object'lerin ortak kullanımına açıktır fakat instance variable'lar değildir.


```
4. public class Try {  
    public static void main(String[] args) {  
    }  
    public static void m1(float x) {  
        System.out.println("Method 1");  
    }  
    public static void m1(int x, double y) {  
        System.out.println("Method 2");  
    }  
}
```

}Eğer aşağıdaki kodları main method içine yazarsak hangileri yanlıştır?

- A) m1(33);
prints Method 1 on the console
- B) m1(33, 44);
prints Method 2 on the console
- C) m1(33.3, 44);
prints Method 2 on the console
- D) m1(33.3);
prints Method 1 on the console

Static Keyword

```
5. class Student{
    String name;
    static String college ="UNF";

    Student(String n, String college){
        this.name = n;
        this.college = college;
    }

    public static void main(String args[]){
        Student s1 = new Student("Mark", "UCF");
        Student s2 = new Student("Kevin", "FIU");

        System.out.println(s1.name);
        System.out.println(s2.name);

        System.out.println(s1.college);
        System.out.println(s2.college);
    }
}
```

Output nedir?

- A) Mark
Kevin
FIU
FIU
- B) Mark
Kevin
UCF
FIU
- C) Kevin
Kevin
UCF
FIU
- D) Kevin
Kevin
FIU
FIU

```
6. class Counter {  
    int count=0;  
    Counter(){  
        count+=2;  
        System.out.println(count);  
    }  
    public static void main(String args[]){  
        Counter c1=new Counter();  
        Counter c2=new Counter();  
        Counter c3=new Counter();  
    }  
}
```

Output nedir?

A) 2

2

2

B) 0

2

4

C) 2

4

6

D) 6

6

6

Static Keyword

```
7. public class Test1 {  
    int y = 5;  
    static int x = 7;  
    public static void main(String[] args) {  
        Test1 t1 = new Test1();  
        Test1 t2 = new Test1();  
        t1.x = 17;  
        t1.y = 15;  
        System.out.print(t1.x + " ");  
        System.out.println(t2.x);  
  
        System.out.print(t1.y + " ");  
        System.out.println(t2.y);  
    }  
}
```

Output nedir?

- A) 17 17
15 5
- B) 17 17
15 15
- C) 17 7
15 5
- D) 7 7
5 5

Static Keyword

```
8. public class StaticDemo {  
    int num1 = 5;  
    static int num2 = 7;  
  
    public static void main(String args[]) {  
        StaticDemo s1 = new StaticDemo();  
        StaticDemo s2 = new StaticDemo();  
        s1.num1 = 13;  
        s1.num2 = 23;  
        s2.num1 = 33;  
        s2.num2 = 43;  
        System.out.println(s1.num1 + " " + s1.num2 + " " + s2.num1 + " " + s2.num2);  
    }  
}
```

Output nedir?

- A) 13 43 33 43
- B) 13 23 33 43
- C) 33 23 33 43
- D) 13 43 13 43

Static Keyword

```
9. public class Static03 {  
    static int counter = 0;  
    int age = 0;  
  
    public Static03() {  
        counter++;  
        age++;  
    }  
    public static void main(String[] args) {  
        Static03 obj1 = new Static03();  
        Static03 obj2 = new Static03();  
        Static03 obj3 = new Static03();  
        System.out.println(obj1.counter);  
        System.out.println(obj2.counter);  
        System.out.println(obj3.counter);  
        System.out.print(" - ");  
        System.out.println(obj1.age);  
        System.out.println(obj2.age);  
        System.out.println(obj3.age);  
    }  
}
```

Output nedir?

- A) 333 - 111
- B) 333 - 333
- C) 123 - 111
- D) 123 - 123

```
10. public class Test {  
    String studentName;  
    String year = "2020";  
    static int counter = 1000;  
    String studentId;  
    public Test() {  
        Scanner scan = new Scanner(System.in);  
        System.out.println("Enter your name");  
        studentName = scan.next();  
        setStudentId();  
    }  
    public static void main(String[] args) {  
        Test student1 = new Test();  
        System.out.println(student1.studentName + student1.studentId); //A  
  
        Test student2 = new Test();  
        System.out.println(student2.studentName + student2.studentId); //B  
    }  
    public void setStudentId() {  
        counter++;  
        this.studentId = year + "" + counter;  
    }  
}
```

Kullanıcı ilk student name için "Ali Can" ve ikinci student name için "Veli Han" girerse A ve B output'ları ne olur?

- A) A ==> Ali20201001
B ==> Veli20201002
- B) A ==> Ali20201001
B ==> Veli20201001
- C) A ==> Ali Can20201001
B ==> Veli Han20201002
- D) A ==> Ali Can20201001
B ==> Veli Han20201001

1. Note: Array oluşturma'nın 2 yolu vardır;

1) `int arr[] = new int[4];`

`arr[0]=11; arr[1]=12; arr[2]=13; arr[3]=14;`

2) `int arr[] = {11, 12, 13, 14};`

`int arr[] = {11, 12, 13, 14};`

Hangisi yanlıştır?

A) `System.out.println(Arrays.toString(arr));`

Ekrana [11, 12, 13, 14] yazdırır.

B) `System.out.println(arr[arr.length-2]);`

Ekrana 13 yazdırır.

C) `System.out.println(arr[1]);`

Ekrana 11 yazdırır.

D) `for(int i=0; i<4; i++) {`

`System.out.print(arr[i] + " ");`

`}`

Ekrana 11 12 13 14 yazdırır.

2. `int arr[] = {2, 11, 0, 23, 7};`

Hangisi ekrana en büyük elemanı yazdırır?

A) `Arrays.sort(arr);`

`System.out.print(arr[arr.length-1]);`

B) `Arrays.sort();`

`System.out.print(arr[arr.length-1]);`

C) `Array.sort(arr);`

`System.out.print(arr[arr.length-1]);`

D) `Arrays.sort(arr);`

`System.out.print(arr[arr.length]);`

3. `char arrCh[] = {'a', 'C', 'c', 'A', 'b'};`
`Arrays.sort(arrCh);`
`System.out.println(Arrays.toString(arrCh));`

Output nedir?

- A) [A, C, a, b, c]
- B) [A, a, b, C, c]
- C) [a, b, c, A, C]
- D) [a, C, c, A, b]

4. `int arr[] = {2, 11, 0, 23, 7};`
`Arrays.sort(arr);`
`for(int i=4; i>=0; i--) {`
 `System.out.print(arr[i] + " ");`
`}`

Output nedir?

- A) 0 2 7 11 23
- B) 23 11 7 2 0
- C) 2 11 0 23 7
- D) Gives Compile Time Error

Static Block Arrays

5. `public class Try {
 static int a = 11;
 public static void main(String[] args) {
 int b = 12;
 System.out.println(b);
 }
 static{
 a = 13;
 System.out.println(a);
 }
}`

Output nedir?

- A) 13
12
- B) 11
12
- C) 12
13
- D) 11
13

6. **Which ones are true for Arrays?**

- A) Arrays can store primitive data types and references
- B) If the index of last element is 5 in an array, it means we can store 5 elements at most inside the array
- C) Data types of all elements must be same in an array
- D) `int arr[] = new int[];` is the syntax to create an array in Java

Static Block Arrays

7. `public class Try {
 static int a = 11;
 public static void main(String[] args) {
 int b = 12;
 System.out.println(b);
 }
 static{
 int c = 13;
 System.out.println(c);
 }
 static{
 a = 14;
 System.out.println(a);
 }
}`

Output nedir?

- A) 13
14
12
- B) 12
13
14
- C) 12
14
13
- D) 13
11
12

Static Block Arrays

8. `public class Try {
 public static void main(String[] args) {
 int arr[] = new int[3];
 System.out.println(arr[1]);

 arr[0] = 11;
 arr[1] = 12;
 System.out.println(Arrays.toString(arr));
 }
}`

Output nedir?

- A) 0
[11, 12, 0]
- B) 0
[11, 12]
- C) [0, 0, 0]
11
12
- D) Gives Compile Time Error

9. `char arr[] = new char[4];
 arr[0] = 'A';
 arr[2] = 'E';
 arr[3] = 'M';
System.out.println(Arrays.toString(arr));`

Output nedir?

- A) [A, , E, M]
- B) [A, 0, E, M]
- C) [A, E, M]
- D) Gives Compile Time Error

```
10. int arr[] = {3, 2, 3, 4, 4, 5};  
    int count = 0;  
    for(int i=0; i<arr.length; i++) {  
        if(arr[i]==3) {  
            count++;  
        }  
    }  
    System.out.println(count);
```

Output nedir?

- A) 0
- B) 1
- C) 2
- D) 3

```
11. public class Try {  
    public static void main(String[] args) {  
        int x[] = { 3, 5, 7 };  
        int a = 0;  
        int b = 0;  
        while (a < x.length) {  
            b = b + x[a];  
            a++;  
        }  
        System.out.println(b);  
    }  
}
```

Output nedir?

- A) [3, 5, 7]
- B) 15
- C) 10
- D) 3 5 7

1. `int arr[] = {11, 12, 13, 14};`

Yukarıda verilen array ile ilgili aşağıdakilerden hangisi yanlıştır?

A) `System.out.println(Arrays.toString(arr));`
Ekranı [11, 12, 13, 14] yazdırır.

B) `System.out.println(arr[arr.length-2]);`
Ekranı 13 yazdırır.

C) `System.out.println(arr[1]);`
Ekranı 11 yazdırır.

D) `for(int i=0; i<4; i++) {`
 `System.out.print(arr[i] + " ");`
 `}`
Ekranı 11 12 13 14 yazdırır.

2. `int arr[] = {2, 11, 0, 23, 7};`

Yukarıda verilen Array'in en büyük elemanını ekrana yazdıran kod aşağıdakilerden hangisidir?

A) `Arrays.sort(arr);`
 `System.out.print(arr[arr.length-1]);`

B) `Arrays.sort();`
 `System.out.print(arr[arr.length-1]);`

C) `Array.sort(arr);`
 `System.out.print(arr[arr.length-1]);`

D) `Arrays.sort(arr);`
 `System.out.print(arr[arr.length]);`

3. `char arrCh[] = {'a', 'C', 'c', 'A', 'b'};`
`Arrays.sort(arrCh);`
`System.out.println(Arrays.toString(arrCh));`

Ekran çıktısı nedir?

- A) [A, C, a, b, c]
- B) [A, a, b, C, c]
- C) [a, b, c, A, C]
- D) [a, C, c, A, b]

4. `int arr[] = {2, 11, 0, 23, 7};`
`Arrays.sort(arr);`
`for(int i=4; i>=0; i--) {`
 `System.out.print(arr[i] + " ");`
`}`

Yukarıda verilen kodun ekran çıktısı nedir?

- A) 0 2 7 11 23
- B) 23 11 7 2 0
- C) 2 11 0 23 7
- D) Compile Time Error verir

5. `int arr1[] = {1, 2, 3};`
`int arr2[] = {1, 2, 3};`

Aşağıdakilerden hangileri ekrana true yazdırır?

- A) `System.out.println(arr1==arr2);`
- B) `System.out.println(arr1==arr1);`
- C) `System.out.println(Arrays.equals(arr1, arr2));`
- D) `System.out.println(arr1.equals(arr2));`

6. `int arr[] = {1, 23, 12, 2, 3};`

Yukarıda verilen array'de 12'nin eleman olarak var olup olmadığını kontrol etmek için aşağıdaki kodlardan hangisi yazılmalıdır?

- A) `Arrays.sort(arr);`
`System.out.println(Arrays.binarySearch(arr, 12));`
- B) `System.out.println(Arrays.binarySearch(arr, 12));`
`Arrays.sort(arr);`
- C) `System.out.println(Arrays.binarySearch(12));`
`System.out.println(Arrays.binarySearch(12));`

7. `int arr[] = {12, 23, 12, 2, 3};`

`Arrays.sort(arr);`
`System.out.println(Arrays.binarySearch(arr, 12));`

Yukarıdaki kod ile ilgili aşağıdakilerden hangisi doğrudur?

- A) Kesin bir sonuç alamayız
- B) 3
- C) [2, 3]
- D) Compile Time Error verir.

8. `int arr[] = {12, 23, 12, 2, 3};`

`Arrays.sort(arr);`
`System.out.println(Arrays.binarySearch(arr, 5));`

Ekran çıktısı nedir?

- A) Compile Time Error
- B) `ArrayIndexOutOfBoundsException`
- C) -3
- D) -2

9. `int arr[] = {3, 2, 3, 4, 4, 5};`
`int count = 0;`
`for(int i=0; i<arr.length; i++) {`
 `if(arr[i]==3) {`
 `count++;`
 `}`
`}`

`System.out.println(count);`

Ekran çıktısı nedir?

- A) 0
- B) 1
- C) 2
- D) 3

10. `char arr[] = new char[4];`
`arr[0] = 'A';`
`arr[2] = 'E';`
`arr[3] = 'M';`
`System.out.println(Arrays.toString(arr));`

Ekran çıktısı nedir?

- A) [A, , E, M]
- B) [A, 0, E, M]
- C) [A, E, M]
- D) Compile Time Error

11. Aşağıdakilerden hangisi yanlıştır?

- A)

```
public static void main(String[] args) {  
    String str = "Cok calis, mutevazi ol";  
    String strArray[] = str.split("");  
    System.out.println(Arrays.toString(strArray));  
}
```


Output [Cok calis, mutevazi ol]
- B)

```
public static void main(String[] args) {  
    String str = "Cok calis, mutevazi ol";  
    String strArray[] = str.split(" "); // çift tırnak içinde space var  
    System.out.println(Arrays.toString(strArray));  
}
```


Output [Cok, calis,, mutevazi, ol]
- C)

```
public static void main(String[] args) {  
    String str = "Cok calis, mutevazi ol";  
    String strArray[] = str.split("X");  
    System.out.println(Arrays.toString(strArray));  
}
```


Output [Cok calis, mutevazi ol]
- D) Hiçbiri

```
1. public class Test01 {  
    public static void main(String[] args) {  
        int[][] x = {{2, 1}, {1, 7, 1}};  
        System.out.println(m(x[1]));  
    }  
  
    public static int m(int[] m) {  
        int result = 0;  
        for (int i = 0; i < m.length; i++){  
            result += m[i];  
        }  
        return result;  
    }  
}
```

Output nedir?

- A) 8
- B) 9
- C) 10
- D) 11



Multi Dimensional Arrays

```
2. public class Test02 {  
    public static void main(String[] args) {  
        int[][] x = {{1, 2}, {3, 4, 5}};  
        System.out.println(m(x));  
    }  
  
    public static int m(int[][] m) {  
        int result = 3;  
        for (int i = 0; i < m.length; i++) {  
            for (int j = 0; j < m[i].length; j++) {  
                result += m[i][j];  
            }  
        }  
        return result;  
    }  
}
```

Output nedir?

- A) 15
- B) 16
- C) 17
- D) 18

Multi Dimensional Arrays

```
3. public class Test03 {  
    public static void main(String[] args) {  
        int[][] x = {{2, 1}, {5, 7, 6}};  
        System.out.println(m(x));  
    }  
  
    public static int m(int[][] m) {  
        int result = m[0][1];  
        for (int i = 0; i < m.length; i++) {  
            for (int j = 0; j < m[i].length; j++) {  
                if (result < m[i][j]) {  
                    result = m[i][j];  
                }  
            }  
        }  
        return result;  
    }  
}
```

Output nedir?

- A) 1
- B) 2
- C) 6
- D) 7

Multi Dimensional Arrays

```
4. public class Test04 {  
    public static void main(String[] args) {  
        int arr[][] = { {3,2,1},{1,2,3} };  
        for (int i = 1; i < arr.length; i++) {  
            for (int k = 1; k < arr[0].length; k++) {  
                if (arr[i][k] % 2 == 1){  
                    arr[i][k] = arr[i][k] + 1;  
                }  
                if (arr[i][k] % 2 == 0){  
                    arr[i][k] = arr[i][k] * 2;  
                }  
            }  
        }  
        System.out.println(Arrays.deepToString(arr));  
    }  
}
```

Output nedir?

- A) [[3, 2, 1], [1, 4, 8]]
- B) [[4, 3, 2], [1, 4, 8]]
- C) [[4, 3, 2], [1, 2, 3]]
- D) [[3, 2, 1], [1, 2, 3]]

Multi Dimensional Arrays

```
5. public class Test05 {  
    public static void main(String[] args) {  
        int arr[][] = {{5, 12}, {3, 8, 4}, {6, 3}};  
        System.out.println(m(arr));  
    }  
  
    public static int m(int[][] arr) {  
        int r, c;  
        int i, k = 0;  
  
        for (r = 0; r < arr.length; r++) {  
            for (c = 0; c < arr[r].length; c++) {  
                i = arr[r][c];  
                if (i > k)  
                    k = i;  
            }  
        }  
        return k;  
    }  
}
```

Output nedir?

- A) 12
- B) 8
- C) 3
- D) 5

```
1. public static void main(String[] args) {  
    ArrayList<String> list = new ArrayList<String>();  
    list.add( "A" );  
    list.add( "B" );  
    list.add( "F" );  
    list.add( "D" );  
}
```

Hangisi doğrudur?

- A) list.set(2, "C");
System.out.println(list);
Ekрана [A, B, C, D] yazar
- B) list.add(1, "*");
System.out.println(list);
Ekрана [*, A, B, C, D] yazar
- C) System.out.println(list.contains("F"));
Ekрана false yazar
- D) System.out.println(list.isEmpty());
Ekрана true yazar

Array Lists

2.

```
public static void main(String[] args) {  
    ArrayList<String> list = new ArrayList<String>();  
    list.add("A");  
    list.add("B");  
    list.add("C");  
    list.add("D");  
    list.remove(3);  
    list.remove("A");  
    System.out.println(list);  
}
```

Output nedir?

- A) [B, C]
- B) Run Time Error
- C) [B, C, D]
- D) Compile Time Error

3.

```
public static void main(String[] args) {  
    ArrayList<String> list = new ArrayList<String>();  
    list.add("A");  
    list.add("B");  
    list.add("C");  
    list.add("D");  
    for (int i = 0; i < list.size(); i++) {  
        System.out.print(list.get(i) + " ");  
    }  
}
```

Output nedir?

- A) A B C D
- B) [A, B, C, D]
- C) Run Time Error
- D) Compile Time Error

Array Lists

4.

```
public static void main(String[] args) {  
    ArrayList<String> list = new ArrayList<String>();  
    list.add("A");  
    list.add("B");  
    list.add("C");  
    list.add("D");  
    System.out.println(list.remove(2));  
    System.out.println(list.remove("C"));  
}
```

Output nedir?

- A) C
false
- B) false
C
- C) true
false
- D) C
C

5.

```
public static void main(String[] args) {  
    int arr[] = new int[5];  
    ArrayList<Integer> list = new ArrayList<Integer>();  
    for (int i = 1; i <= arr.length; i++) {  
        list.add(i);  
    }  
    System.out.println(list);  
}
```

Output nedir?

- A) [1, 2, 3, 4, 5]
- B) []
- C) Compile Time Error
- D) [1, 2, 3, 4]

Array Lists

6.

```
public class Test01 {  
    List<String> list1 = new ArrayList<String>();  
    public static void main(String[] args) {  
        Test01 obj1 = new Test01();  
        obj1.myMethod(obj1.list1);  
        obj1.list1.add("z");  
        obj1.list1.add("t");  
        System.out.println(obj1.list1);  
    }  
    public List<String> myMethod(List<String> list1) {  
        list1.add("x");  
        list1.add("y");  
        return list1;  
    }  
}
```

Output nedir?

- A) [x, y, z, t]
- B) [x, y]
- C) [z, t]
- D) Compile Time Error

7.

```
public static void main(String[] args) {  
    ArrayList<Character> list = new ArrayList<Character>();  
    for (char i = 'a'; i <= 'e' ; i++) {  
        list.clear();  
        list.add(i);  
    }  
    System.out.println(list);  
}
```

Output nedir?

- A) []
- B) [e]
- C) [a, b, c, d, e]
- D) Compile Time Error

8. Hangileri Compile Time Error verir?

- A) `List<Character> list = new ArrayList<Character>();`
- B) `List<int> list = new ArrayList<>();`
- C) `ArrayList<> list = new ArrayList<String>();`
- D) `ArrayList<boolean> list = new ArrayList<>();`

9. public static void main(String[] args) {

```
    List<Integer> list = new ArrayList<>();
```

```
    list.add(3);
```

```
    list.add(5);
```

```
    list.add(7);
```

```
    list.add(9);
```

```
    int i = 0;
```

```
    int x = 0;
```

```
    while(i<list.size()) {
```

```
        x = x + list.get(i);
```

```
        i++;
```

```
    }
```

```
    System.out.println(x);
```

```
}
```

Output nedir?

- A) 24
- B) Compile Time Error
- C) 9
- D) 15

```
10. public static void main(String[] args) {  
    int arr[] = {2, 4, 6, 8};  
    List<Integer> list = new ArrayList<>();  
    int i = 0;  
    do {  
        list.add(arr[i]);  
        i++;  
    } while (i < arr.length);  
    System.out.println(list);  
}
```

Yukarıdaki kod ne yapıyor?

- A) Array'i list'e çeviriyor
- B) Compile Time Error verir
- C) List'i array'e çeviriyor
- D) Run Time Error verir

```
11. List<Integer> list2 = new ArrayList<>();  
list2.add(5);  
list2.add(2);  
list2.add(4);  
list2.add(3);  
list2.add(1);  
list2.remove(4);  
System.out.println(list2);
```

Output nedir?

- A) [5, 2, 4, 3]
- B) [5, 2, 3, 1]
- C) [5, 2, 3]
- D) Compile Time Error

1.

```
public static void main(String[] args) {  
    int arr[] = { 1, 3, 5 };  
    int x = 0;  
    for (int w : arr) {  
        x = x + w * w;  
    }  
    System.out.print(x);  
}
```

Output nedir?

- A) 35
- B) 9
- C) 23
- D) Compile Time Error

2.

```
public static void main(String[] args) {  
    int arr[] = { 1, 3, 5, 7, 9 };  
    for (int w : arr) {  
        if(w<4) {  
            continue;  
        }  
        System.out.print(w + " ");  
    }  
}
```

Output nedir?

- A) 1 3 5 7 9
- B) 1 3
- C) 5 7 9
- D) None

For Each Loops

3.

```
public static void main(String[] args) {  
    int arr[] = { 1, 3, 5, 7, 9 };  
    for (int w : arr) {  
        System.out.print(w + " ");  
        if(w>4) {  
            break;  
        }  
    }  
}
```

Output nedir?

- A) 1 3 5
- B) 1 3
- C) 1 3 5 7 9
- D) None

4.

```
public static void main(String[] args) {  
    int arr[][] = {{ 1, 2 }, { 3 }, { 4, 5, 6 }};  
    for (int[] w : arr) {  
        for (int z : w) {  
            System.out.print(z + " ");  
        }  
    }  
}
```

Output nedir?

- A) 1 2 3 4 5 6
- B) [1, 2] [3] [4, 5, 6]
- C) 1 2
- D) [1, 2]

For Each Loops

5.

```
public static void main(String[] args) {  
    int arr[][] = {{ 1, 2 }, { 3 }, { 4, 5, 6 }};  
    int x = 0;  
    for (int[] w : arr) {  
        for (int z : w) {  
            x = x + z;  
        }  
    }  
    System.out.println(x);  
}
```

Output nedir?

- A) 21
- B) 1 2 3 4 5 6
- C) 3
- D) 1 2

6.

```
List<Integer> list1 = new ArrayList<>();  
list1.add(21);  
list1.add(22);  
list1.add(23);  
list1.add(24);  
for(int w : list1) {  
    if(w%2==0) {  
        continue;  
    }  
    System.out.print(w+ " ");  
}
```

Output nedir?

- A) 21 23
- B) 22 24
- C) 21 22 23 24
- D) None

For Each Loops

7.

```
public static void main(String[] args) {  
    int arr[] = {3, 4, 5, 6};  
    List<Integer> list1 = new ArrayList<>();  
    for(int w : arr) {  
        list1.add(w);  
    }  
    System.out.println(list1.get(2));  
}
```

Output nedir?

- A) 5
- B) 4
- C) Run Time Error
- D) Compile Time Error

8.

```
public static void main(String[] args) {  
    int arr[] = {3, 4, 5, 6};  
    List<Integer> list1 = new ArrayList<>();  
    for(int w : arr) {  
        list1.add(w);  
    }  
}
```

Hangisi aynı output'u verir?

- A)

```
int arr[] = {3, 4, 5, 6};  
List<Integer> list1 = Arrays.asList(arr);
```
- B)

```
int arr[] = {3, 4, 5, 6};  
List<Integer> list1 = Arrays.toList(arr);
```
- C)

```
int arr[] = {3, 4, 5, 6};  
List<Integer> list1 = asList(arr);
```
- D)

```
int arr[] = {3, 4, 5, 6};  
List<Integer> list1 = toList(arr);
```

For Each Loops

9. `List<Integer> list = new ArrayList<>();`
`list.add(21);`
`list.add(22);`
`list.add(23);`

Hangileri doğrudur?

A) `int count = 0;`
`for(int w : list) {`
 `System.out.print(w + " ");`
 `count++;`
 `if(count==2) {`
 `break;`
 `}`
`}`

Prints 21 22 on the console

B) `int count = 0;`
`for(int w : list) {`
 `count++;`
 `if(count==2) {`
 `continue;`
 `}`
 `System.out.print(w + " ");`
`}`

Prints 21 23 on the console

C) `int count=0;`
`for(int w : list) {`
 `count++;`
 `if(count<3) {`
 `System.out.print(w + " ");`
 `}`
`}`

Prints 21 22 on the console

For Each Loops

```
10. public class Test01 {  
    static int x = 1000;  
    public static void main(String[] args) {  
  
        Scanner scan = new Scanner(System.in);  
        List<String> products = new ArrayList<>();  
        int i=0;  
        do {  
            System.out.println("Enter " + (i+1) + ". product name");  
            products.add(scan.nextLine());  
            i++;  
        }while(i<3);  
        System.out.println(products);  
        List<String> productId = new ArrayList<>();  
        for(String w : products) {  
            productId.add("" + w.charAt(0) + w.charAt(1) + x);  
            x++;  
        }  
        System.out.println(productId);  
        scan.close();  
    }  
}
```

Kullanıcı Toy, Lego, ve Shoes girerse output ne olur?

- A) [Toy, Lego, Shoes]
[To1000, Le1001, Sh1002]
- B) [Toy, Lego, Shoes]
[To1000, Le1000, Sh1000]
- C) [Toy, Lego, Shoes]
[To, Le, Sh]
- D) [Toy, Lego, Shoes]
[T1000, L1001, S1002]

1. `LocalDate date = LocalDate.now();`

Eğer tarih 29 Haziran 2020 ise hangileri doğrudur?

- A) `System.out.println(date);`
Output 2020-06-29
- B) `System.out.println(date.plusMonths(2));`
Output 2020-08-29
- C) `System.out.println(date.minusDays(5));`
Output 2020-06-24
- D) `DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd/MMM/yyyy");`
`System.out.println(dtf.format(date));`
Output 29/Haz/2020

2. `LocalTime time = LocalTime.now();`

Eğer saat 09:34:02.960 PM ise hangileri doğrudur?

- A) `System.out.println(time.plusMinutes(11));`
Output 09:45:02.960
- B) `System.out.println(time.minusHours(2));`
Output 07:34:02.960
`DateTimeFormatter dtf = DateTimeFormatter.ofPattern("hh:mm");`
- C) `System.out.println(dtf.format(time));`
Output 09:34
- D) `DateTimeFormatter dtf = DateTimeFormatter.ofPattern("HH:mm");`
`System.out.println(dtf.format(time));`
Output 21:34

3. `public class A{
 public static void main(String[] args){
 int x = 5;
 change(x);
 System.out.println(x);
 }
 public static void change(int x){
 x = 20;
 }
}`

Output nedir?

- A) 5
- B) 20
- C) Compile Time Error
- D) 0

4. `class Test{
 int x;
 Test(int i) { x = i; }
 Test() { x = 0; }
}

class A{
 public static void main(String[] args){
 Test t = new Test(5);
 change(t);
 System.out.println(t.x);
 }
 public static void change(Test t){
 t = new Test();
 t.x = 10;
 }
}`

Output nedir?

- A) 5
- B) 10
- C) 0

```
5. class Test{
    int x;
    Test(int i) { x = i; }
    Test() { x = 0; }
}

class A{
    public static void main(String[] args){
        Test t = new Test(5);
        change(t);
        System.out.println(t.x);
    }
    public static void change(Test t){
        t.x = 10;
    }
}
```

Output nedir?

- A) 5
- B) 10
- C) 0

6. `LocalDate date = LocalDate.now();`
Eğer tarih 1 Temmuz 2020 ise hangisi yanlıştır?

- A) `DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd:MMM:yy");`
`System.out.println(dtf.format(date));`
Output 01:Jul:20
- B) `DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd/MMMM/yyyy");`
`System.out.println(dtf.format(date));`
Output 01/Temmuz/2020
- C) `DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd/MM/yyyy");`
`System.out.println(dtf.format(date.plusMonths(2)));`
Output 01/09/2020
- D) None

```
7. for(int i=1; i<4; i++) {  
    for(int k=3; k>1; k--) {  
        System.out.println(i + "==>" + k);  
    }  
}
```

Output nedir?

- A) 1==>3
1==>2
2==>3
2==>2
3==>3
3==>2
- B) 1==>3==>2
2==>3==>2
3==>3==>2
- C) 1==>3
2==>3
3==>3
- D) 3==>1
2==>1
3==>2
2==>2
3==>3
2==>3



8. `int a[] = {1, 2};`
`int b[] = {3, 2, 1};`
`for(int w : a) {`
 `for(int m : b) {`
 `System.out.print(w + m + " ");`
 `}`
`}`

Output nedir?

- A) 4 3 2 5 4 3
- B) 21
- C) 5 4 3 4 3 2
- D) 8

9. `LocalDate currentDate = LocalDate.now();`
`System.out.println(currentDate.plusDays(3).minusMonths(4).plusYears(2));`

Eğer tarih 2020-10-08 ise output nedir?

- A) 2022-06-11
- B) 2023-02-11
- C) 2022-10-11
- D) 2022-10-08

10. Tarih 2020-10-08 ve saat 03:00:00pm.

Hangisi yanlıştır?

- A) `LocalDate date = LocalDate.now();`
`DateTimeFormatter dtf =`
`DateTimeFormatter.ofPattern("dd-MMMM-yy");`
`System.out.println(dtf.format(date));`
Output 08-Ekim-20
- B) `LocalDate date = LocalDate.now();`
`DateTimeFormatter dtf = DateTimeFormatter.ofPattern("MM/dd/yyyy");`
`System.out.println(dtf.format(date));`
Output 10/08/2020
- C) `LocalTime time = LocalTime.now();`
`DateTimeFormatter dtf = DateTimeFormatter.ofPattern("HH:mm");`
`System.out.println(dtf.format(time.plusHours(1)));`
Output 04:00
- D) `LocalTime time = LocalTime.now();`
`DateTimeFormatter dtf = DateTimeFormatter.ofPattern("hh:mm:ss");`
`System.out.println(dtf.format(time.minusHours(1)));`
Output 02:00:00

1. `public static int m1(boolean b1, boolean... b2) {
 return b2.length;
}`

Hangisi 2 return eder?

- A) `m1(true, false, true);`
- B) `m1(false, true);`
- C) `m1(true, false, true, true);`
- D) `m1();`

2. `public static void m2(int i, int j, int... k) {
 System.out.println(k.length);
}`

Hangisi yanlıştır?

- A) `m2(1, 2, 3, 4);`
Output 4
- B) `m2(1, 2, 3, 4, 5);`
Output 3
- C) `m2(1, 2);`
Output 0
- D) `m2(1);`
Compile Time Error

3. `public static void m3(String x, int... y) {`
 `int p = 1;`
 `for(int w: y) {`
 `p = p * w;`
 `}`
 `System.out.print(x);`
 `System.out.println(p);`
`}`

Hangisi yanlıştır?

- A) `m3("Result: ", 2, 3, 4);`
Output Result: 24
- B) `m3("Result: ", 2, 2.5);`
Output Result: 5
- C) `m3("Result: ");`
Output Result: 1
- D) `m3("Result: ", 2);`
Output Result: 2

4. **Hangileri Compile Time Error verir?**

- A) `public static void m4(int i, int... j, int... k) {`
 `System.out.println(k.length);`
 `}`
- B) `public static void m4(int... j, int k) {`
 `System.out.println(j.length);`
 `}`
- C) `private class Test02 {`
 `}`
- D) `protected class Test02 {`
 `}`

Varargs Access Modifiers

```
5. public class Test1 {  
    public static void main(String args[]) {  
        m1(10);  
        m1(11, 12, 13, 14);  
    }  
    static void m1(int... a) {  
        System.out.println("==> " + a.length);  
        for (int i : a) {  
            System.out.print(i + " ");  
        }  
        System.out.println();  
    }  
}
```

Output nedir?

- A) ==> 1
10
==> 4
11 12 13 14
- B) ==> 10
1
- C) ==> 11 12 13 14
4
- D) ==> 10
==> 11 12 13 14



Varargs Access Modifiers

6. `public class Test2 {
 static void m2(String str, int... a) {
 System.out.println(str);
 for (int i : a) {
 System.out.print(i + " ");
 }
 System.out.println();
 }
 public static void main(String args[]) {

 }
}`

***** yerine aşağıdaki kodlar yazılırsa hangileri doğru olur?

- A) `m2("Java", 101, 102);`
Output Java
101 102
- B) `m2(100, 101, 102);`
Output 100
101 102
- C) `m2(101, 102);`
Output 101 102
- D) `m2(100, 101, "102");`
Compile Time Error

7. Hangileri default ve protected access modifier'lerin farklarındandır?

- A) Bir class member'i default yapmak için herhangi bir access modifier yazmayız. Fakat bir class member'i protected yapmak için "protected" yazmak zorundayız.
- B) default class member'lara diğer package'lardan ulaşamaz fakat protected olanlara diğer package'lardaki child class'lardan ulaşılabilir.
- C) protected access modifier'in diğer ismi package private'dir.
- D) default access modifier'in başka ismi yoktur.

8. Hangileri Compile Time Error?

A) package pack1;
class A{ }

```
package pack2;  
class B {  
    public static void main(String[] args) {  
        A obj = new A();  
    }  
}
```

B) package pack1;
public class A{ }

```
package pack2;  
class B {  
    public static void main(String[] args) {  
        A obj = new A();  
    }  
}
```

C) package pack1;
public class A{ }

```
package pack2;  
public class B {  
    public static void main(String[] args) {  
        A obj = new A();  
    }  
}
```

D) package pack1;
class A{ }

```
package pack2;  
public class B {  
    public static void main(String[] args) {  
        A obj = new A();  
    }  
}
```

9. `package pack1;`
`public class Test3 {`
 `protected int i = 12;`
 `void m3() {`
 `System.out.println(i);`
 `}`
`}`

`package pack1;`
`public class Test4 {`
 `public static void main(String[] args) {`
 `Test3 obj = new Test3();`
 `System.out.println(obj.i);`
 `obj.m3();`
 `}`
`}`

Output nedir?

- A) 12
 12
- B) 12
- C) Compile Time Error
- D) Run Time Error

10. Hangileri Compile Time Error verir?

- A) `public class Test3 {}`
- B) `private class Test3 {}`
- C) `class Test3 {}`
- D) `protected class Test3 {}`

1. `StringBuilder strBld = new StringBuilder(6);`
`strBld.append("Learn");`
`strBld.append("Java");`
`System.out.println(strBld);`

Output?

- A) LearnJ
- B) LearnJava
- C) Run Time Error
- D) Compile Time Error

2. `StringBuilder strBld = new StringBuilder("LearnJava");`

Hangileri doğrudur?

- A) `strBld.substring(3);`
`System.out.println(strBld);`
Output LearnJava
- B) `strBld.insert(0,"You");`
`System.out.println(strBld);`
Output YouLearnJava
- C) `strBld.reverse();`
`System.out.println(strBld);`
Output avaJnraeL
- D) `strBld.delete(5, 9);`
`System.out.println(strBld);`
Output Learn

String Builder

3. `StringBuilder strBld = new StringBuilder("John ");`
`strBld.append("Woo ").append("Leo").deleteCharAt(6);`
`System.out.println(strBld);`

Output?

- A) John Wo Leo
- B) John Woo Leo
- C) John oo Leo
- D) John W

4. **Hangisi yanlıştır?**

- A) `public static void main(String[] args) {`
 `StringBuilder str = new StringBuilder("LearnJava");`
 `str.deleteCharAt(3);`
 `System.out.println(str.substring(2, 6));`
`}`

Output anJa

- B) `public static void main(String[] args) {`
 `StringBuilder str = new StringBuilder("LearnJava");`
 `str.setCharAt(5, 'L');`
 `System.out.println(str);`
`}`

Output LearnLava

- C) `public static void main(String[] args) {`
 `StringBuilder str = new StringBuilder(7);`
 `str.append("Java");`
 `System.out.println(str.capacity() + " " + str.length());`
`}`

Output 4,7

- D) None

5. Hangileri doğrudur?

A)

```
public static void main(String[] args) {  
    StringBuilder str = new StringBuilder("LearnJava");  
    int index = str.indexOf("earn");  
    System.out.println(index);  
}
```

Output 1

B)

```
public static void main(String[] args) {  
    StringBuilder str = new StringBuilder("LearnJava");  
    int index = str.indexOf("a", 4);  
    System.out.println(index);  
}
```

Output 6

C)

```
public static void main(String[] args) {  
    StringBuilder str = new StringBuilder("LearnJava");  
    int index = str.indexOf("e", 4);  
    System.out.println(index);  
}
```

Compile Time Error

6.

```
public static void main(String[] args) {  
    StringBuilder str = new StringBuilder("LearnJava");  
    str.reverse();  
    System.out.println(str);  
}
```

Hangisinin output'u yukarıdaki ile aynıdır?

A)

```
public static void main(String[] args) {  
    String str2 = "LearnJava";  
    String str3 = "";  
    for(int i = str2.length()-1; i>=0; i--) {  
        str3 = str3 + str2.charAt(i);  
    }  
    System.out.println(str3);  
}
```

B)

```
public static void main(String[] args) {  
    String str2 = "LearnJava";  
    String str3 = "";  
    for(int i = 0; i<=str2.length()-1; i++) {  
        str3 = str3 + str2.charAt(i);  
    }  
    System.out.println(str3);  
}
```

C)

```
public static void main(String[] args) {  
    String str2 = "LearnJava";  
    String str3 = "";  
    for(int i = str2.length()-1; i>=0; i++) {  
        str3 = str3 + str2.charAt(i);  
    }  
    System.out.println(str3);  
}
```

String Builder

7. **Bilgi:** append(char[] ch) method'u char array eklemek için kullanılır.

```
public static void main(String[] args) {  
    StringBuilder stb = new StringBuilder("Learn ");  
    char[] ch = new char[] { 'J', 'A', 'V', 'A' };  
    stb.append(ch);  
    System.out.println(stb);  
}
```

Output nedir?

- A) Learn JAVA
- B) JAVA
- C) JAVA Learn
- D) Learn

8. **Hangileri doğrudur?**

A)

```
public static void main(String[] args) {  
    String str = "Learn";  
    str = str + "Java";  
    System.out.println(str);  
}
```

Output LearnJava

B)

```
public static void main(String[] args) {  
    String str = "Learn";  
    System.out.println(str + "Java");  
}
```

Output LearnJava

C)

```
public static void main(String[] args) {  
    String str = "Learn";  
    join("Java");  
    System.out.println(str);  
}  
  
public static String join(String str) {  
    return str + "Java";  
}
```

Output LearnJava

- D) None

9. `StringBuilder sb3 = new StringBuilder("Learn");`
`System.out.println("Before trim: " + sb3.capacity());`
`sb3.trimToSize();`
`System.out.println("After trim: " + sb3.capacity());`

Output nedir?

- A) Before trim: 21
After trim: 5
- B) Before trim: 5
After trim: 5
- C) Before trim: 5
After trim: 21
- D) Before trim: 16
After trim: 5

10. Hangileri doğrudur?

- A) `StringBuilder sb = new StringBuilder(9);`
`System.out.println(sb.length() + " - " + sb.capacity());`
Output 0 - 9
- B) `StringBuilder sb = new StringBuilder();`
`System.out.println(sb.length() + " - " + sb.capacity());`
Output 0 - 16
- C) `StringBuilder sb = new StringBuilder("Java");`
`System.out.println(sb.length() + " - " + sb.capacity());`
Output 4 - 20
- D) `StringBuilder sb = new StringBuilder("StringBuilder");`
`System.out.println(sb.delete(3, 6).toString() + " - " + sb.length());`
Output StrBuilder - 10

```
1. public class Encapsulation {  
    private int ssn = 100003123;  
    private String disease = "Cancer";  
    private double salary = 200000;  
  
    public int getSsn() {  
        return ssn;  
    }  
    public void setSsn(int ssn) {  
        this.ssn = ssn;  
    }  
    public void setDisease(String disease) {  
        this.disease = disease;  
    }  
    public double getSalary() {  
        return salary;  
    }  
}
```

Hangileri doğrudur?

- A) ssn, disease, ve salary variables encapsulated yapılmıştır.
- B) ssn ve salary variable'lerinin degerleri diğer class'lardan okunabilir.
- C) salary variable'inin değeri diğer class'lardan değiştirilebilir.
- D) disease variable'inin değeri diğer class'lardan okunabilir.

2. Encapsulation için hangisi yanlıştır?

- A) Static ve instance variable'lar encapsulated yapılabilir.
- B) Sadece private variable'lar encapsulated yapılabilir.
- C) getters methodlar variable degerlerini diğer class'lardan okumak için kullanılır.
- D) setter methodlar variable degerlerini diğer class'lardan değiştirmek için kullanılır.

Encapsulation

```
3. public class Encapsulation {  
    private int ssn = 1234;  
    private String disease = "Headahce";  
    private double salary = 100000;  
  
    public int getSsn() {  
        return ssn;  
    }  
    public void setSsn(int ssn) {  
        this.ssn = ssn;  
    }  
    public void setDisease(String disease) {  
        this.disease = disease;  
    }  
    public double getSalary() {  
        return salary;  
    }  
}
```

Hangisi yanlıştır?

- A) Encapsulation obj = new Encapsulation();
System.out.println(obj.getSsn());
Output 1234
- B) Encapsulation obj = new Encapsulation();
obj.setSsn(6789);
System.out.println(obj.getSsn());
Output 6789
- C) Encapsulation obj = new Encapsulation();
System.out.println(obj.getDisease());
Output Headache

4. Hangisi yanlıştır?

- A) Eğer encapsulated datalar için sadece getter methodlari oluştursanız class "immutable class" olmuş demektir.
- B) Eğer encapsulated dataların okunmasını istemiyorsanız getter methodlari oluşturmamalısınız.
- C) Eğer encapsulated dataların update edilmesini istemiyorsanız setter methodlari oluşturmamalısınız.
- D) Hiçbiri

5. `public class Student {
 public int id;
 protected String name;

 public int getId() {
 return id;
 }
 public void setId(int id) {
 this.id = id;
 }
 public String getName() {
 return name;
 }
}`

Aşağıdakilerden hangilerini yaparsak class immutable class olur?

- A) id ve name variable'lerinin access modifierlari private yapılmalı
- B) setId(int id) methodu silinmeli
- C) Tume getter methodlar silinmeli
- D) Student class'inin access modifier'i "private" yapılmalı


```
6. public class Test1 {  
    private String gender = "Male";  
    private int age = 23;  
    private boolean disabled = false;  
    public int getAge() {  
        return age;  
    }  
    public void setAge(int age) {  
        this.age = age;  
    }  
    public boolean isDisabled() {  
        return disabled;  
    }  
    public void setDisabled(boolean disabled) {  
        this.disabled = disabled;  
    }  
    public String getGender() {  
        return gender;  
    }  
}
```

Hangisi yanlıştır?

- A) `public static void main(String[] args) {
 Test1 obj1 = new Test1();
 obj1.setDisabled(true);
 System.out.println(obj1.isDisabled());
}`
Output true
- B) `public static void main(String[] args) {
 Test1 obj1 = new Test1();
 obj1.setGender("Female");
 System.out.println(obj1.getGender());
}`
Output Female
- C) `public static void main(String[] args) {
 Test1 obj1 = new Test1();
 System.out.println(obj1.setAge());
}`
Compile Time Error
- D) Hiçbiri

7.

```
public class Test1 {  
    public int id;  
    protected String name;  
    public void getId() {  
        return id;  
    }  
    public int setId(int id) {  
        this.id = id;  
    }  
    public String getName() {  
        return name;  
    }  
}
```

Fully Encapsulation için aşağıdakilerden hangileri yapılmalı?

- A) id ve name variable'lerinin access modifierleri private yapılmalı
- B) getId() methodunun return type'i int yapılmalı
- C) setId(int id) methodunun return type'i void yapılmalı
- D) name variable'inin setter methodu oluşturulmalı

```
8. public class Test1 {  
    private double width;  
    private double height;  
    private double screenSize;  
    private int maxVolume;  
    private int volume = 12;  
    private boolean power;  
    public Test1(double width, double height, double screenSize) {  
        this.width = width;  
        this.height = height;  
        this.screenSize = screenSize;  
    }  
    public double channelTuning(int channel) {  
        switch (channel) {  
            case 1:  
                return 34.56;  
            case 2:  
                return 54.89;  
            case 3:  
                return 73.89;  
            case 4:  
                return 94.98;  
        }  
        return 0;  
    }  
    public int decreaseVolume() {  
        if (0 < volume) {  
            volume--;  
        }  
        return volume;  
    }  
    public void powerSwitch() {  
        this.power = !power;  
    }  
}
```

Encapsulation

```
    }  
    public int increaseVolume() {  
        if (maxVolume > volume)  
            volume++;  
        return volume;  
    }  
}
```

Hangileri doğrudur?

A)

```
public class Test2 {  
    public static void main(String args[]){  
        Test1 t= new Test1(11.5,7,9);  
        t.powerSwitch();  
        System.out.println(t.channelTuning(2));  
    }  
}
```

Output 54.89

B)

```
public class Test2 {  
    public static void main(String args[]){  
        Test1 t= new Test1(11.5,7,9);  
        t.powerSwitch();  
        System.out.println(t.decreaseVolume());  
    }  
}
```

Output 11

C)

```
public class Test2 {  
    public static void main(String args[]){  
        Test1 t= new Test1(11.5,7,9);  
        t.powerSwitch();  
        System.out.println(t.width);  
    }  
}
```

Output 0

```
9. public class Student {  
    private int id = 12345;  
    private String name = "Ali Can";  
  
    public int getId() {  
        return id;  
    }  
    public void setId(int id) {  
        this.id = id;  
    }  
    public String getName() {  
        return name;  
    }  
}
```

Hangisi yanlıştır?

- A) id read ve update edilebilir
- B) name update edilebilir ama read edilemez
- C) Student class'ından oluşturulan tüm object'lerin name value'su Ali Candır.
- D) Student class'ından oluşturulan her object için farklı bir id oluşturabiliriz.

```
1. public class Animal {  
    public int weight;  
    public static void main(String[] args) {  
    }  
}  
  
public class Mammal extends Animal {  
    protected boolean feed = true;  
    public static void main(String[] args) {  
  
    }  
}  
  
public class Dog extends Mammal {  
    private boolean bark = true;  
    public static void main(String[] args) {  
        Dog dog = new Dog();  
        System.out.println(dog.bark);  
        System.out.println(dog.feed);  
        System.out.println(dog.weight);  
    }  
}
```

Hangisi doğrudur?

- A) true
true
0
- B) System.out.println(dog.bark);
kodu Compile Time Error verir
- C) Çalışır ama ekrana birşey yazdırmaz
- D) Run Time Error

2. `public class A { }`
`public class B extends A { }`
`public class C extends B { }`

Hangileri doğrudur?

- A) C IS-A B
- B) C IS-A A
- C) B HAS-A C
- D) A HAS-A C

3. **Hangisi yanlıştır?**

- A) `public class Animal { }`
`public class Mammal extends Animal { }`
Bu "Single Inheritance" ve Java bunu kabul eder.
- B) `public class Animal { }`
`public class Mammal extends Animal { }`
`public class Mammal extends Creator { }`
Bu "Multiple Inheritance" ve Java bunu kabul eder.
- C) `public class Animal { }`
`public class Mammal extends Animal { }`
`public class Dog extends Mammal { }`
Bu "Multi Level Inheritance" ve Java bunu kabul eder.
- D) `public class Animal { }`
`public class Dog extends Animal { }`
`public class Cat extends Animal { }`
Bu "Hierarchical Inheritance" ve Java bunu kabul eder.

4. Java'da inheritance ile alakalı aşağıdakilerden hangileri doğrudur?

- A) Java'da tüm classlar Object class'dan inherit yaparlar.
- B) Java'da Object class haric tüm class'ların parent class'i vardır.
- C) public ve protected class memberlar problemsiz bir şekilde inherit edilebilirler.
- D) default class memberlar inherit edilemezler

5.

```
public class Animal {  
    public void m1()  
    public void m2()  
}  
public class Mammal extends Animal {  
    public void m3()  
}  
public class Dog extends Mammal {  
    public void m4()  
    public void m5()  
}
```

Hangisi doğrudur?

- A) Dog obj = new Dog();
m1(), m2(), m3(), m4(), ve m5() methodlarına obj object kullanılarak ulaşılabilir.
- B) Dog obj = new Dog();
obj object kullanılarak sadece m4() ve m5() methodlarına ulaşılabilir.
- C) Dog obj = new Dog();
obj object kullanılarak sadece m3(), m4() ve m5() methodlarına ulaşılabilir.
- D) Compile Time Error verir

6. package x

```
public class Animal {  
    void m1()  
    private void m2()  
}
```

package y

```
public class Mammal extends Animal {  
    public void m3()  
    protected void m4()  
    void m5()  
}
```

package y

```
public class Dog extends Mammal {  
    public void m6()  
    private void m7()  
}
```

Hangisi doğrudur?

- A) Dog obj = new Dog();
obj object kullanılarak sadece m3(), m4(), m5(), m6() ve m7() methodlarına ulaşılabilir
- B) Dog obj = new Dog();
obj object kullanılarak m1(), m2(), m3(), m4(), m5(), m6() ve m7() methodlarına ulaşılabilir
- C) Dog obj = new Dog();
obj object kullanılarak sadece m2(), m3(), m4(), m5(), m6() ve m7() methodlarına ulaşılabilir
- D) Dog obj = new Dog();
obj object kullanılarak sadece m3(), m4(), m5(), m6() methodlarına ulaşılabilir

7. **Hangileri doğrudur?**

- A) Eğer X IS-A Y ise X classı Y classinin sub classıdır ve Y classı X classinin super classıdır.
- B) Eğer X HAS-A Y ise X super class ve Y sub classıdır
- C) Eğer X IS-A Y ve Y HAS-A Z ise X ve Z kardes olabilirler
- D) Eğer X IS-A Y ve Z HAS-A X ise bu Java'nin desteklemediği bir inheritance olabilir.

```
1. public class Animal {  
    Animal(){  
        System.out.println("Animal Constructor");  
    }  
    public static void main(String[] args) {  
    }  
}  
public class Mammal extends Animal {  
    Mammal(){  
        System.out.println("Mammal Constructor");  
    }  
    public static void main(String[] args) {  
    }  
}  
public class Dog extends Mammal {  
    Dog(){  
        System.out.println("Dog Constructor");  
    }  
    public static void main(String[] args) {  
        Dog dog = new Dog();  
    }  
}
```

Output?

- A) Animal Constructor
Mammal Constructor
Dog Constructor
- B) Dog Constructor
Mammal Constructor
Animal Constructor
- C) Dog Constructor
- D) Runs but no output

Inheritance Constructor Call

```
2. public class Animal {
    public int weight = 12;
    protected boolean feed = true;
    public static void main(String[] args) {
    }
}

public class Mammal extends Animal {
    public int weight = 13;
    public boolean bark = false;
    public static void main(String[] args) {

    }
}

public class Dog extends Mammal {
    private boolean bark = true;
    public static void main(String[] args) {
        Dog dog = new Dog();
        System.out.println(dog.bark);
        System.out.println(dog.feed);
        System.out.println(dog.weight);
    }
}
```

Hangisi doğrudur?

- A) true
true
13
- B) System.out.println(dog.bark); Compile Time Error verir.
- C) true
- D) Run Time Error

Inheritance Constructor Call

```
3. public class Animal {
    public int weight;
    public static void main(String[] args) {
    }
}

public class Mammal extends Animal {
    protected boolean feed = true;
    public static void main(String[] args) {

    }
}

public class Dog extends Mammal {
    public boolean bark = true;
    public static void main(String[] args) {
        Mammal dog = new Dog();
        System.out.println(dog.bark);
        System.out.println(dog.feed);
        System.out.println(dog.weight);
    }
}
```

Hangisi doğrudur?

- A) true
true
0
- B) System.out.println(dog.bark); Compile Time Error verir.
- C) Çalışır ama ekrana birşey yazdırmaz
- D) Run Time Error

```
4. public class Animal {  
    public int weight;  
    public static void main(String[] args) {  
    }  
}  
  
public class Mammal extends Animal {  
    protected boolean feed = true;  
    public static void main(String[] args) {  
  
    }  
}  
  
public class Dog extends Mammal {  
    public boolean bark = true;  
    public static void main(String[] args) {  
        Animal dog = new Dog();  
        System.out.println(dog.bark);  
        System.out.println(dog.feed);  
        System.out.println(dog.weight);  
    }  
}
```

Hangisi doğrudur?

- A) true
true
0
- B) System.out.println(dog.bark); ve System.out.println(dog.feed); Compile Time Error verir
- C) Çalışır ama ekrana birşey yazdırmaz
- D) Run Time Error

Inheritance Constructor Call

5. `public class A { }`
`public class B extends A { }`
`public class C extends B { }`

Hangileri doğrudur?

- A) A ve B super classdır
- B) B ve C sub classdır
- C) Bu multi level inheritance'dir
- D) Java bu inheritance'i desteklemez

6. **Hangileri aynıdır?**

- A) `public class C1 { }`
- B) `public class C1 { C1(){} }`
- C) `public class C1 { C1(){ super() } }`
- D) `public class C1 { C1(){ super(int i) } }`

7. `1: class X{`
`2: public X(int i){`
`3: System.out.println(1);`
`4: }`
`5: }`
`6: class Y extends X{`
`7: public Y(){`
`8: System.out.println(2);`
`9: }`
`10: }`

Hangisi doğrudur?

- A) Line 2 de Compile Time Error verir
- B) Line 6 da Compile Time Error verir
- C) Line 7 de Compile Time Error verir
- D) Compile Time Error yoktur

Inheritance Constructor Call

```
8. public class C {
    public int x = 11;
    public void m() { System.out.println("Good"); }
}

public class B extends C {
    public int x = 12;
    public void m() { System.out.println("Bad"); }
}

public class A extends B {
    public static void main(String[] args) {
        B b = new B();
        b.m();
        System.out.println(b.x);

        C c = new B();
        c.m();
        System.out.println(c.x);

        C d = new C();
        d.m();
        System.out.println(d.x);
    }
}
```

Output?

- A) Bad
12
Bad
11
Good
11
- B) Bad
12
Bad
12
Good
11
- C) Bad
12
Good
11
Good
11

Inheritance Constructor Call

```
9. public class School {
    String name = "School";
    public void grade() {
        System.out.println("S");
    }
}

public class Room extends School {
    String name = "Class";
    public static void main(String[] args) {
        *****

        obj1.grade();
    }
    public void grade() {
        System.out.println("C");
    }
}
```

Asagidaki kodlar *** yerine yazılırsa hangileri doğru olur?**

- A) Room obj1 = new Room();
output C
- B) School obj1 = new School();
Output S
- C) Room obj1 = new Room();
Output S
- D) School obj1 = new School();
Output C


```
10. class A{
    public A(int i){
        System.out.println("Class A int Constructor");
    }
    public A(){
        System.out.println("Class A Constructor");
    }
}
class B extends A{
    public B(){
        super(5);
        System.out.println("Class B Constructor");
    }
    public B(double d){
        System.out.println("Class B double Constructor");
    }
}
class C extends B{
    public C(){
        System.out.println("Class C Constructor");
    }
    public C(String s){
        super(1.2);
        System.out.println("Class C String Constructor");
    }
}
```

Hangileri doğrudur?

```
A) class MainClass{
    public static void main(String[] args){
        B obj = new C();
    }
}
```

Output Class A int Constructor

Class B Constructor

Class C Constructor

Inheritance Constructor Call

B)

```
class MainClass{
    public static void main(String[] args){
        A obj = new C("Java");
    }
}
```

Output Class A Constructor
 Class B double Constructor
 Class C String Constructor

C)

```
class MainClass{
    public static void main(String[] args){
        C obj = new C();
    }
}
```

Output Class A Constructor
 Class B Constructor
 Class C Constructor

```
1. public class A {  
    int i = 11;  
    public void display() {  
        System.out.println(i);  
    }  
}
```

```
class B extends A {  
    int i = 12;  
    public void display() {  
        System.out.println(i);  
    }  
}
```

Hangisi yanlıştır?

A)

```
public class C {  
    public static void main(String args[]) {  
        B obj1 = new B();  
        obj1.display();  
        System.out.println(obj1.i);  
    }  
}
```

Output 12
12

B)

```
public class C {  
    public static void main(String args[]) {  
        A obj2 = new B();  
        obj2.display();  
        System.out.println(obj2.i);  
    }  
}
```

Output 12
11

Overriding

C)

```
public class C {  
    public static void main(String args[]) {  
        A obj3 = new A();  
        obj3.display();  
        System.out.println(obj3.i);  
    }  
}
```

Output 11
11

D)

```
class C {  
    public static void main(String args[]) {  
        B obj4 = new A();  
        obj4.display();  
        System.out.println(obj4.i);  
    }  
}
```

Output 11
11

2.

```
public class A {  
    void display() {  
        System.out.println("Good");  
    }  
}  
class B extends A {  
    @Override  
    **** void display() {  
        System.out.println("Bad");  
    }  
}  
****" yerine hangisi yazılamaz?
```

- A) private
- B) nothing
- C) protected
- D) public

```
3. public class A {  
    public A display() {  
        return new A();  
    }  
}  
  
class B extends A {  
}  
  
class C extends B {  
    @Override  
    public **** display() {  
        return new C();  
    }  
}
```

******" yerine hangileri yazılabilir?**

- A) A
- B) B
- C) C
- D) None

4. Overloading ve Overriding için hangileri doğrudur?

- A) Overloading'de method signature farklı olmalıdır fakat Overriding'de method signature aynı olmalıdır.
- B) static, private, ve final method'lar override edilemezler fakat overload edilebilirler.
- C) Overriding için inheritance şarttır fakat Overloading için inheritance şart değildir.
- D) Overriding ve Overloading de method body değiştirilmek zorundadır.

5. Hangileri method overriding için yanlıştır?

A)

```
public class A {  
    private void fly(){  
        System.out.println("A flies");  
    }  
}  
  
public class B extends A {  
    @Override  
    public void fly(){  
        System.out.println("B flies");  
    }  
}
```

B)

```
public class A {  
    public final void fly(){  
        System.out.println("A flies");  
    }  
}  
  
public class B extends A {  
    @Override  
    public final void fly(){  
        System.out.println("B flies");  
    }  
}
```

C)

```
public class A {  
    public static void fly(){  
        System.out.println("A flies");  
    }  
}  
  
public class B extends A {  
    @Override  
    public static void fly(){  
        System.out.println("B flies");  
    }  
}
```

Overriding

```
D) public class A {  
    public int fly(){  
        System.out.println("A flies");  
        return 5;  
    }  
}  
  
public class B extends A {  
    @Override  
    public byte fly(){  
        System.out.println("B flies");  
        return 7;  
    }  
}
```



6. Hangileri method overriding değildir?

A)

```
public class A {  
    public int fly(int i){  
        return 2*i;  
    }  
}  
  
public class B extends A {  
    public int fly(int i, int j){  
        return i+j;  
    }  
}
```

B)

```
public class A {  
    public int fly(int i){  
        return 2*i;  
    }  
}  
  
class B extends A {  
    public int fly(short i){  
        return i;  
    }  
}
```

C)

```
public class A {  
    public int fly(){  
        return 2;  
    }  
}  
  
class B extends A {  
    public short fly(){  
        return 2;  
    }  
}
```


Overriding

```
7. public class Shape{
    public void draw(){
        System.out.println("Shape");
    }
    public void paint(){
        System.out.println("Shape painted");
    }
}
```

```
public class Circle extends Shape{
    public void draw(){
        System.out.println("Circle");
    }
}
```

```
public class Square extends Shape{
    public void draw(){
        System.out.println("Square");
    }
    public void paint(){
        System.out.println("Square painted");
    }
}
```

Hangisi yanlıştır?

```
A) public class Main{
    public static void main(String aga[]){
        Shape c = new Circle();
        c.draw();
        c.paint();
    }
}
```

Output Circle

Shape painted

Overriding

B)

```
public class Main{  
    public static void main(String aga[]){  
        Shape s = new Square();  
        s.draw();  
        s.paint();  
    }  
}
```

Output Square

Square painted

C)

```
public class Main{  
    public static void main(String aga[]){  
        Shape t = new Shape();  
        t.draw();  
        t.paint();  
    }  
}
```

Output Shape

Shape painted

D) None

8. Hangileri Compile Time Error verir?

A)

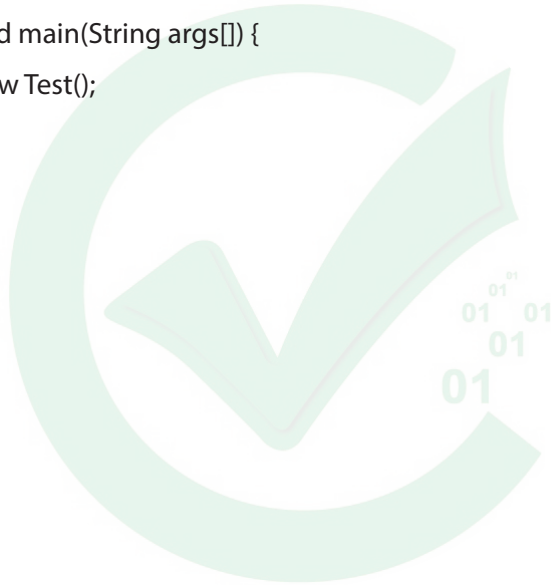
```
public class Test06 {  
    private void go() {  
        System.out.println("Good");  
    }  
}  
  
public class Test extends Test06 {  
    @Override  
    public void go() {  
        System.out.println("Bad");  
    }  
    public static void main(String args[]) {  
        Test obj = new Test();  
        obj.go();  
    }  
}
```

B)

```
public class Test06 {  
    private void go() {  
        System.out.println("Good");  
    }  
}  
  
public class Test extends Test06 {  
    public void go() {  
        System.out.println("Bad");  
    }  
    public static void main(String args[]) {  
        Test obj = new Test();  
        obj.go();  
    }  
}
```

Overriding

```
C) public class Test06 {  
    public Integer go() {  
        return 11;  
    }  
}  
  
public class Test extends Test06 {  
    @Override  
    public Short go() {  
        return 6;  
    }  
    public static void main(String args[]) {  
        Test obj = new Test();  
        obj.go();  
    }  
}
```



9. Hangileri Compile Time Error verir?

A)

```
public class Test06 {  
    public int go() {  
        return 11;  
    }  
}  
  
public class Test extends Test06 {  
    @Override  
    protected int go() {  
        return 12;  
    }  
    public static void main(String args[]) {  
        Test obj = new Test();  
        obj.go();  
    }  
}
```

B)

```
public class Test06 {  
    int go() {  
        return 11;  
    }  
}  
  
class Test extends Test06 {  
    @Override  
    protected int go() {  
        return 12;  
    }  
    public static void main(String args[]) {  
        Test obj = new Test();  
        obj.go();  
    }  
}
```

Overriding

C)

```
public class Test06 {  
    int go() {  
        return 11;  
    }  
}  
class Test extends Test06 {  
    @Override  
    byte go() {  
        return 12;  
    }  
    public static void main(String args[]) {  
        Test obj = new Test();  
        obj.go();  
    }  
}
```

D)

```
public class Test06 {  
    final int go() {  
        return 11;  
    }  
}  
class Test extends Test06 {  
    @Override  
    final int go() {  
        return 12;  
    }  
    public static void main(String args[]) {  
        Test obj = new Test();  
        obj.go();  
    }  
}
```

Overriding

```
10. public class A extends B{
    int age = 11;
    public static String sing(){
        return "fa";
    }
    public static void main(String[] args){
        A a = new A();
        B b = new B();
        System.out.println(a.sing() + " " + b.sing());
        System.out.println(a.age + " " + b.age);
    }
}
class B {
    int age = 12;
    public static String sing(){
        return "la";
    }
}
```

Output nedir?

- A) fa la
11 12
- B) la fa
11 12
- C) fa la
12 11
- D) fa fa
11 11

```
1. public class Test01 {  
    public static void main(String s[]) {  
        go();  
        go(12);  
        go(5 >= 5);  
    }  
    public static void go() {  
        System.out.println("Bad");  
    }  
    public static void go(double i) {  
        System.out.println("Good");  
    }  
    public static void go(boolean b) {  
        if (b) {  
            System.out.println("The best");  
        } else {  
            System.out.println("The worst");  
        }  
    }  
}
```

Output?

- A) Bad
Good
The best
- B) Bad
Good
The worst
- C) Compile Time Error
- D) Bad
The best


```
2. public class Test02 {  
    public static void main(String s[]) {  
        String str = print(8) + print();  
        print(str);  
    }  
    public static int print() {  
        return 3;  
    }  
    public static String print(int i) {  
        return "5";  
    }  
    public static void print(String j) {  
        System.out.println(j);  
    }  
}
```

Output?

- A) 53
- B) 83
- C) 8
- D) Compile Time Error

Overloading-Overriding

```
3. public class Test03 {  
    public static void main(String[] args) {  
        int x = add(2, 8);  
        double y = add(3, 4.5);  
        double z = add(x, y, 20);  
        System.out.println(z);  
    }  
    static int add(int i, int j){  
        return i + j;  
    }  
    static double add(double i, double j){  
        return i + j;  
    }  
    static double add(int i, double j, int k) {  
        return i + j + k;  
    }  
}
```

Output?

- A) 37.5
- B) Compile Time Error
- C) 27.5
- D) 37

```
4. public class Test04 {  
    public static void main(String[] args) {  
        System.out.println(avg(1.2, 4.8));  
        System.out.println(avg(1.2, 4.8, 6));  
    }  
    public static double avg(double i, double j) {  
        System.out.print("Good ");  
        return (i+j)/2;  
    }  
    public static double avg(double... k) {  
        System.out.print("Bad ");  
        double sum = 0;  
        for(int i=0; i<k.length; i++) {  
            sum = sum + k[i];  
        }  
        return sum/k.length;  
    }  
}
```

Output?

- A) Good 3.0
Bad 4.0
- B) Good 3
Bad 4
- C) Compile Time Error
- D) Bad 3.0
Bad 4.0

```
5. public class Test05 {  
    int x = 11;  
    public void go() {  
        System.out.println("Good");  
    }  
}  
  
public class Test extends Test05 {  
    int x = 12;  
    public void go() {  
        System.out.println("Bad");  
    }  
    public static void main(String args[]) {  
        Test05 obj = new Test();  
        obj.go();  
        System.out.println(obj.x);  
    }  
}
```

Output?

- A) Bad
11
- B) Bad
12
- C) Good
11
- D) Good
12

```
6. final class A {  
    int i = 11;  
}  
class B extends A {  
    int i = 12;  
}  
class Runner {  
    public static void main(String args[])  
    {  
        B obj = new B();  
        System.out.println(obj.i);  
    }  
}
```

Hangisi doğrudur?

- A) Output 11
- B) Output 12
- C) Compile Time Error
- D) Run Time Error

```
7. class A {  
    int i;  
    public void display() {  
        System.out.println(i);  
    }  
}  
class B extends A{  
    int j;  
    public void display() {  
        System.out.println(j);  
    }  
}
```

Hangileri doğrudur?

A) class Main {
 public static void main(String args[]){
 B obj2 = new B();
 obj2.i = 1;
 obj2.j = 2;
 A r = obj2;
 r.display();
 }
}

Output 2

B) class Main {
 public static void main(String args[]){
 B obj2 = new B();
 obj2.i = 1;
 obj2.j = 2;
 A r = obj2;
 System.out.println(r.j);
 }
}

Compile Time Error

```
C) class Main {  
    public static void main(String args[]){  
        B obj2 = new B();  
        obj2.i = 1;  
        obj2.j = 2;  
        A r = obj2;  
        System.out.println(r.i);  
    }  
}
```

Output 1

8. Hangileri doğrudur?

- A) final variable'ların değerleri değiştirilemez, sabit değer sahiptirler.
- B) final variable'lar initialize edilmelidirler.
- C) final variable'lar isimlendirilirken büyük harfler kullanılır, isimlendirmede birden fazla kelime kullanılırsa kelimeler arasına "_" konulur.
- D) final keyword program boyunca değerinin değiştirilmesini istemediğimiz variable'lar için kullanılır.

9. Which ones are true?

- A) Bir final class'a extend edemeyiz.
- B) Bir class için inheritance kullanılmaması istersek o class'i final class yaparız.
- C) Bir final method override edilemez.
- D) Tüm child class'ların parent class'daki bir methodu, body'sini değiştirmeden kullanmasını istiyorsak o methodu final method yaparız.

10. Which ones are true?

- A) "Method Overloading" Compile Time Polymorphism'dir, "Method Overriding" Run Time Polymorphism'dir.
- B) static method'lar overload edilebilir ama override edilemez.
- C) Static variable'lara ve static method'lara sadece class ismi kullanılarak da ulaşılabilir.
- D) None

1.

```
public static void main(String[] args){
    FileInputStream f = new FileInputStream("src/day/TextFile");
}
```

Hangileri doğrudur?

- A) Problemsiz çalışır
- B) Compile Time Exception verir, ")" ve "{" arasına throws FileNotFoundException yazarak Exception'i handle edebiliriz.
- C) Compile Time Exception verir, ")" ve "{" arasına throws IOException yazarak Exception'i handle edebiliriz.
- D) Run Time Exception verir.

2.

```
public static void main(String[] args) {
    FileInputStream f = new FileInputStream("src/day/TextFile");
    int i = 0;
    while ((i = f.read()) != -1) {
        System.out.print((char) i);
    }
    f.close();
}
```

Hangisi doğrudur?

- A) Problemsiz çalışır
- B) Compile Time Exception verir, ")" ve "{" arasına throws FileNotFoundException yazarak Exception'i handle edebiliriz.
- C) Compile Time Exception verir, ")" ve "{" arasına throws IOException yazarak Exception'i handle edebiliriz.
- D) Run Time Exception verir.

3. **Hangileri doğrudur?**

- A) FileNotFoundException path yanlış olduğu zaman, dosya silinmiş olduğu zaman, veya dosya var ama ulaşamıyor olduğu zaman devreye girer.
- B) IOException input ve output işlemlerinde problem olduğu zaman devreye girer.
- C) FileNotFoundException extends IOException
- D) FileNotFoundException ve IOException Run Time Exception'dirlar.

4. Hangileri doğrudur?

A)

```
public class E02 {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 2;  
        System.out.println(x / y);  
    }  
}
```

Output 2.5

B)

```
public class E02 {  
    public static void main(String[] args) {  
        int x = 0;  
        int y = 2;  
        System.out.println(x / y);  
    }  
}
```

Output 0

C)

```
public class E02 {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 0;  
        System.out.println(x / y);  
    }  
}
```

Throws ArithmeticException

D)

```
public class E02 {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 0;  
        System.out.println(x / y);  
    }  
}
```

Compile Time Exception verir ")" ve "{" arasına throws ArithmeticException yazılmalıdır.

5. Hangileri ekrana "Good" yazar?

A)

```
public class E02 {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 0;  
        System.out.println("Good");  
        System.out.println(x / y);  
    }  
}
```

B)

```
public class E02 {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 0;  
        System.out.println(x / y);  
        System.out.println("Good");  
    }  
}
```

C)

```
public class E02 {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 2;  
        System.out.println(x / y);  
        System.out.println("Good");  
    }  
}
```

D)

```
public class E02 {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 2;  
        System.out.println("Good");  
        System.out.println(x / y);  
    }  
}
```

6. Hangileri doğrudur?

A)

```
public class E {  
    public static void main(String[] args) {  
        int arr[] = {9, 8, 7, 6};  
        try {  
            System.out.print("Good");  
            System.out.print(arr[4]);  
        } catch (ArrayIndexOutOfBoundsException e) {  
            System.out.print("Better");  
        }  
    }  
}
```

Output GoodBetter

B)

```
public class E {  
    public static void main(String[] args) {  
        String s2 = null;  
        try {  
            System.out.println(s2.length());  
            System.out.print("Good");  
        } catch (NullPointerException e) {  
            System.out.print("Better");  
        }  
    }  
}
```

Output GoodBetter

C)

```
String s = "123";  
System.out.println(s + 1); // A  
int num = Integer.parseInt(s);  
System.out.println(num + 1); // B  
A = 1231 ve B = 124
```

Exceptions - 1

```
D) public class E {  
    public static void main(String[] args) {  
        String s2 = "abc";  
        try{  
            System.out.println(Integer.parseInt(s2));  
        }catch(NullPointerException e) {  
            System.out.print("Better");  
        }catch(NumberFormatException e) {  
            System.out.print("The Best");  
        }  
    }  
}
```

Output The Best

```
7. public static void main(String[] args) {  
    try {  
        FileInputStream f = new FileInputStream("src/daypackage/TextFile");  
  
    } catch(FileNotFoundException e) {  
        System.out.println("Good");  
    } catch(IOException e) {  
        System.out.println("Bad");  
    } finally {  
        System.out.println("Not bad");  
    }  
}
```

Hangisi yanlıştır?

- A) Eğer FileNotFoundException ve IOException yer degistirirse Compile Time Error verir.
- B) Eğer IOException silinirse Compile Time Error verir
- C) Eğer finally silinirse ekranda ya hiçbir şey görünmez veya output Good olur
- D) Output "Bad" olamaz.

```
8. public static void main(String[] args) {  
    try {  
        FileInputStream f = new FileInputStream("src/daypackage/TextFile");  
        int i = 0;  
        while ((i = f.read()) != -1) {  
            System.out.print((char) i);  
        }  
        f.close();  
    } catch(FileNotFoundException e) {  
        System.out.println("Most probably path is wrong or someone deleted my file");  
  
    } catch(IOException e) {  
        System.out.println("Text could not be read or the file could not be closed");  
  
    } finally {  
        System.out.println("Do not forget to close the file!");  
    }  
}
```

Hangisi doğrudur?

- A) File silinmiş ise output
Most probably path is wrong or someone deleted my file
Do not forget to close the file!
- B) Path doğru ve file silinmemiş ise output just the text inside the file
- C) Compile Time Error verir
- D) Run Time Error verir

Exceptions - 1

9. **Bilgi:** `ArrayIndexOutOfBoundsException` class'i `Exception` class'in child class'idir.

Yukarıdaki bilgiye göre hangisi yanlıştır?

A)

```
public static void main(String[] args){
    try {
        int a[] = {7, 8, 9};
        for (int i = 1; i <= 3; i++){
            System.out.println(a[i]);
        }
    } catch (Exception e) {
        System.out.println ("Exception");
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println ("Out of index");
    } finally {
        System.out.println("Done!");
    }
}
```

Compile time error verir

B)

```
public static void main(String[] args){
    try {
        int a[] = {7, 8, 9};
        for (int i = 1; i <= 2; i++){
            System.out.println(a[i]);
        }
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println ("Out of index");
    } finally {
        System.out.println("Done!");
    }
}
```

Output

8

9

Done!

C)

```
public static void main(String[] args){  
    try {  
        int a[]={7, 8, 9};  
        for (int i = 1; i <= 7; i++){  
            System.out.println(a[i]);  
        }  
    } catch(Exception e) {  
        System.out.println ("Exception");  
    }finally {  
        System.out.println("Done!");  
    }  
}
```

Output

8

9

Exception

Done!

D) Hiçbiri

Exceptions - 1

```
10. public class Test2 {
    String str = "a";
    void A() {
        try {
            str = str + "b";
            B();
        } catch (Exception e) {
            str = str + "c";
        }
    }
    void B() throws Exception {
        try {
            str = str + "d";
            C();
        } catch (Exception e) {
            str = str + "e";
        } finally {
            str = str + "f";
        }
        str = str + "g";
    }
    void C() throws Exception {
        System.out.println(12/0);
    }
    void display() {
        System.out.println(str);
    }
    public static void main(String[] args) {
        Test2 object = new Test2();
        object.A();
        object.display();
    }
}
```

Output nedir?

- A) abdefg
- B) abcdefg
- C) abd
- D) abcefg

1. Hangisi yanlıştır?

- A) throw method body içine yazılır, throws method parantezi ile curly brace arasına yazılır.
- B) throw method body içinde birçok kez kullanılabilir. throws bir method için sadece bir kez kullanılabilir.
- C) 1 throw sadece 1 exception atar fakat 1 throws 1'den fazla exception atabilir.
- D) throws new NullPointerException();
kodu NullPointerException atar.

2. Eğer console'da sadece exception mesajını görmek isterseniz aşağıdakilerden hangisini yatmalısınız?

- A)

```
catch (Exception e) {  
    System.out.println(e);  
}
```
- B)

```
catch (Exception e) {  
    System.out.println(e.getMessage());  
}
```
- C)

```
catch (Exception e) {  
    e.printStackTrace();  
}
```
- D)

```
catch (Exception e) {  
    e.getMessage();  
}
```

Exceptions - 2

3. Exception hakkında detayli aciklama gormek isterseniz aşağıdakilerden hangisini yazmalisiniz?

- A)

```
catch (Exception e) {  
    e.printStackTrace();  
}
```
- B)

```
catch (Exception e) {  
    System.out.println(e.printStackTrace());  
}
```
- C)

```
catch (Exception e) {  
    System.out.println(e.getMessage());  
}
```
- D)

```
catch (Exception e) {  
    e.getMessage();  
}
```

4.

```
public static void main(String[] args) {  
    try {  
        run();  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
}  
  
private static void run() {  
    throw new RuntimeException("Run more");  
}
```

Output nedir?

- A) java.lang.RuntimeException: Run more
- B) Run more
- C) java.lang.RuntimeException: Run more
at day30exceptions.TTryCatch06.run(TTryCatch06.java:16)
at day30exceptions.TTryCatch06.main(TTryCatch06.java:7)

5. Hangisi yanlıştır?

A)

```
public static void main(String[] args) {  
    Integer z = 0;  
    try {  
        String s = "1234";  
        z = Integer.parseInt(s);  
    } catch (NumberFormatException e) {  
        System.out.println(e);  
    }  
    System.out.println(z);  
}
```

Output 1234

B)

```
public static void main(String[] args) {  
    Integer z = 0;  
    try {  
        String s = "abcd";  
        z = Integer.parseInt(s);  
    } catch (NumberFormatException e) {  
        System.out.println("Good");  
    }  
    System.out.println(z);  
}
```

Output

Good

0

Exceptions - 2

C)

```
public static void main(String[] args) {  
    Integer z = 0;  
    try {  
        String s = "abcd";  
        z = Integer.parseInt(s);  
    } catch (NumberFormatException e) {  
        System.out.println("Good");  
    }  
    System.out.println(z);  
}
```


Run Time Exception verir

D)

```
public static void main(String[] args) {  
    String s = "abcd";  
    try {  
        Object obj = new Integer(12);  
        s = (String)obj;  
    } catch (ClassCastException e){  
        System.out.println("Good");  
    }  
    System.out.println(s);  
}
```


Output abcd

6. Hangisi yanlıştır?

A)

```
public static void main(String[] args) {  
    try {  
        m(-12);  
    } catch (IllegalArgumentException e) {  
        System.out.println(e.getMessage());  
    }  
}  
  
public static void m(int x) {  
    if (x < 0 || x > 9) {  
        throw new IllegalArgumentException("Good");  
    } else {  
        System.out.println(x);  
    }  
}
```


Output Good

B)

```
public static void main(String[] args) {  
    try {  
        m(12);  
    } catch (IllegalArgumentException e) {  
        System.out.println(e.getMessage());  
    }  
}  
  
public static void m(int x) {  
    if (x < 0 || x > 9) {  
        throw new IllegalArgumentException("Good");  
    } else {  
        System.out.println(x);  
    }  
}
```


Output Good

Exceptions - 2

```
C) public static void main(String[] args) {  
    try {  
        m(5);  
    } catch (IllegalArgumentException e) {  
        System.out.println(e.getMessage());  
    }  
}  
  
public static void m(int x) {  
    if(x<0 || x>9) {  
        throw new IllegalArgumentException("Good");  
    } else {  
        System.out.println(x);  
    }  
}
```

Output

Good

5

```
D) public static void main(String[] args) {  
    try {  
        m(9);  
    } catch (IllegalArgumentException e) {  
        System.out.println(e.getMessage());  
    }  
}  
  
public static void m(int x) {  
    if(x<0 || x>9) {  
        throw new IllegalArgumentException("Good");  
    } else {  
        System.out.println(x);  
    }  
}
```

Output 9

Exceptions - 2

```
7. public class Test {  
    public static void main(String[] args) {  
        try {  
            int a = 0;  
            System.out.println("Hi");  
            int b = 20 / a;  
            System.out.println("How are you?");  
        } catch (ArithmeticException e) {  
            System.out.println("Good");  
        } finally {  
            System.out.println("Bad");  
        }  
    }  
}
```

Output nedir?

- A) Hi
Good
Bad
- B) Hi
How are you?
Bad
- C) Hi
Good
- D) Hi
Bad

Exceptions - 2

```
8. public class Test2 {  
    public static void main(String[] args){  
        try {  
            int a[]={7, 8, 9};  
            for (int i = 1; i <= 3; i++){  
                System.out.println(a[i]);  
            }  
        } catch(Exception e) {  
            System.out.println("Out of index");  
        } finally {  
            System.out.println("Done!");  
        }  
    }  
}
```

Output nedir?

- A) 8
9
Out of index
Done!
- B) Out of index
Done!
- C) 8
9
Out of index
- D) Out of index


```
9. try {  
    System.out.println(num1/num2);  
} catch(ArithmeticException e) {  
    System.out.println("Do not divide by 0");  
} finally {  
    System.out.println("Done");  
}
```

Hangisi doğrudur?

- A) If int num1 = 22; and int num2 = 0; the output is
Do not divide by 0
- B) If int num1 = 22; and int num2 = 2; the output is
11
Do not divide by 0
Done
- C) If int num1 = 0; and int num2 = 22; the output is
0
Do not divide by 0
Done
- D) If int num1 = 2; and int num2 = 22; the output is
0
Done

10. Hangileri doğrudur?

- A) finally{} try-catch block'dan sonra kullanılır.
finally block exception atılsa da atılması da çalışır.
- B) final variable'ların değerleri değiştirilemez ve final variable'lar mutlaka initialize edilmelidirler.
- C) final method'lar override edilemezler.
- D) final class'lara extend edilemez.
- E) finalize() method, Garbage Collector dataları imha etmeden önce çalıştırılır.

Exceptions - 2

```
11. public class Test01 {  
    public static void main(String[] args) {  
        String s1 = "";  
        String s2 = null;  
        try {  
            s1 = s1 + "J";  
            System.out.println(s2.length());  
            s1 = s1 + "a";  
        } catch (ArithmeticException e) {  
            s1 = s1 + "v";  
        } catch (NullPointerException e) {  
            s1 = s1 + "u";  
        } finally {  
            s1 = s1 + "n";  
        }  
        s1 = s1 + "e";  
        System.out.println(s1);  
    }  
}
```

Output nedir?

- A) June
- B) Jave
- C) Jaune
- D) 0Jane

```
1. public abstract class Ab01 {  
    public int a = 10;  
    public abstract void set(int a);  
    public abstract void get();  
}  
  
public class Test extends Ab01{  
    public void set(int a){  
        this.a = a;  
    }  
    public static void main(String[] args){  
        Test obj = new Test();  
        obj.set(20);  
        System.out.println(obj.a);  
    }  
}
```

Aşağıdakilerden hangisi doğrudur?

- A) The output is 20
- B) The output is 10
- C) Gives compile time error
- D) Gives run time error

Abstract Classes

```
2. public abstract class Ab01 {  
    public int a = 10;  
    public abstract void set(int a);  
    public int get() {  
        return a;  
    }  
}  
  
public class Test extends Ab01{  
    public void set(int a){  
        this.a = a;  
    }  
    public static void main(String[] args){  
        Test obj = new Test();  
        obj.set(20);  
        System.out.println(obj.get());  
    }  
}
```

Aşağıdakilerden hangisi doğrudur?

- A) Output 20
- B) Output 10
- C) Compile Time Error verir
- D) Run Time Error verir

Abstract Classes

3. Aşağıdakilerden hangileri Compile Time Error verir?

- A)

```
public class Ab01 {  
    public int a = 10;  
    public abstract void set(int a);  
    public int get() {  
        return a;  
    }  
}
```
- B)

```
public abstract class Ab01 {  
    public int a = 10;  
    public int get() {  
        return a;  
    }  
}
```
- C)

```
public abstract class Ab01 {  
    public int a = 10;  
    public abstract void set(int a);  
    public int get();  
}  
  
public class abstract Test extends Ab01{  
    public void set(int a){  
        this.a = a;  
    }  
    public static void main(String[] args){  
    }  
}
```
- D)

```
public abstract class Ab01 {  
    public int a = 10;  
    public abstract void set(int a);  
    public int get() {  
        return a;  
    }  
    public static void main(String[] args) {  
        Ab01 obj = new Ab01();  
    }  
}
```

4. Aşağıdakilerden hangileri Compile Time Error verir?

A)

```
public abstract class Ab01 {  
    public int a = 10;  
    public abstract void set(int a);  
    public abstract int get() {  
        return a;  
    }  
}
```

B)

```
public abstract class Ab01 {  
    public int a = 10;  
    public abstract void set(int a);  
    public int get();  
}
```

C)

```
public class Ab01 {  
    public int a = 10;  
    public abstract void set(int a);  
    public int get() {  
        return a;  
    }  
}
```

D)

```
public abstract class Ab01 {  
    public int a = 10;  
    public void set(int a) {  
        this.a = a;  
    }  
    public int get() {  
        return a;  
    }  
}
```

5. Aşağıdakilerden hangisi abstraction için yanlıştır?

- A) Bir abstract class'a extend ettiğimizde, parent class'daki tüm abstract method'lari implement etmezsek, Compile Time Error almamak için child class'i abstract yapabiliriz.
- B) Abstract class'in içinde hiç abstract method üretmesek de olur.
- C) Bir concrete class abstract method içerebilir, ama bir abstract class concrete method içerebilir.
- D) Abstract class bir class olduğundan, abstract class'lardan object üretilebilir.

6. public abstract class Ab01 {

int b = 12;

abstract void getNumber(int x, int y);

}

public class Test extends Ab01 {

int b = 10;

Test() {

b = super.b;

}

void getNumber(int x, int y) {

int i = x * y + b;

System.out.print(i);

}

public static void main(String[] args) {

Test sub = new Test();

sub.getNumber(7, 3);

}

}

Output nedir?

- A) 33
- B) 31
- C) Compile Time Error verir
- D) Run Time Error verir

Abstract Classes

7. `public abstract class Y { public int method(){ return 5; } }`
`public class Z { public abstract void method(); }`
`public abstract class T { }`
`public class U { }`

Aşağıdakilerden hangisi veya hangileri Compile Time Error verir?

- A) Z
- B) Y
- C) Y, Z
- D) U, T, Z

8. 1) Bir okulda 6'inci, 7'inci ve 8'inci sınıflar var.
2) Matematik ve İngilizce dersleri tüm sınıflar için farklı içeriklerle mecburi derstir.
3) Resim tüm sınıflar için aynı içerikle seçmeli bir derstir.

Yukarıdaki senaryo için aşağıdakilerden hangisi en uygundur?

- A) Okul class'i abstract parent class olmalıdır.
Matematik ve İngilizce dersleri abstract method'lar olmalıdır.
Resim dersi concrete method olmalıdır.
- B) Okul class'i abstract parent class olmalıdır.
Matematik, İngilizce ve Resim dersleri abstract method'lar olmalıdır.
- C) Okul class'i concrete parent class olmalıdır.
Matematik, İngilizce ve Resim dersleri concrete method'lar olmalıdır.
- D) Okul class'i concrete parent class olmalıdır.
Matematik ve İngilizce dersleri abstract method'lar olmalıdır.
Resim dersi concrete method olmalıdır.

Abstract Classes

```
9. abstract class A{
    abstract void firstMethod();
    void secondMethod(){
        System.out.println("SECOND");
        firstMethod();
    }
}

abstract class B extends A{
    @Override
    void firstMethod(){
        System.out.println("FIRST");
    }
    abstract void thirdMethod();
}

class C extends B{
    @Override
    void thirdMethod(){
        System.out.println("THIRD");
    }
}

public class MainClass{
    public static void main(String[] args){
        C c = new C();
        c.firstMethod();
        c.secondMethod();
        c.thirdMethod();
    }
}
```

Output nedir?

- A) FIRST
SECOND
FIRST
THIRD
- B) FIRST
SECOND
THIRD

- C) SECOND
FIRST
FIRST
THIRD
- D) SECOND
FIRST
THIRD

10. `public abstract class A {
 abstract int getSum();
 abstract int getSum(int x);
 abstract int getSum(int x, int y);
}`

```
class Sub extends A {  
    int x = 4;  
    int getSum(){  
        return x + x;  
    }  
    int getSum(int x, int y){  
        int z = x + y;  
        System.out.print(z);  
        return z;  
    }  
    public static void main(String[] args){  
        A sub = new Sub();  
        sub.getSum(4,2);  
    }  
}
```

Aşağıdakilerden hangileri doğrudur?

- A) Compile Time Error yoktur ve output 6 olur
- B) Compile Time Error vardır, sub class'i abstract class yaparak Compile Time Error'dan kurtulabiliriz.
- C) Compile Time Error vardır, getSum method'lardan Compile Time Error'a sebep olanini override ederek Compile Time Error'dan kurtulabiliriz.
- D) Compile Time Error vardır, getSum method'lardan Compile Time Error'a sebep olanini silerek Compile Time Error'dan kurtulabiliriz.

```
1. public interface I01{
    int myMethod();
}

public class Test01 implements I01 {
    public static void main(String[] args) {
        Test01 obj = new Test01();
        System.out.println(obj.myMethod());
    }
    @Override
    public int myMethod() {
        return 12;
    }
}
```

Hangisi doğrudur?

- A) "implements" keyword yerine "extends" keyword kullanılmalıydı
- B) Output 12 dir.
- C) int myMethod(); yerine
public abstract int myMethod(); yapılmalıydı.
- D) myMethod() methodunu çağırmak için object oluşturmaya gerek yok, sadece method ismini kullanmak yeterlidir.

2. Hangileri Compile Time Error verir?

A)

```
public interface I02{  
    int x = 3;  
    x++;  
}
```

B)

```
public interface I02{  
    int myMethod() {  
        return 12;  
    }  
}
```

C)

```
public interface I02{  
    int x;  
}
```

D)

```
public interface I02{  
    public abstract int myMethod();  
}
```



```
3. public interface I02{
    int x = 2;
    char c = 'A';
    void myMethod();
    void myMethod(char c, int x);
}

public class Test01 implements I02 {
    int x = 3;
    char c = 'B';
    public void myMethod() {
        System.out.println("No parameter: " + I02.c + x);
    }
    public void myMethod(char c, int x) {
        System.out.println("One parameter: " + c + I02.x);
    }
    public static void main(String[] args) {
        Test01 obj = new Test01();
        obj.myMethod();
        obj.myMethod('Z', 7);
    }
}
```

Output nedir?

- A) No parameter: A3
One parameter: Z2
- B) No parameter: B3
One parameter: Z7
- C) No parameter: A2
One parameter: Z2
- D) No parameter: A3
One parameter: B2

Interface - 1

4. `public class A { }`
`public abstract class B { }`
`public interface C { }`
`public interface D { }`

Hangisi compile time error verir?

- A) `public class A implements C { }`
- B) `public abstract class B implements D { }`
- C) `public interface C extends D { }`
- D) `public interface D extends B { }`

5. **Hangisi yanlıştır?**

- A) `public interface I02 {`
`int x = 12;`
`}`
x'in değeri değiştirilemez.
- B) `public interface I02 {`
`protected int x = 12;`
`}`
Compile time error verir
- C) `public interface I02 {`
`int x = 12;`
`}`
x'in access modifier'i default'tur.
- D) `public interface I02 {`
`int x = 12;`
`}`
x static variable'dir.

```
6. public interface I02{
    int x = 5;
    void cal(int x);
}

public interface I03 {
    int x = 6;
    void cal(int x);
}

public class Test03 implements I02, I03 {
    int x;
    public void cal(int item){
        x = item * item;
    }
    public static void main(String args[]){
        Test03 obj = new Test03();
        obj.cal(I03.x);
        System.out.println(obj.x);
        obj.cal(I02.x);
        System.out.println(obj.x);
    }
}
```

Output nedir?

- A) 36
25
- B) 36
36
- C) 25
25
- D) 25
36

Interface - 1

```
7. public interface I02{
    int x = 5;
    void cal(int i);
}

public class Test03 implements I02 {
    int x;
    public void cal(int item){
        x = item * item;
    }
}

public class Test04 implements I02 {
    int x;
    public void cal(int item){
        x = item + item;
    }

    public static void main(String args[]){
        Test03 obj1 = new Test03();
        Test04 obj2 = new Test04();
        obj1.cal(I02.x);
        System.out.println(obj1.x);
        obj2.cal(I02.x);
        System.out.println(obj2.x);
    }
}
```

Output nedir?

- A) 25
10
- B) 25
25
- C) 10
10
- D) 0
0


```
8. public interface I01 {  
    int v1 = 2000;  
    String str1 = "Honda";  
}  
  
public interface I02 {  
    int v1 = 3000;  
    String str2 = "Audi";  
}
```

Hangileri doğrudur?

A) public class C01 implements I01, I02{
 System.out.println(v1);
}
Compile time error verir.

B) public class C01 implements I01, I02{
 System.out.println(I01.v1);
}
Output 2000 dir.

C) public class C01 implements I01, I02{
 System.out.println(str2);
}
Compile time error verir.

D) public class C01 implements I01, I02{
 System.out.println(I02.str2);
}
Output Audi'dir.

```
1. public interface Animal {  
    int x = 5;  
    abstract void eat();  
}  
  
public interface Mammal {  
    int x = 7;  
    void eat();  
}
```

Hangisi doğrudur?

A) `public class Cat implements Animal, Mammal{`
 `public static void main(String[] args) {`
 `Cat cat = new Cat();`
 `System.out.println(Animal.x);`
 `cat.eat();`
 `}`
 `@Override`
 `public void eat() {`
 `System.out.println("Cats like cheese");`
 `}`
}

Output

5

Cats like cheese

B) `public class Cat implements Animal, Mammal{`
 `public static void main(String[] args) {`
 `Cat cat = new Cat();`
 `System.out.println(Animal.x);`
 `cat.eat();`
 `}`
 `@Override`
 `public void eat() {`
 `System.out.println("Cats like cheese");`
 `}`
}

Compile Time Error verir çünkü 2 tane eat() method'u var.

Java hangisini seçeceğine karar veremez.

```
C) public class Cat implements Animal, Mammal{
    public static void main(String[] args) {
        Cat cat = new Cat();
        System.out.println(Animal.x);
        cat.eat();
    }
    @Override
    public void eat() {
        System.out.println("Cats like cheese");
    }
}
```

Compile Time Error cunku 2 tane x variable'i var.
Java hangisini sececegine karar veremez.

2. Hangileri Compile Time Error verir?

- A)

```
public interface I02 {
    default void add() {
        System.out.println("Makes addition");
    }
}
```
- B)

```
public interface I02 {
    static void add() {
        System.out.println("Makes addition");
    }
}
```
- C)

```
public interface I02 {
    void add() {
        System.out.println("Makes addition");
    }
}
```
- D)

```
public interface I02 {
    default void add();
}
```

Interface - 2

```
3. interface Animal {  
    int height = 10;  
}  
interface Mammal {  
    int height = 12 ;  
    int weight = 300;  
}
```

Hangisi yanlıştır?

A) `public class Cat implements Animal, Mammal {
 public static void main(String args[]){
 System.out.println(Animal.height);
 }
}`
Output 10

B) `public class Cat implements Animal, Mammal {
 public static void main(String args[]){
 System.out.println(Mammal.height);
 }
}`
Output 12

C) `public class Cat implements Animal, Mammal {
 public static void main(String args[]){
 System.out.println(weight);
 }
}`
Compile Time Error verir

D) `public class Cat implements Animal, Mammal {
 public static void main(String args[]){
 System.out.println(Mammal.weight);
 }
}`
Output 300

```
4. public interface A {  
    void myMethod();  
}  
  
public class B {  
    public void myMethod() {  
        System.out.println("Good");  
    }  
}  
  
public class C extends B implements A {  
}  
  
public class Test01 {  
    public static void main(String[] args) {  
        A a = new C();  
        a.myMethod();  
    }  
}
```

Hangisi doğrudur?

- A) Compile Time Error verir
- B) Output Good
- C) Run Time Error verir
- D) Console'da hiçbirşey gözükmez

Interface - 2

5.

```
public interface I02{
    void myMethod();
}

public class Test02 implements I02{
    void myMethod(){
        System.out.println("Good");
    }
}
```

Yukarıdaki code Compile Time Error verir cunku ...

- A) Interface'deki methodların access modifier'ları public 'dir. override edildiklerinde access modifier'ları daraltılmaz.
- B) Interface'deki variable'lar public, static ve final'dirlar. Bu yuzden bir variable initialize edildikten sonra değeri değıştirilemez.
- C) Bir interface ancak bir başka interface'e extend edebilir. Bir interface'in parent'i class olamaz.
- D) Interface'lerin constructor'i olamaz.

6.

```
public interface I02{
    int i = 12;
}

public class Test02 implements I02{
    void myMethod(){
        i = 13;
    }
}
```

Yukarıdaki code Compile Time Error verir cunku ...

- A) Interface'deki variable'lar public, static ve final'dirlar. Bu yuzden bir variable initialize edildikten sonra değeri değıştirilemez.
- B) Interface'deki methodların access modifier'ları public 'dir. override edildiklerinde access modifier'ları daraltılmaz.
- C) Bir interface ancak bir başka interface'e extend edebilir. Bir interface'in parent'i class olamaz.
- D) Interface'lerin constructor'i olamaz.

Interface - 2

```
7. interface K {  
    String a = "Awesome";  
    String mA();  
}  
interface L extends K {  
    String b = "Bad";  
    String mB();  
}  
class M implements K, L {  
    public String mA() {  
        return a + " " + b;  
    }  
    public String mB() {  
        return b + " " + a;  
    }  
}  
public class Test02 {  
    public static void main(String[] args) {  
        M m = new M();  
        System.out.println(m.mA());  
        System.out.println(m.mB());  
    }  
}
```

Output nedir?

- A) Awesome Bad
Bad Awesome
- B) Awesome Bad
Awesome Bad
- C) Bad Awesome
Bad Awesome
- D) Compile Time Error

Interface - 2

```
8. public class D implements E {  
    public int methodB(int i) {  
        return i = i * i;  
    }  
}  
interface E {  
    int methodB(int i);  
}  
public class Test03 {  
    public static void main(String[] args) {  
        E b = new D();  
        System.out.println(b.methodB(5));  
    }  
}
```

Output nedir?

- A) 25
- B) 5
- C) Compile Time Error
- D) Run Time Error


```
9. interface InfB {  
    int x = 3;  
    int myMethod();  
}  
interface InfA extends InfB{  
    int x = 7;  
    int y = 9;  
    int myMethod();  
}  
class MyClass implements InfA{  
    int x = 5;  
    public int myMethod(){  
        return x;  
    }  
}
```

Hangisi yanlıştır?

A)

```
public static void main(String[] args){  
    InfA obj = new MyClass();  
    System.out.print(obj.myMethod());  
}
```

Output 5

B)

```
public static void main(String[] args){  
    InfA obj = new MyClass();  
    System.out.print(InfA.x);  
}
```

Output 7

C)

```
public static void main(String[] args){  
    MyClass mc = new MyClass();  
    System.out.print(mc.y);  
    System.out.print(mc.myMethod());  
}
```

Output 95

D)

```
public static void main(String[] args){  
    System.out.print(y);  
    System.out.print(x);  
}
```

Output 97

10. Hangileri Compile Time Error verir?

- A)

```
interface B {  
    protected int x = 3;  
}
```
- B)

```
interface B {  
    public static final int x = 3;  
}
```
- C)

```
interface B {  
    protected int m();  
}
```
- D)

```
interface B {  
    default int m() {  
        return 1;  
    }  
}
```
- E)

```
interface B {  
    private static int m() {  
        return 1;  
    }  
}
```



1. Hangileri doğrudur?

A)

```
public class Iterator02 {  
    public static void main(String[] args) {  
        List<String> list = new ArrayList<>();  
        list.add("X");  
        list.add("Y");  
        list.add("Z");  
        ListIterator<String> it = list.listIterator();  
        while(it.hasNext()) {  
            Object el = it.next();  
            System.out.print(el + " ");  
        }  
    }  
}
```


Output X Y Z

B)

```
public class Iterator02 {  
    public static void main(String[] args) {  
        List<String> list = new ArrayList<>();  
        list.add("X");  
        list.add("Y");  
        list.add("Z");  
        ListIterator<String> it = list.listIterator();  
        while(it.hasNext()) {  
            Object el = it.next();  
            it.remove();  
        }  
        System.out.println(list);  
    }  
}
```


Output []

Iterators

C)

```
public class Iterator02 {  
    public static void main(String[] args) {  
        List<String> list = new ArrayList<>();  
        list.add("X");  
        list.add("Y");  
        list.add("Z");  
        ListIterator<String> it = list.listIterator();  
        while(it.hasNext()) {  
            Object el = it.next();  
            it.set(el + "!");  
        }  
        System.out.println(list);  
    }  
}
```

Output [X!, Y!, Z!]

D)

```
public class Iterator02 {  
    public static void main(String[] args) {  
        List<String> list = new ArrayList<>();  
        list.add("X");  
        list.add("Y");  
        list.add("Z");  
        for(String w : list) {  
            w = w + "!";  
        }  
        System.out.println(list);  
    }  
}
```

Output [X!, Y!, Z!]

Iterators

2. **Not:** ListIterator interface'inin previousIndex() method 'u verilen elemanın index'ini return eder. Bu method pointer list'in başına yerleştirildiğinde -1 return eder.

Yukarıda verilen bilgiye göre aşağıdakilerden hangileri doğrudur?

A) class Test{
 public static void main(String args[]){
 List<String> list = new ArrayList<>();
 list.add("AB");
 list.add("CD");
 list.add("EF");
 ListIterator<String> it = list.listIterator();
 if(it.previousIndex() == -1) {
 while(it.hasNext()) {
 System.out.print(it.next() + " ");
 }
 }
 else {
 System.out.print("Good Morning!");
 }
 }
}

Output AB CD EF

B) class Test{
 public static void main(String args[]){
 List<String> list = new ArrayList<>();
 list.add("AB");
 list.add("CD");
 list.add("EF");
 ListIterator<String> it = list.listIterator();
 if(it.previousIndex() != -1) {
 while(it.hasNext()) {
 System.out.print(it.next() + " ");
 }
 }
 else {
 System.out.print("Good Morning!");
 }
 }
}

Output Good Morning!

- C) Compile Time Error
D) Run Time Error

Iterators

3. **Not:** ListIterator interface'inin nextIndex() method 'u verilen elemanın index'ini return eder. Bu method pointer list'in sonuna yerleştirildiğinden -1 return eder.

Yukarıda verilen bilgiye göre aşağıdakilerden hangileri doğrudur?

A) class Test{
 public static void main(String args[]){
 List<String> list = new ArrayList<>();
 list.add("AB");
 list.add("CD");
 list.add("EF");
 ListIterator<String> it = list.listIterator();
 if(it.nextIndex()==-1) {
 while(it.hasNext()) {
 System.out.print(it.next() + " ");
 }
 }else {
 System.out.print("Good Morning!");
 }
 }
}

Output Good Morning!

B) class Test{
 public static void main(String args[]){
 List<String> list = new ArrayList<>();
 list.add("AB");
 list.add("CD");
 list.add("EF");
 ListIterator<String> it = list.listIterator();
 if(it.nextIndex()!=-1) {
 while(it.hasNext()) {
 System.out.print(it.next() + " ");
 }
 }else {
 System.out.print("Good Morning!");
 }
 }
}

Output AB CD EF

- C) Compile Time Error
D) Run Time Error

```
4. class Test{
    public static void main(String args[]){
        List<String> list = new ArrayList<>();
        list.add("AB");
        list.add("CD");
        list.add("EF");
        ListIterator<String> it = list.listIterator();
        Collections.reverse(list);
        while(it.hasNext()) {
            System.out.print(it.next() + " ");
        }
    }
}
```

Output nedir?

- A) EF CD AB
- B) AB CD EF
- C) Compile Time Error
- D) AB CD EF EF CD AB

5. iterator ile ilgili aşağıdakilerden hangileri doğrudur?

- A) Iterator kullanarak, collection'daki elemanları okuyabilir ve silebiliriz.
- B) Iterator collection içinde bastan sona doğru ilerleyebilmemize imkan sağlar.
- C) Iterator kullandığımızda index'lerle çalışamayız.
- D) Iterator kullanarak bir collection'a eleman ekleyemeyiz.

6. ListIterator ile ilgili aşağıdakilerden hangileri doğrudur?

- A) ListIterator collection içinde bastan sona ve sondan basa doğru ilerleyebilmemize imkan sağlar.
- B) ListIterator sadece List'ler için kullanılabilir.
- C) ListIterator'da set() method'unu kullanarak elemanları değiştirebiliriz.
- D) ListIterator'da aşağıdaki method'ların hepsini kullanmak mümkündür; next(), previous(), hasNext(), hasPrevious(), add(), set(), nextIndex(), previousIndex().

7. List<Integer> list = new ArrayList<>();

```
list.add(22);  
list.add(23);  
list.add(24);  
ListIterator<Integer> it = list.listIterator();  
while(it.hasNext()) {  
    it.next();  
}  
while(it.hasPrevious()) {  
    Integer el = it.previous();  
    it.set(el = el + 3);  
    System.out.print(el + " ");  
}
```

Output nedir?

- A) 25 26 27
- B) 27 26 25
- C) 22 23 24
- D) 24 23 22


```
8. List<Integer> list = new ArrayList<>();  
    list.add(12);  
    list.add(13);  
    list.add(14);  
    list.add(15);  
    list.add(16);  
    System.out.println(list);  
    ListIterator<Integer> it = list.listIterator();  
    int idx = 0;  
    while(it.hasNext()) {  
        Integer el = it.next();  
        if(idx>2) {  
            break;  
        }  
        it.set(el = el*2);  
        System.out.print(el + " ");  
        idx++;  
    }
```

Output nedir?

- A) 24 26 28
- B) 24 26 28 30 32
- C) 30 32
- D) Nothing will be printed on the console

1. Verilen sekle göre aşağıdakilerden hangileri doğrudur?

- A) LinkedList'in 2 interface parent'i var.
- B) PriorityQueue class implements Queue interface
- C) Set interface extends Collection interface.
- D) TreeSet class, Set interface'deki tüm abstract methodlari override etmelidir.

2. Verilen şekle göre aşağıdakilerden hangisi doğrudur?

- A) Queue interface Collections interface'deki tüm abstract methodlari override etmelidir.
- B) `Queue obj1 = new Queue();`
Yukarıdaki code bir object oluşturmaktadır.
- C) `SortedSet obj2 = new TreeSet();`
Yukarıdaki code bir object oluşturmaktadır.
- D) `ArrayList<String> obj3 = new LinkedList<>();`
Yukarıdaki code bir object oluşturmaktadır.

3. Aşağıdakilerden hangileri doğrudur?

- A) `List<String> list1 = new LinkedList<>();`
- B) `Queue<String> list2= new LinkedList<>();`
- C) `Collection<String> list3= new LinkedList<>();`
- D) `LinkedList<String> list4= new LinkedList<>();`

4. Aşağıdakilerden hangileri doğrudur?

- A) linkedList'de her eleman Node olarak adlandırılır.
Her Node iki bolumden oluşur:
 - 1) Data/Value
 - 2) Pointer/Address/Reference.
- B) Head sadece ilk node'un address'ini içerir.
- C) Son node'un pointer'i null'i point eder.
- D) Insert ve delete işlemleri LinkedList'de kolaylıkla yapılır. Bu yüzden insert ve delete işlemlerinin çoğunlukla yapılması gereken durumlarda LinkedList kullanılır.

5. Knowledge:

- 1) addFirst() method LinkedList'in en başına eleman eklemek için kullanılır.
- 2)addLast() method LinkedList'in en sonuna eleman eklemek için kullanılır.

```
LinkedList<String> linkedList = new LinkedList<>();
```

```
linkedList.add("A");
```

```
linkedList.add("B");
```

```
linkedList.add("C");
```

```
linkedList.add("B");
```

```
linkedList.add("D");
```

Hangisi doğrudur?

- A) linkList.remove(2);
LinkedList'den B'yi siler.
- B) linkedList.add(1, "Ali"); ve linkedList.addFirst("Ali");
aynı işlemi yapar.
- C) linkedList.addLast("Veli"); ve linkedList.add("Veli");
aynı işlemi yapar
- D) linkedList.remove() ve linkedList.removeLast() aynı işlemi yapar

6. Hangi senaryolar LinkedList kullanmak için uygundur?

- A) Müze ziyaretçilerini listelemek
- B) Bir ülkedeki şehirleri listelemek
- C) Bir şehirdeki posta kod'larını listelemek
- D) Bir toptan satış deposundaki ürünleri listelemek

7. `LinkedList<String> linkedList = new LinkedList<>();`

```
linkedList.add("A");  
linkedList.add("B");  
linkedList.add("C");  
linkedList.add("B");
```

Hangileri doğrudur?

- A) `linkedList.removeLastOccurrence("B");`
`System.out.println(linkedList);`
Console'a [A, B, C] yazar
- B) `System.out.println(linkedList.get(2));`
Console'a C yazar.
- C) `Collections.sort(linkedList);`
`System.out.println(linkedList);`
Console'a [C, B, B, A] yazar
- D) `linkedList.set(0, "X");`
`System.out.println(linkedList);`
Console'a [X, A, B, C, B] yazar

8. `public class Test02 {`
 `public static void main(String[] args) {`
 `LinkedList<String> sll = new LinkedList<>();`
 `sll.add("B");`
 `sll.add("A");`
 `sll.addFirst("B");`
 `sll.addFirst("C");`
 `sll.add("B");`
 `}`
`}`

Hangileri doğrudur?

- A) `System.out.println(sll);`
Console'a [C, B, B, A, B] yazar
`sll.removeFirstOccurrence("B");`
- B) `System.out.println(sll);`
Console'a [C, B, A, B] yazar
`sll.removeLastOccurrence("B");`
- C) `System.out.println(sll);`
Console'a [C, B, B, A] yazar
- D) `System.out.println(sll.getLast() + sll.getFirst() + sll.get(2));`
Console'a BCB yazar

9. Hangileri aynı output'u verir?

- A)

```
public static void main(String[] args) {  
    List<String> m = new ArrayList<>();  
    m.add("B");  
    m.add("C");  
    m.add("A");  
    Collections.sort(m);  
    for(int i = m.size()-1; i>=0; i--) {  
        System.out.print(m.get(i) + " ");  
    }  
}
```
- B)

```
public static void main(String[] args) {  
    List<String> m = new ArrayList<>();  
    m.add("B");  
    m.add("C");  
    m.add("A");  
    Collections.sort(m);  
    Collections.reverse(m);  
    for(int i = 0; i<m.size(); i++) {  
        System.out.print(m.get(i) + " ");  
    }  
}
```
- C)

```
public static void main(String[] args) {  
    List<String> m = new ArrayList<>();  
    m.add("B");  
    m.add("C");  
    m.add("A");  
    Collections.reverse(m);  
    for(int i = 0; i<m.size(); i++) {  
        System.out.print(m.get(i) + " ");  
    }  
}
```

10. `public static void main(String[] args) {`
 `LinkedList<String> c1= new LinkedList<String>();`
 `c1.add("Red");`
 `c1.add("Green");`
 `c1.add("Black");`
 `c1.add("White");`
 `c1.add("Pink");`
}

Hangileri doğrudur?

- A) `System.out.println(c1);`
Console'a [Red, Green, Black, White, Pink] yazar
- B) `System.out.println(c1.isEmpty());`
Console'a false yazar
`c1.removeAll(c1);`
- C) `System.out.println(c1);`
Console'a [] yazar
- D) `System.out.println(c1.removeAll(c1));`
Console'a false yazar

1. public class Test01 {

public static void main(String[] args) {

Queue<Integer> queue = new LinkedList<>();

queue.add(3);

queue.add(5);

queue.add(7);

queue.add(9);

queue.add(11);

queue.add(13);

int count = 0;

while (queue.isEmpty() == false) {

queue.remove();

count++;

if(count == 3) {

break;

}

}

System.out.println(queue);

}

}

Output nedir?

- A) [9, 11, 13]
- B) [3, 5, 7]
- C) [11, 13]
- D) [7, 9, 11, 13]


```
2. public class Test02 {  
    public static void main(String[] args) {  
        TreeSet<String> tSet = new TreeSet<>();  
        tSet.add("C");  
        tSet.add("B");  
        tSet.add("C");  
        tSet.add("A");  
        for (String w : tSet){  
            System.out.print(w + " ");  
        }  
    }  
}
```

Output nedir?

- A) A B C
- B) C B A
- C) C B C A
- D) A B C C

3. Hangileri doğrudur?

- A) Eleman ekleme ve silme işlemlerini çok yaparsanız LinkedList kullanmak en iyi seçenektir.
- B) Natural order'da unique elemanlar içeren bir collection lazım olduğunda önce HashSet oluşturup elemanları ekledikten sonra, HashSet'i TreeSet'e çevirmek iyi bir yöntemdir.
- C) Collection'daki elemanları tekrar tekrar bulmanız gerekirse veya tekrar tekrar update etmeniz gerekirse ArrayList kullanmak iyidir.
- D) Inertion order'da unique elemanlar içeren bir collection lazım olduğunda HashSet kullanmak iyidir.

4. 1) Elemanlar random order'dadır.
2) Tekrarli eleman kullanilmaz.
3) null value'lara izin verilir.
4) Eleman eklemede cok hizlidir.

Yukarıdaki ozelliklerin hepsi aşağıdakilerden hangisinde vardır?

- A) HashSet
- B) LinkedHashSet
- C) PriorityQueue
- D) LinkedList

5. Hangileri doğrudur?

- A) remove() ve poll() method'larinin ikisi de queue'daki ilk elemanı siler ve return eder.
- B) element() ve peek() method'larinin ikisi de queue'daki ilk elemanı silmeden return eder.
- C) Eğer queue bos ise, element() methodu "NoSuchElementException" atar, peek() methodu null return eder.
- D) Eğer queue bos ise, remove() methodu "NoSuchElementException" atar, poll() methodu null return eder.

6. Hangileri doğrudur?

- A) İlk eklenen elemanın, ilk kullanılmasını gerektiren durumlarda Queue kullanmak iyidir.
- B) Knowledge: ArrayList'ler index kullanır fakat Queue'lar index kullanmaz.
Queue'lar element ekleme ve silmede başarılıdır, fakat istenilen elemanı bulmakta yavaşlardır. Diğer taraftan, ArrayList'ler eleman ekleme ve silmede biraz yavaş olmalarına rağmen istenilen elemanı bulmakta başarılıdır.
- C)

```
Queue<String> q = new PriorityQueue<>();  
q.add("Teddy");  
q.add("Mark");  
q.add("Leo");  
q.remove();
```

 kesinlikle Leo elemanını siler

```
7. public class Test05 {  
    public static void main(String[] args) {  
        HashSet<String> hSet = new HashSet<>();  
        hSet.add("Z");  
        hSet.add("Y");  
        hSet.add("X");  
        hSet.add("X");  
        System.out.println(hSet);//A  
        TreeSet<String> tSet = new TreeSet<>(hSet);  
        System.out.println(tSet);//B  
    }  
}
```

Hangisi doğrudur?

- A) A kesinlikle [X, Y, Z] dir
- B) B kesinlikle [X, Y, Z] dir
- C) A [X, X, Y, Z] olabilir
- D) B [X, X, Y, Z] olabilir

8. Eslestirmelerden hangileri doğrudur?

- A) Uniqueness of elements ==> HashSet, TreeSet, LinkedHashSet
- B) Insertion order ==> ArrayList, LinkedList, LinkedHashSet
- C) FIFO ==> Queue
- D) Eleman ekleme ve silmede basari ==> LinkedList, HashSet

```
9. public class Test01 {  
    public static void main(String[] args) {  
        HashSet<String> hs = new HashSet<>();  
        hs.add("A");  
        hs.add("B");  
        hs.add("A");  
        hs.add(null);  
        hs.add(null);  
        System.out.println(hs);  
    }  
}
```

Aşağıdakilerden hangisi output olabilir?

- A) [null, A, B]
- B) [null, null, A, B]
- C) [null, A, A, B]
- D) [null, null, A, A, B]

10. Hangileri doğrudur?

- A) Set'ler tekrarlı eleman icerebilir fakat List'ler içeremez.
- B) List'ler elemanları ekleme sirasina göre dizer fakat HashSet'ler elemanları rastgele dizer.
- C) Hashing technique her data için kod'lar oluşturur.
- D) HashSet, Set'lerin en yavaşdır.

- 1) Herhangi bir 3 öğenin fiyatları için double data tipinde değişkenler oluşturunuz. Fiyatların toplamını konsola yazdırınız.**

```
double gomlekFiyat = 12.99;  
double sapkaFiyat= 23.99;  
double ayakkabiFiyat = 15.99;  
System.out.println("Toplam Fiyat: "+ (gomlekFiyat + sapkaFiyat + ayakkabiFiyat));
```

- 2) Herhangi bir 3 öge için float, long ve integer data tipinde değişkenler oluşturunuz. Bu değişkenlerin değerlerinin çarpımını konsolda yazdırınız.**

```
float x = 15.99F;  
long y = 13L;  
int z = 12;  
System.out.println("Degiskenlerin carpimi: " + x*y*z);
```

- 3) Basit faizi bulmak için bir kod yazınız.**

Not: Basit faiz formülü = anapara * oran * yılDegeri /100

```
int anapara = 10000, oran = 6, yılDegeri = 3;  
int basitFaiz = anapara * oran * yılDegeri /100;  
System.out.println("Basit faiz degeri:" + basitFaiz);
```

- 4) Bir String ve iki long data tipinde değişken oluşturunuz. Konsolda bu long değişkenlerin toplamını ve çarpımını String ile yazdırınız.**

```
String s = "Sonuc ";  
Long a = 123L;  
Long b = 9123456782L;  
System.out.println(s + (a+b));  
System.out.println(s + a*b);
```

- 5) Değerleri farklı olan iki boolean data tipinde değişken oluşturunuz ve bu değerleri iki ardışık değer arasında boşluk bırakacak şekilde aynı satırda yazdırınız.

```
boolean a = true;
boolean b = false;
System.out.println(a + " " + b);
```

- 6) 3 öge (kitap, defter ve laptop) fiyatı için long data tipinde değişkenler oluşturunuz. Konsolda 2 kitap, 4 defter ve 3 laptop için toplam fiyatını yazdırınız.

```
float kitap = 12.99F, defter = 23.45F, laptop = 34.12F;
float toplamFiyat = 2* kitap + 4* defter + 3*laptop;
System.out.println("Toplam fiyat : " + toplamFiyat);
```

- 7) İki tamsayıyı değiştirmek için bir kod yazınız.

```
Scanner scan = new Scanner(System.in);
System.out.println("2 sayı giriniz :");
double n1 = scan.nextDouble();
double n2 = scan.nextDouble();
System.out.println("Degistirmeden once : " + n1 + " - " + n2);
```

1.Yol: 3. değişken kullanarak

```
double temp = n1;
n1 = n2;
n2 = temp;
System.out.println("Degistirdikten sonra : " + n1 + " - " + n2);
```

2.Yol: 3. değişken kullanmadan

```
n1 = n1 + n2;
n2 = n1 - n2;
n1 = n1 - n2;
System.out.println("Degistirdikten sonra : " + n1 + " - " + n2);
```

- 1) Short data tipinde bir değişken oluşturunuz ve onu bir int değişkenine dönüştürünüz.**

```
short yas = 12;  
int yeniYas = yas;
```

- 2) Long data tipinde bir değişken oluşturunuz ve onu bir int değişkenine dönüştürünüz.**

```
long yas = 12L;  
int yeniYas = (int) yas;
```

- 3) Double data tipinde bir değişken oluşturunuz ve onu bir float değişkenine dönüştürünüz.**

```
double fiyat = 2.34;  
float yeniFiyat = (float) fiyat;
```

- 4) Double data tipinde bir değişken oluşturunuz ve onu bir short değişkenine dönüştürünüz. Sonrasında bu short değişkenin değerini konsolda yazdırınız.**

Not : Çıktıya dikkat ediniz, ondalık bir değer olmayacaktır.

```
double fiyat = 1.2;  
short yeniFiyat = (short) fiyat;  
System.out.println(yeniFiyat);
```

- 5) Byte data tipinde bir değişken oluşturunuz ve onu bir double değişkenine dönüştürünüz. Sonrasında bu double değişkenin değerini konsolda yazdırınız.**

Not : Çıktıya dikkat ediniz, ondalık bir değer olmayacaktır.

```
byte age = 23;  
double newAge = age;  
System.out.println(newAge);
```


- 1) Byte data tipinin minimum değeri ile short data tipinin maksimum değerinin toplamını bulmak için bir kod yazınız.

```
byte min = Byte.MIN_VALUE;  
short max = Short.MAX_VALUE;  
System.out.println(min + max);
```

- 2) Data tipi String olan "103" değerini byte data tipine dönüştürmek için bir kod yazınız ve yine data tipi String olan "2351" değerini short data tipine dönüştürüp konsolda iki değişken arasındaki farkı yazdırınız.

```
byte sayi1 = Byte.valueOf("113");  
short sayi2 = Short.valueOf("2351");  
System.out.println(sayi2 - sayi1);
```

- 3) `valueOf(String s)` ile `parseByte(String s)` methodları arasındaki farklar nelerdir?

`parseByte()` ilkel tamsayı (primitive integer) tipini (int) döndürürken; `valueOf()`, ilkel tamsayının nesne temsilcisi olan `Java.lang.Integer`'ı döndürür. Primitive türler yerine, bir tamsayı nesnesini isteyebileceğimiz durumlar da vardır.

Bariz olan bir diğer fark ise; `valueOf()` bir örnek metod(instance method) iken `parseByte()` statik bir metoddur.

- 1) Sadece tek bir kelimeden oluşan şehir isimleri için bir String değişkeni oluşturun. Şehir ismini, baş harfi büyük ve diğer tüm karakterleri küçük harfler olacak şekilde konsolda yazdırınız.**

Örnek: mIAMI - Miami
miami - Miami
MIAMI - Miami
mIaMi - Miami vb.

```
String sehirlsim = " mIAMI ";  
String konsolSehirlsim = sehirlsim.trim().toLowerCase();  
konsolSehirlsim = konsolSehirlsim.substring(0,1).toUpperCase() + konsolSehirlsim.substring(1);  
System.out.println(konsolSehirlsim);
```

- 2) Kişi isimleri için 3 String değişken oluşturunuz. Boşluk karakterleri hariç 3 isimdeki karakter sayısının toplamını yazdırınız.**

Örnek: İsimler Ali Can, Merve Star, Mark Tom ise konsolda 22 görmelisiniz.

```
String isim1 = "Ali Can";  
String isim2 = "Aliye Canan";  
String isim3 = "Aliyev Can Cananov";  
Integer c1 = isim1.replaceAll("\\s","").length();  
Integer c2 = isim2.replaceAll("\\s","").length();  
Integer c3 = isim3.replaceAll("\\s","").length();  
System.out.println("Bosluk karakteri olmadan 3 ismin toplam karakter sayisi: " + (c1 + c2 + c3));
```

- 3) Bir String değişkeni oluşturunuz, String'deki toplam alfabetik ve sayısal karakter sayısını konsolda yazdırınız.**

Örnek: String : ' Miami 33018!!! ' ise konsolda 10 yazdırmanız gerekir.

```
String s = " Miami 33018!!! ";  
Integer karakterSayisi = s.replaceAll("\\s","").replaceAll("[_!A-Za-z0-9]","").length();  
System.out.println("Toplam alfabetik ve sayısal karakter sayisi:" + karakterSayisi);
```

String Manipulations

- 4) Bir String değişkeni oluşturunuz, String'deki rakam olmayan karakterlerin sayısını konsolda yazdırınız.

Örnek: String: '1a3Bcf4!...' ise konsolda 8 yazdırmanız gerekir.

```
String s = "2a3B4?-!5";  
Integer rakamOlmayanKrkSayisi = s.replaceAll("[0-9]","").length();  
System.out.println("rakam olmayan toplam karakter sayisi: " + rakamOlmayanKrkSayisi);
```

- 5) Bir String değişkeni oluşturunuz ve String değişkenin konsolda boşluk olmayan son karakteri yazdırınız.

Örnek: 'Ali Can' için n yazdırmalısınız.

'Miami' için i yazdırmalısınız.

```
String s = "Miami  ";  
int boslukOlmayanSonKrkIndex= s.trim().length()-1;  
String boslukOlmayanSonKrk = s.substring(boslukOlmayanSonKrkIndex,  
boslukOlmayanSonKrkIndex +1);  
System.out.println("Bosluk olmayan son karakter: " + boslukOlmayanSonKrk);
```

- 6) Bir String değişkeni oluşturunuz ve String'in ilk ve son karakterlerinin ASCII değerlerinin toplamını bulunuz.

```
String s = "Miami";  
int sonKrkIndex = s.length()-1;  
int ilkKrkAscii= s.charAt(0);  
int sonKrkAscii = s.charAt(sonKrkIndex);  
System.out.println("ilk ve son karakterlerin ASCII degerlerinin toplami: " + (ilkKrkAscii +  
sonKrkAscii));
```

String Manipulations

- 7) **Bir String değişkeni oluşturunuz ve ilk karakteri dışındaki tüm karakterleri konsolda yazdırınız.**

Örnek: String 'Java' ise konsola 'ava' yazdırmalısınız.

```
String s = "Java";  
String ilkKrkDisindakiler = s.substring(1);  
System.out.println("ilk karakter disindaki tum karakterleri: " + ilkKrkDisindakiler);
```

- 8) **Bir String değişkeni oluşturunuz ve son karakteri dışındaki tüm karakterlerini konsolda büyük harfle yazdırınız.**

Örnek: String 'Java' ise konsola 'JAV' yazdırmalısınız.

```
String s = "Java";  
Integer sonKrkIndex = s.length()-1;  
String sonKrkDisindakiler = s.substring(0, indexOfLastChar).toUpperCase();  
System.out.println("son karakter disindaki tum karakterler buyuk harf ile : " + sonKrkDisindakiler);
```

- 9) **Bir String değişkeni oluşturunuz ve konsolda ilk karakter ve son karakter dışındaki tüm karakterleri büyük harflerle yazdırınız.**

Örnek: String 'Java' ise konsolda 'AV' yazdırmalısınız.

```
String s = "Java";  
Integer sonKrkIndex = s.length()-1;  
String ilkSonKrkDisindakiler = s.substring(1, sonKrkIndex).toUpperCase();  
System.out.println("ilk ve son karakter disindaki tum karakterler buyuk harf ile : " +  
ilkSonKrkDisindakiler);
```

String Manipulations

- 10) Bir String in ortadaki herhangi bir konumda yalnızca tek bir boşluk karakteri olup olmadığını kontrol etmek için kod yazınız.**

Örnek: 'Ali Can' için konsolda false yazmalıdır.

' Ali Can ' için konsolda false yazmalıdır.

' Ali Can ' için konsolda false yazmalıdır.

'Ali Can' için konsolda true yazmalıdır.

```
String s = "Tom Hanks";
```

```
String trimlenmis = s.trim();
```

```
String hicBoslukYok = trimlenmis.replaceAll("\\s", "");
```

```
Boolean ortadaTekBoslukVarMi = trimlenmis.length() - hicBoslukYok.length() == 1;
```

```
System.out.println("ortada tek bosluk var " + ortadaTekBoslukVarMi);
```

- 11) Bir String'in başında ve sonunda boşluk karakteri olup olmadığını kontrol etmek için kod yazınız.**

Örnek: ' Ali ' için kodunuz konsolda false yazmalıdır

'Ali' için kodunuz konsolda true yazmalıdır

```
String s = " Tom ";
```

```
String trimlenmis = s.trim();
```

```
Boolean sonuc = s.equals(trimlenmis);
```

```
System.out.println("basinda ve sonunda bosluk var mi : " + !sonuc);
```

- 12) Bir String'in başında büyük harf ve sonunda nokta olup olmadığını kontrol etmek için kod yazınız.**

Örnek: 'Ali' için kodunuz konsolda false yazdırmalıdır

'ali.' için kodunuz konsolda false yazdırmalıdır

' Ali. ' için kodunuz konsolda false yazdırmalıdır

'Ali.' için kodunuz konsolda true yazdırmalıdır

'ALI.' için kodunuz konsolda true yazdırmalıdır

```
String s = "Tom Hanks.";
```

```
char ilkKrk = s.charAt(0); char sonKrk = s.charAt(s.length()-1);
```

```
Boolean ilkKrkBuyukMu = ilkKrk >='A' && sonKrk <='Z';
```

```
Boolean sonKrkNoktaMi = sonKrk == '.';
```

```
Boolean ilkKrkBuyukSonKrkNoktaMi = ilkKrkBuyukMu && sonKrkNoktaMi;
```

```
System.out.println("ilk karakter buyuk harf, son karakter nokta mi? " + ilkKrkBuyukSonKrkNoktaMi);
```

- 13) Aşağıdaki koşullar için bir parolanın geçerli olup olmadığını kontrol etmek için kod yazınız; Şifre boşluk karakterinden farklı olacak şekilde en az 8 karakter içermelidir. Şifrede en az 1 sembol olmalıdır.**

Örnek:'A2b!' için kodunuz konsolda false yazdırmalıdır.

'A2b3cdef' için kodunuz konsolda false yazdırmalıdır

'!1a23b4' için kodunuz konsolda false yazdırmalıdır.

'!1a23b4?es' için kodunuz konsolda true yazdırılmalıdır.

'! a b 3 k' kodunuz konsolda false yazdırmalıdır.

```
String sifre = "!1a23b4?es";
```

```
Boolean enAzSekizKrkMi= sifre.replaceAll("\\s","").length()>=8;
```

```
Boolean enAzBirSembolVarMi = sifre.replaceAll("\\s","").replaceAll("[A-Za-z0-9]","").length()>0;
```

```
System.out.println("sifre gecerli mi ? " + (enAzSekizKrkMi && enAzBirSembolVarMi));
```

- 14) Bir String'in belirli bir tek karaktere sahip olup olmadığını üç farklı şekilde kontrol etmek için kod yazınız.**

1. Yol:

```
String s = "Java";
```

```
Boolean sonuc = s.contains("v");
```

```
System.out.println("karakteri iceriyor mu? " + sonuc);
```

2. Yol:

```
String s = "Java";
```

```
Boolean sonuc = s.replaceAll("[\\u0000-\\u000F]", "").length()>0;
```

```
System.out.println("karakteri iceriyor mu? " + sonuc);
```

3. Yol:

```
String s = "Java";
```

```
Boolean sonuc = s.indexOf("v")!= -1;
```

```
System.out.println("karakteri iceriyor mu? " + sonuc);
```

String Manipulations

15) String `gomlekFiyat = '$12.99';`

`String kitapFiyat = '$35.99';`

Gömlek ve kitap fiyatlarının toplamını bulmak için gereken kodu yazınız.

```
String s = gomlekFiyat.replaceAll("[$.]", "");
String b = kitapFiyat.replaceAll("[$.]", "");
double x = Double.valueOf(s);
double y = Double.valueOf(b);
System.out.println("Toplam Fiyat : " + (x+y)/100);
```

16) Verilen bir ismin adının ve soyadının baş harflerini almak için kodu yazınız.

Not: İkinci isim kapsam dışındadır.

Örnek: Tom Hanks ==> TH, Mary Star ==> MS

1.Yol:

```
String isim = "Tom Hanks";
String adIlkHarf = isim.substring(0,1);
int soyadIlkHarfIndex = isim.indexOf(" ") + 1;
String soyadIlkHarf= isim.substring(soyadIlkHarfIndex, soyadIlkHarfIndex + 1);
System.out.println(adIlkHarf + soyadIlkHarf);
```

2.Yol:

```
String isim = "Tom Hanks";
String adIlkHarf = isim.substring(0,1);
String soyadIlkHarf = isim.split(" ")[1].substring(0, 1);
System.out.println(adIlkHarf + soyadIlkHarf);
```

17) Bir String' de kullanılan noktalama işaretlerinin sayısını bulmak için bir kod yazınız.

```
String s = "Vay be! Ali 13 yaşında ama üniversite öğrencisi.";
int toplamKrkSayisi = s.length();
int noktalamalsaretleriDisindakilerinSayisi = s.replaceAll("\\p{Punct}", "").length();
int noktalamalsaretlerininSayisi= toplamKrkSayisi - noktalamalsaretleriDisindakilerinSayisi;
System.out.println(noktalamalsaretlerininSayisi);
```

1) Ali'nin kaç gün yaşadığını bulan kodu yazınız.

Ali'nin doğum tarihi 12 Mayıs 2002'dir.

```
LocalDate dogumGunuTarihi = LocalDate.of(2002, 5, 12);  
LocalDate bugunTarihi = LocalDate.now();  
long yasadigiGunSayisi = ChronoUnit.DAYS.between(dogumGunuTarihi, bugunTarihi);  
System.out.println("Ali' nin su ana kadar yasadigi gun sayisi : " + yasadigiGunSayisi);
```

2) Ali'nin kaç ay yaşadığını bulan kodu yazınız.

Ali'nin doğum tarihi 4 Haziran 1997'dir.

```
LocalDate dogumGunuTarihi = LocalDate.of(1997, 06, 04);  
LocalDate bugunTarihi = LocalDate.now();  
long yasadigiAySayisi = ChronoUnit.MONTHS.between(dogumGunuTarihi, bugunTarihi);  
System.out.println("Ali'nin su ana kadar yasadigii ay sayisi : " + yasadigiAySayisi);
```

3) Ali'nin doğum tarihi 4 Haziran 1997'dir. Ali'nin doğum tarihinden 2 yıl, 3 ay ve 12 gün sonraki tam tarihi bulmak için kodu yazınız.

```
LocalDate dogumGunuTarihi = LocalDate.of(1997, 06, 04);  
LocalDate kesinTarih = dogumGunuTarihi.plusYears(2).plusMonths(3).plusDays(12);  
System.out.println("Kesin Tarih : " + kesinTarih);
```


4) Ali, 29 Ekim 1923'ten 45 yıl 8 ay 5 gün sonra doğmuştur.

Veli, 15 Eylül 1993'ten 24 yıl 2 ay 11 gün önce doğmuştur.

Ali ve Veli'nin kesin doğum tarihini hesaplamak için kodu yazınız.

Ali ve Veli'nin doğum tarihinin aynı olup olmadığını kontrol etmek için kodu yazınız.

```
LocalDate dogumGunuTarihiAli =  
    LocalDate.of(1923,10,29).plusYears(45).plusMonths(8).plusDays(5);  
LocalDate dogumGunuTarihiVeli =  
    LocalDate.of(1993, 9, 15).minusYears(24).minusMonths(2).minusDays(11);  
System.out.println("Tarihler aynı mi ? : " + dogumGunuTarihiAli.equals(dogumGunuTarihiVeli));
```

5) Veli, Ali'den 3 yıl 11 gün sonra doğmuştur.

Ali size doğum tarihi bilgisini 24 Kasım 2012 olarak vermiştir.

Veli'nin doğum tarihini hesaplamak için gerekli kodu yazınız.

```
LocalDate dogumGunuTarihiAli = LocalDate.parse("2012-11-24");  
LocalDate dogumGunuTarihiVeli = dogumGunuTarihiAli.plusYears(3).plusDays(11);  
System.out.println("Veli'nin dogum tarihi " + dogumGunuTarihiVeli);
```

6) Kendinizin ve çocuğunuzun doğum tarihi için Tarih Değerleri oluşturup, ardından farkı gün olarak hesaplayınız.

```
LocalDate benimDogumTarihim = LocalDate.of(1986, 8, 21);  
LocalDate cocugumunDogumTarihi = LocalDate.of(2015, 3, 12);  
long fark = ChronoUnit.DAYS.between(benimDogumTarihim, cocugumunDogumTarihi);  
System.out.println("iki tarih arasindaki gun sayisi : " + fark);
```

7) Belirli bir tarihte yılın son 2 hanesini alınız.

```
LocalDate bugunTarih = LocalDate.of(1996, 8, 21);  
int sonlkiRakam = bugunTarih.getYear()%100;  
System.out.println("yilin son 2 hanesi : " + sonlkiRakam);
```

8) Belirli bir yılın "Artık yıl" olup olmadığını kontrol etmek için kodu yazınız.

Artık Yıl:

- i) Bir yıl 100'e ve 400'e tam bölünüyorsa buna artık yıl denir. Örneğin; 2000 artık yıl, 1900 değildir.
- ii) Bir yıl 100'e ve 4'e tam bölünemiyorsa artık yıl olarak adlandırılır. Örneğin; 2004 artık yıl, 2007 değildir.

```
LocalDate tarih = LocalDate.of(1996, 8, 21);  
boolean artikYilMi = tarih.isLeapYear();  
System.out.println("tarihteki yıl artık yıldır " + artikYilMi);
```

9) İki farklı tarihin ay numaralarının toplamını bulunuz.

```
LocalDate tarih1 = LocalDate.of(1996, 8, 21);  
LocalDate tarih2 = LocalDate.of(2015, 3, 12);  
System.out.println("Toplam ay degerleri : " + tarih1.getMonth() + tarih2.getMonth());
```

10) İki farklı tarih için saat farkını bulunuz.

```
LocalDate tarih1 = LocalDate.of(1996, 8, 21);  
LocalDate tarih2 = LocalDate.of(2015, 3, 12);  
long saatFarki = ChronoUnit.HOURS.between(tarih1, tarih2);  
System.out.println("Saatler arasindaki fark : " + saatFarki);
```

1) Eğer saat

- i) 24:00 ile 05:00 arasında ise konsola 'Uyku zamanı' yazdırınız.
- ii) 08:00 ile 16:00 arasında ise konsola 'Çalışma zamanı' yazdırınız.
- iii) 19:00 ile 22:00 arasında ise konsola 'Aile zamanı' yazdırınız.
- iv) Diğerleri için konsolda 'Kişisel zaman' yazdırınız.

```
LocalTime bugunTarihi = LocalTime.now();  
int saat = now.getHour();
```

```
if(saat >0 && saat <5){  
    System.out.println("Uyku zamani");  
}else if(saat >8 && saat <16){  
    System.out.println("Calisma Zamani");  
}else if(saat >19 && saat <22){  
    System.out.println("Aile Zamani");  
}else{  
    System.out.println("Kisisel Zaman");  
}
```

2) Japonya saat dilimi ile Almanya saat dilimi arasındaki saat farkını bulmak için kodu yazınız.

```
LocalDateTime zamanJaponya = LocalDateTime.now(ZoneId.of("Asia/Tokyo"));  
System.out.println(zamanJaponya);  
LocalDateTime zamanAlmanya = LocalDateTime.now(ZoneId.of("Europe/Berlin"));  
System.out.println(zamanAlmanya);  
long fark = ChronoUnit.HOURS.between(zamanAlmanya, zamanJaponya);  
System.out.println(fark);
```

- 3) Ali 5 Şubat 2015 TRT 10:00'da İstanbul' da, Mark 5 Şubat 2015 12:00 EST' de ABD' de doğmuştur. Ali'nin doğum saati ile Mark'ın doğum saati arasındaki saat cinsinden fark nedir?**

```
LocalDateTime dogumTrhAli = LocalDateTime.of(2015, 2, 5, 10, 0, 0);
ZonedDateTime yerelDogumTrhAli = dogumTrhAli.atZone(ZoneId.of("Europe/Istanbul"));
LocalDateTime dogumTrhVeli = LocalDateTime.of(2015, 2, 5, 12, 0, 0);
ZonedDateTime yerelDogumTrhVeli = dogumTrhVeli.atZone(ZoneId.of("America/New_York"));
long saatFarki = ChronoUnit.HOURS.between(yerelDogumTrhAli, yerelDogumTrhVeli);
System.out.println("Saatler arasi fark : " + saatFarki);
```

- 4) Ali 5 Şubat 2015 saat 10:15 EST'de doğmuştur, Veli 6 Mart 2015 saat 12:25 EST'de doğmuştur. Ali ve Veli'nin doğum süreleri arasındaki fark nedir?**

```
LocalDateTime dogumTrhAli = LocalDateTime.of(2015, 2, 5, 10, 15, 0);
LocalDateTime dogumTrhVeli = LocalDateTime.of(2015, 3, 6, 12, 25, 0);
long saatFarki = ChronoUnit.MINUTES.between(dogumTrhAli, dogumTrhVeli);
System.out.println("Saatler arasi fark : " + saatFarki);
```

- 5) Mark 23.02.2018 tarihinde saat 15:25'te GMT ' de doğmuştur, Mark'ın Japonya'daki tam doğum tarihi ve saati nedir?**

```
LocalDateTime dogumTrhTokyo = LocalDateTime.now(ZoneId.of("Asia/Tokyo"));
LocalDateTime dogumTrhGMT= LocalDateTime.now(ZoneId.of("GMT"));
long saatFarki = ChronoUnit.HOURS.between(dogumTrhGMT, dogumTrhTokyo);
LocalDateTime dogumTrhMark = LocalDateTime.of(2018, 2, 23, 15, 25, 0).minusHours(saatFarki);
System.out.println("Tokyo daki dogum tarihi : " + dogumTrhMark);
```

1) Aşağıdaki seçenekler doğrultusunda gereken kodu yazınız.

- a) Aralık, Ocak, Şubat için "Kış"
- b) Mart, Nisan, Mayıs için "İlkbahar"
- c) Haziran, Temmuz, Ağustos için "Yaz"
- d) Eylül, Ekim, Kasım için "Sonbahar"
- e) Diğerleri için "Geçersiz ay adı"

```
String ayAdi = "Aralik";
ayAdi = ayAdi.toLowerCase();
if(ayAdi.equals("aralik ") || ayAdi.equals("ocak") || ayAdi.equals("subat")){
    System.out.println("Kis");
}else if(ayAdi.equals("mart") || ayAdi.equals("nisan") || ayAdi.equals("mayis")){
    System.out.println("Ilkbahar ");
}else if(ayAdi.equals("haziran") || ayAdi.equals("temmuz") || ayAdi.equals("agustos")){
    System.out.println("Yaz ");
}else if(ayAdi.equals("eylul") || ayAdi.equals("ekim") || ayAdi.equals("kasim")){
    System.out.println("Sonbahar ");
}else{
    System.out.println("Gecersiz ay adi ");
}
```

2) Aşağıdaki seçenekler doğrultusunda gereken kodu yazınız.

- a) Şifre boşluk karakterinden farklı en az 8 karakter içeriyorsa "Geçerli Şifre"
- b) Herhangi bir pozisyonda şifrede boşluk karakteri varsa "Şifrede boşluk karakteri kullanmayınız"
- c) a ve b koşullarının sağlanmaması durumunda "Geçersiz Şifre"

Not: Çözümdeki koşulların sıralarına dikkat ediniz.

```
String sifre = "a1b2c3d4";
if(sifre.replaceAll("\\s", "").length()>0){
    System.out.println("Sifrede bosluk karakteri kullanmayiniz");
}else if(sifre.replaceAll("\\s", "").length()>=8){
    System.out.println("Gecerli Sifre");
}else{
    System.out.println("Gecersiz Sifre ");
}
```

3) Ayın numarasını girdiğinizde ayın adını yazdırmak için gereken kodu yazınız.

Örnek; 1 için çıktı: "Ocak", 2 için çıktı: "Şubat" vb.

```
int ayNo = 3;
if(ayNo ==1){
    System.out.println("Ocak");
}else if(ayNo ==2){
    System.out.println("Subat");
}else if(ayNo ==3){
    System.out.println("Mart");
}else if(ayNo ==4){
    System.out.println("Nisan");
}else if(ayNo ==5){
    System.out.println("Mayis");
}else if(ayNo ==6){
    System.out.println("Haziran");
}else if(ayNo ==7){
    System.out.println("Temmuz");
}else if(ayNo ==8){
    System.out.println("Agustos");
}else if(ayNo ==9){
    System.out.println("Eylul");
}else if(ayNo ==10){
    System.out.println("Ekim");
}else if(ayNo ==11){
    System.out.println("Kasim");
}else if(ayNo ==12){
    System.out.println("Aralik");
}else{
    System.out.println("Gecersiz ay numarasi");
}
```

If Statement

4) Aşağıdaki seçenekler doğrultusunda gereken kodu yazınız.

- a) Son basamak 5 'e eşit ve 5 den büyükse, yuvarlama işlemi: "Son basamağı bir üst ondalığa yuvarla"
- b) Son basamak 5 den küçükse, yuvarlama işlemi: "Son basamağı bir alt ondalığa yuvarla"

```
int i = 124;
if(i%10>=5){
    System.out.println("Son basamağı bir üst ondalığa yuvarla :"+ (i/10+1)*10);
}else{
    System.out.println("Son basamağı bir alt ondalığa yuvarla :"+ (i/10)*10);
}
```

5) Aşağıdaki seçenekler doğrultusunda gereken kodu yazınız.

- a) Bir üçgenin iki kenarının uzunluğu eşitse "İkizkenar Üçgen".
- b) Bir üçgenin tüm kenarlarının uzunluğu eşitse "Eşkenar Üçgen".
- c) a ve b koşulları sağlanmıyorsa "Çeşit Kenar Üçgen"

```
int a = 5;
int b = 3;
int c = 4;

if (a == b && b == c) {
    System.out.println("Eşkenar Ucgen");
} else if (a == b && b != c || a == c && b != c || b == c && a != c) {
    System.out.println("Ikizkenar Ucgen ");
} else {
    System.out.println("Cesit Kenar Ucgen ");
}
```

If Statement

- 6) Mili km'ye, saniyeyi saate, fahrenheitı santigrata çeviren bir dönüştürücü oluşturmak için kod yazınız. Google'dan dönüşümler için formülleri bulunuz.
- a) Kullanıcı mile değerini 10 ve operatorü mildenKmye olarak girdiğinde, konsolda kodunuz "16 km" yazmalıdır (sayı dinamik olacak)
 - b) Kullanıcı saniye değerini 7200 ve operatorü saniyedenSaate olarak girdiğinde, konsolda kodunuz "2" yazmalıdır (sayı dinamik olacak)
 - c) Kullanıcı fahrenheit değerini 83 ve operatorü fahrenheitdanSantigrata olarak girdiğinde, konsolda kodunuz "28.3333 santigrat" yazmalıdır (sayı dinamik olacak)

```
double mil = 10;
```

```
double saniye = 7200;
```

```
double fahrenheit = 83;
```

```
String operator = " fahrenheitdanSantigrata ";
```

```
if(operator.equals("mildenKmye")){
```

```
    System.out.println(mil*1.60934);
```

```
}else if(operator.equals("saniyedenSaate")){
```

```
    System.out.println((saniye /60)/60);
```

```
}else if(operator.equals("fahrenheitdanSantigrata")){
```

```
    System.out.println((fahrenheit -32)*5/9);
```

```
}
```


If Statement

- 7) Tam isim için dilbilgisi kurallarını kontrol etmek için kodu yazınız.
- a) Kodunuz "ali Can", "Ali can", "ali can" için "Baş harflerinde hata" yazmalıdır.
 - b) Kodunuz, aşağıdaki gibi tek kelimeler için "Ad veya soyadı eksik" yazmalıdır.
 - c) "Ali" veya "Can" veya "ali"Kodunuz "ALI CAN" gibi tüm formatlar için "Format hatası" yazmalıdır.
 - d) Kodunuz bir veya daha fazla boşluk (" " veya " " gibi karakterler) için "Isim girilmedi" yazmalıdır.
 - e) Ad, harfler ve boşluktan başka herhangi bir karakteri içeriyorsa, kodunuz "Geçersiz Ad" yazmalıdır.

Not : Kısaltmada 1'den fazla hata varsa, ilgili tüm hata mesajları çıktı olarak konsolda görünmelidir. Örneğin; "ali3" için kodunuz "Baş harflerinde hata", "Ad veya soyadı eksik" ve "Geçersiz Ad"

```
String isim = "ali Can?";
String adinIlkHarfi = isim.substring(0,1);
Integer boslukIndex = isim.trim().indexOf(' ');
String soyadinIlkHarfi = isim.substring(indexOfSpace+1,indexOfSpace+2);
Boolean adinIlkHarfiBuyukMu = (adinIlkHarfi.charAt(0)>='A' && adinIlkHarfi.charAt(0)<='Z');
Boolean soyadinIlkHarfiBuyukMu = (soyadinIlkHarfi.charAt(0)>='A' && soyadinIlkHarfi.charAt(0)<='Z');

if(boslukIndex ==-1){
    System.out.println("Ad veya soyadi eksik ");
}
if(!adinIlkHarfiBuyukMu || !soyadinIlkHarfiBuyukMu){
    System.out.println("Bas harflerinde hata");
}
if(isim.equals(isim.toUpperCase())){
    System.out.println("Format hatasi");
}
if(isim.replaceAll("\s","").length()==0){
    System.out.println("Isim girilmedi ");
}
if(isim.replaceAll("\s","").replaceAll("[A-Za-z]","").length()>0){
    System.out.println("Gecersiz Isim");
}
```

8) ABD'deki eyalet kısaltmalarının biçimini kontrol etmek için kod yazınız.

- a) Kısaltma 2'den fazla karakter içeriyor ise kodunuz, "Eyalet kısaltmaları 2 karakterden fazla olamaz" yazmalıdır.
- b) Kısaltmada küçük harfler var ise kodunuz, "Eyalet kısaltmaları küçük harf içeremez" yazmalıdır.
- c) Kısaltmanın harflerinden farklı karakterler var ise kodunuz, "Eyalet kısaltmaları aşağıdakilerden farklı karakterler içeremez" yazmalıdır.

Not: Kısaltmada 1'den fazla hata varsa ilgili tüm hata mesajları yazdırılmalıdır.

örnek; "fl3" için kodunuz; "Eyalet kısaltmaları 2 karakterden fazla olamaz", "Eyalet kısaltmaları küçük harf içeremez", ve "Eyalet kısaltmaları aşağıdakilerden farklı karakterler içeremez" yazmalıdır.

```
String eyaletKisaltma = "F13";
if(eyaletKisaltma.length()>2){
    System.out.println("Eyalet kisaltmalari 2 karakterden fazla olamaz ");
}
if(eyaletKisaltma.replaceAll("[\a-z]","").length()!=0){
    System.out.println("Eyalet kisaltmalari kucuk harf iceremez ");
}
if(eyaletKisaltma.replaceAll("[\A-Za-z]","").length()!=0){
    System.out.println("Eyalet kisaltmalari asagidakilerden farkli karakterler iceremez ");
}
```

If Statement

- 9) Herhangi bir 2 sayıyla toplama, çıkarma, çarpma ve bölme yapan basit bir hesap makinesi oluşturmak için kod yazınız.
- a) Kullanıcı 10.2 ve 5 ve + işaretini girdiğinde kodunuz konsolda "Sonuç 15.2" yazmalıdır.
 - b) Kullanıcı 10 ve 5 ve - işaretini girdiğinde kodunuz konsolda "Sonuç 5.0" yazmalıdır.
 - c) Kullanıcı 10 ve 5.3 ve * işaretini girdiğinde kodunuz konsolda "Sonuç 53.0" yazmalıdır.
 - d) Kullanıcı 10 ve -5 ve / işaretini girdiğinde kodunuz konsolda "Sonuç -2.0" yazmalıdır.

```
double a = 10.2, b=5;
String operator = "+";
if(operator.equals("+")){
    System.out.println(a+b);
}else if(operator.equals("-")){
    System.out.println(a-b);
}else if(operator.equals("*")){
    System.out.println(a*b);
}else if(operator.equals("/")){
    System.out.println(a/b);
}
```

- 10) Verilen BMI değerleri için Vücut Kitle İndeksi (BMI) ile ilgili mesajları yazdırmak için kod yazınız.

Geçersiz BMI değeri < 0

Zayıf = <18.5

Normal ağırlık = 18.5–24.9

Fazla kilolu = 25–29.9

Obezite = 30 veya daha büyük BMI

```
double bmi = 12;
if(bmi<0){
    System.out.println("Gecersiz BMI degeri");
}else if(bmi<18.5){
    System.out.println("Zayif ");
}else if(bmi<24.9 && bmi>=18.5){
    System.out.println("Normal agirlik ");
}else if(bmi<29.9 && bmi>=25){
    System.out.println("Fazla kilolu");
}else if(bmi>30){
    System.out.println("Obezite");
}
```

- 1) Ternary kullanarak, aşağıdaki kodları konsolda yazdırınız.
Parola boşluk karakterinden farklı olarak en az 8 karakter içeriyorsa, "Geçerli Parola",
Parola boşluk karakterinden farklı olarak 8 karakterden daha az ise, "Geçersiz Parola"

```
String parola = "a1b 2c3";  
String gecerliMi = parola.replaceAll("\\s","").length()>7 ? "Gecerli Parola" : "Gecersiz Parola";  
System.out.println(gecerliMi);
```

- 2) Ternary kullanarak, aşağıdaki kodları konsolda yazdırınız.
- a) Bir üçgenin iki kenarının uzunluğu eşitse "İkizkenar Üçgen".
 - b) Bir üçgenin tüm kenarlarının uzunluğu eşitse "Eşkenar Üçgen".
 - c) a ve b koşulları sağlanmıyorsa "Çeşit Kenar Üçgen "

```
int a = 5, b = 4, c = 3;  
String ucgenTipi = a==b && b==c ? "Eskenar Ucgen" : (a==b && b!=c || a==c && b!=c || b==c && a!=c ? "Ikizkenar Ucgen" : "Cesit Kenar Ucgen");  
System.out.println(ucgenTipi);
```

- 3) Ternary kullanarak, aşağıdaki kodları konsolda yazdırınız.
- a) Son basamak 5'e eşit ve 5 den büyükse, yuvarlama işlemi: "Son basamağı bir üst ondalığa yuvarla"
 - b) Son basamak 5 den küçükse, yuvarlama işlemi: "Son basamağı bir alt ondalığa yuvarla"

Örneğin; 127 yukarı yuvarlanacak ve değer 130 olacaktır.

125 yukarı yuvarlanacak ve değer 130 olacaktır.

123 aşağı yuvarlanacak ve değer 120 olacaktır.

```
int i = 125;  
String sonuc = i%10>=5 ? ("Son basamagi bir ust ondaliga yuvarla:" + (i/10+1)*10) : ("Son basamagi bir alt ondaliga yuvarla:" + (i/10)*10);  
System.out.println(sonuc);
```

Ternary Statement

4) Nested Ternary kullanarak Apex kodunu yazınız.

Bir yılın artık yıl olup olmadığını kontrol eden bir program yazınız.

Yıl 100'e tam bölünüyorsa 400'e tam bölünmelidir.

Bir yıl 100'e tam bölünemiyorsa 4'e tam bölünmelidir.

```
int y = 1800;
String artikYilMi = (y%100==0) ? ( (y%400==0) ? ("Artik Yil") : ("Artik Yil Degil") ) : ( (y%4==0 ? ("Artik Yil ") : ("Artik Yil Degil ") ));
System.out.println(artikYilMi);
```

5) Nested Ternary kullanarak;

Şifreyi kontrol etmek için kodu yazınız.

8'den fazla karakter içeriyorsa, baş harfi 'i' olmalıdır.

8 karakterden fazla içermiyorsa, baş harfi 'K' olmalıdır.

```
String p = "XXXXXXXXX";
String r = p.length()>8 ? p.startsWith("i") ? "Gecerli": "Gecersiz": p.startsWith("K") ? "Gecerli ": "Gecersiz";
System.out.println(r);
```

6) Bir sayının mutlak değerini hesaplamak için kodu yazınız.

Pozitif sayılar ve sıfır mutlak değer için sayı ile aynıdır.

Negatif sayıların mutlak değerini bulmak için sayıyı -1 ile çarpınız.

```
int i = -5;
int sonuc = i<0 ? -1*i : i;
System.out.println(sonuc);
```

Ternary Statement

- 7) Ternary kullanarak; 2 tamsayıdan küçük olanı konsola yazdıran bir program yazınız.

```
int i = 13, k = 12;  
int sonuc = i < k ? i : k;  
System.out.println(sonuc);
```

- 8) Sayı 3 basamaklı ise konsolda kodunuz "Bu sayı 3 basamaklıdır" olacaktır. Aksi takdirde, kodunuz " Bu sayı 3 basamaklı değildir" olacaktır.

1.Yol:

```
int i = 750;  
String r = (i > 99 && i < 1000) || (i > -1000 && i < -99) ? "Bu sayı 3 basamaklıdır " : "Bu sayı 3 basamaklı  
değildir";  
System.out.println(r);
```

2. Yol:

```
int i = -750;  
i = Math.abs(i);  
String r = (i > 99 && i < 1000) ? "Bu sayı 3 basamaklıdır " : "Bu sayı 3 basamaklı değildir ";  
System.out.println(r);
```

- 9) Ternary kullanarak; konsolda tek sayılar için "Tek" , çift sayılar için "Çift" yazdırınız.

```
int i = 14;  
String sonuc = i % 2 == 0 ? "Cift" : "Tek";  
System.out.println(sonuc);
```

- 10) Ternary kullanarak; konsolda sayı pozitif ise "Pozitif" , negatif ise "Pozitif Değil" yazdırınız.

```
int i = -12;  
String sonuc = i > 0 ? " Pozitif " : " Pozitif Degil ";  
System.out.println(sonuc);
```

1) Yazdırmak için switch ifadesini kullanınız.

- a) Aralık, Ocak, Şubat için "Kış"
- b) Mart, Nisan, Mayıs için "Bahar"
- c) Haziran, Temmuz, Ağustos için "Yaz"
- d) Eylül, Ekim, Kasım için "Güz"
- e) Diğerleri için "Geçersiz ay adı"

```
String ayAdi = "Haziran";
ayAdi = ayAdi.toLowerCase();
switch(ayAdi){
    case "aralik":
    case "ocak":
    case "subat":
        System.out.println("Kis");
        break;
    case "mart":
    case "nisan":
    case "mayis":
        System.out.println("Bahar");
        break;
    case "haziran":
    case "temmuz":
    case "agustos":
        System.out.println("Yaz");
        break;
    case "eylul":
    case "ekim":
    case "kasim":
        System.out.println("Guz");
        break;
    default:
        System.out.println("Gecersiz ay adi");
        break;
}
```

Switch-On Statement

- 2) Ayın numarasını girdiğinizde ayın adını yazdırmak için bir switch ifadesi kullanınız.
Örneğin;

kullanıcı 1 girerse kodunuz "Ocak" yazmalıdır,

kullanıcı 2 girerse kodunuz "Şubat" yazmalıdır vb.

Kullanıcı geçersiz ay numarası girerse kodunuz "Geçersiz numara" yazmalıdır.

```
int ayNo = 8;
String aylsim;
switch (ayNo) {
    case 1: aylsim = "Ocak";
        break;
    case 2: aylsim = "Subat";
        break;
    case 3: aylsim = "Mart";
        break;
    case 4: aylsim = "Nisan";
        break;
    case 5: aylsim = "Mayis";
        break;
    case 6: aylsim = "Haziran";
        break;
    case 7: aylsim = "Temmuz";
        break;
    case 8: aylsim = "Agustos";
        break;
    case 9: aylsim = "Eylul";
        break;
    case 10: aylsim = "Ekim";
        break;
    case 11: aylsim = "Kasim";
        break;
    case 12: aylsim = "Aralik";
        break;
    default: aylsim = "Gecersiz numara ";
        break;
}
System.out.println(aylsim);
```


Switch-On Statement

3) Verilen ay numaralarından başlayarak tüm ay adlarını yazdıran bir kod yazınız.

Örnek: Ay numarası 9 ise konsolda kodunuz Eylül Ekim Kasım Aralık olacaktır.

```
List<String> gelecekAylar = new ArrayList<>();
int ayNo = 8;
switch (ayNo) {
    case 1: futureMonths.add("Ocak");
    case 2: futureMonths.add("Subat");
    case 3: futureMonths.add("Mart");
    case 4: futureMonths.add("Nisan");
    case 5: futureMonths.add("Mayis");
    case 6: futureMonths.add("Haziran");
    case 7: futureMonths.add("Temmuz");
    case 8: futureMonths.add("Agustos");
    case 9: futureMonths.add("Eylul");
    case 10: futureMonths.add("Ekim");
    case 11: futureMonths.add("Kasim");
    case 12: futureMonths.add("Aralik");
        break;
    default: break;
}

if (gelecekAylar.isEmpty()) {
    System.out.println("Gecersiz ay numarası");
} else {
    for (String ayAdi : gelecekAylar) {
        System.out.print(ayAdi + " ");
    }
}
```

Switch-On Statement

- 4) Cinsiyet "Erkek" ise "Erkek" yazdırınız.
Cinsiyet "Kadın" ise "Kız" yazdırınız.
Cinsiyet "Erkek" ve "Kadın"dan farklıysa "Diğerleri" yazdırınız.
Not : Bu seçenekler dışındakilerini yoksayınız.

```
String cinsiyet = "Kadin";  
switch(cinsiyet.toLowerCase()){  
    case "kadin":  
        System.out.println("Kadin");  
        break;  
    case "erkek":  
        System.out.println("Erkek");  
        break;  
    default:  
        System.out.println("Digerleri");  
}
```

Switch-On Statement

- 5) Belirli bir yılın belirli bir ayındaki gün sayısını görüntülemek için bir kod yazınız.
Örnek: 2000 yılının Şubat ayında gün sayısı 29.

```
int ay = 2;
int yıl = 2000;
int gunSayisi = 0;

switch (ay) {
    case 1: case 3: case 5:
    case 7: case 8: case 10:
    case 12:
        gunSayisi = 31;
        break;
    case 4: case 6:
    case 9: case 11:
        gunSayisi = 30;
        break;
    case 2:
        if (((yıl % 4 == 0) && !(yıl % 100 == 0)) || (yıl % 400 == 0)) {
            gunSayisi = 29;
        } else {
            gunSayisi = 28;
        }
        break;
    default:
        System.out.println("Gecersiz Ay");
        break;
}
System.out.println("Gun Sayisi = " + gunSayisi);
```

Switch-On Statement

- 6) Switch ifadesini kullanarak, herhangi bir 2 sayıyla toplama, çıkarma, çarpma ve bölme yapan basit bir hesap makinesi oluşturmak için kod yazınız.
- a) Kullanıcı 10.2 ve 5 ve + işaretini girdiğinde kodunuz konsolda "Sonuç 15.2" yazmalıdır.
 - b) Kullanıcı 10 ve 5 ve - işaretini girdiğinde kodunuz konsolda "Sonuç 5.0" yazmalıdır.
 - c) Kullanıcı 10 ve 5.3 ve * işaretini girdiğinde kodunuz konsolda "Sonuç 53.0" yazmalıdır.
 - d) Kullanıcı 10 ve -5 ve / işaretini girdiğinde kodunuz konsolda "Sonuç -2.0" yazmalıdır.
 - e) Kullanıcı 10 ve -5 girdiğinde +, -, *, / 'den farklı bir işlem yaptığında kodunuz "Geçersiz İşlem" yazmalıdır.

```
double a = 10.2;
double b = 5;
String operator = "+";
switch(operator){
    case "+":
        System.out.println(a+b);
        break;
    case "-":
        System.out.println(a-b);
        break;
    case "*":
        System.out.println(a*b);
        break;
    case "/":
        System.out.println(a/b);
        break;
    default:
        System.out.println("Gecersiz islem");
}
```

Switch-On Statement

- 7) Switch ifadesini kullanarak, mili km'ye, saniyeyi saate, fahrenheitı santigrata çeviren bir dönüştürücü oluşturmak için kod yazınız. Google'dan dönüşümler için formülleri bulunuz.
- a) Kullanıcı mile değerini 10 ve operatorü mildenKmye olarak girdiğinde, konsolda kodunuz "16 km" yazmalıdır (sayı dinamik olacak)
 - b) Kullanıcı saniye değerini 7200 ve operatorü saniyedenSaate olarak girdiğinde, konsolda kodunuz "2" yazmalıdır (sayı dinamik olacak)
 - c) Kullanıcı fahrenheit değerini 83 ve operatorü fahrenheitdanSantigrata olarak girdiğinde, konsolda kodunuz "28.3333 santigrat" yazmalıdır (sayı dinamik olacak)
 - d) Kullanıcı mildenKmye, saniyedenSaate, fahrenheitdanSantigrata'dan farklı bir işlem girdiğinde, kodunuz "Bu operatör, o dönüştürücü için tanımlanmadı" yazmalıdır.

```
double mil = 10;
```

```
double saniye = 7200;
```

```
double fahrenheit = 83;
```

```
String operator = " fahrenheitdanSantigrata ";
```

```
switch(operator){
```

```
    case "mildenKmye":
```

```
        System.out.println(mil*1.60934);
```

```
        break;
```

```
    case "saniyedenSaate":
```

```
        System.out.println((saniye/60)/60);
```

```
        break;
```

```
    case " fahrenheitdanSantigrata":
```

```
        System.out.println((fahrenheit-32)*5/9);
```

```
        break;
```

```
    default:
```

```
        System.out.println("Bu operatör, o donusturucu icin tanimlanmadı");
```

```
}
```

Switch-On Statement

- 8) Bir Enum oluşturun ve içine CHROME, SAFARI, IE, FIREFOX, YANDEX gibi tarayıcı adlarını koyunuz. Switch ifadesini kullanın ve CHROME için 'CHROME kullanıyorum', SAFARI için 'SAFARI kullanıyorum', 'IE için IE kullanıyorum', 'FIREFOX için FIREFOX kullanıyorum' ve YANDEX için 'YANDEX kullanıyorum', CHROME, SAFARI, IE, FIREFOX, YANDEX'ten farklı tarayıcılar için 'Geçersiz tarayıcı' yazınız.

Not: Geliştirici konsolunda enum oluşturmalsınız.

```
public enum Browsers{
    CHROME, SAFARI, IE, FIREFOX, YANDEX
}

String browser = Browsers.CHROME.toString();
switch(browser){
    case "CHROME":
        System.out.println("CHROME kullanıyorum");
        break;
    case "SAFARI":
        System.out.println("SAFARI kullanıyorum");
        break;
    case "IE":
        System.out.println("IE kullanıyorum");
        break;
    case "FIREFOX":
        System.out.println("FIREFOX kullanıyorum");
        break;
    case "YANDEX":
        System.out.println("YANDEX kullanıyorum");
        break;
    default:
        System.out.println("Gecersiz tarayici");
}
```

Switch-On Statement

- 9) Adı 'Seasons' olan bir Enum oluşturun ve Enum da switch-on deyimini kullanarak aşağıdakileri yazdırınız.

Kış için 'Snowboard yapmak'

Yaz ve bahar için 'Balık tutmak'

Sonbahar için 'Doğa yürüyüşü yapmak'

```
public enum Seasons {  
    WINTER, SUMMER, FALL, SPRING  
}  
  
String browser = Seasons.WINTER.toString();  
switch(browser){  
    case "Kis":  
        System.out.println("Snowboard yapmak");  
        break;  
    case "Yaz":  
    case "Bahar":  
        System.out.println("Balik tutmak");  
        break;  
    case "Sonbahar":  
        System.out.println("Doga yuruyusu yapmak");  
        break;  
    default:  
        System.out.println("Gecersiz data");  
}
```

Switch-On Statement

10) myClass isminde bir class oluşturunuz, class'ta size rastgele bir alfabetik karakter veren bir yöntem oluşturun ve ardından yöntemi kullanarak aşağıdakileri yazdırınız.

'A' ve 'a' için "İlk Karakter" yazdırınız.

'B' ve 'b' için "İkinci Karakter" yazdırınız.

'C' ve 'c' için "Üçüncü Karakter" yazdırınız.

'D' ve 'd' için "Dördüncü Karakter" yazdırınız.

Diğerleri için "Diğer Karakterleri" yazdırınız.

```
public class myClass {  
    public static String getRandomAlphabet(){  
        String alfabe = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";  
        int maxIndex = alfabe.length()-1;  
        int randomIndex = (int)Math.round(Math.random()*maxIndex);  
        return alfabe.substring(randomIndex, randomIndex+1);  
    }  
}  
  
String alfabe = myClass.getRandomAlphabet().toLowerCase();  
switch(alfabe){  
    case "a":  
        System.out.println("İlk Karakter");  
        break;  
    case "b":  
        System.out.println("İkinci Karakter");  
        break;  
    case "c":  
        System.out.println("Üçüncü Karakter");  
        break;  
    case "d":  
        System.out.println("Dördüncü Karakter");  
        break;  
    default:  
        System.out.println("Diğer Karakterler");  
}
```


- 1) 120'den 11'e kadar 4 ile bölünebilen ve 6 ile bölünebilen tüm tam sayıları aynı satırda iki ardışık tam sayı arasında boşluk bırakarak yazınız.

1. Yol:

```
String s = "";
for(Integer i=120; i>10; i--){
    if(i%4==0 && i%6==0){
        s = s + i + " ";
    }
}
System.out.println(s);
```

2. Yol:

```
String s = "";
Integer i=120;
while(i>10){
    if(i%4==0 && i%6==0){
        s = s + i + " ";
    }
    i--;
}
System.out.println(s);
```

3. Yol:

```
String s = "";
Integer i=120;
do{
    if(i%4==0 && i%6==0){
        s = s + i + " ";
    }
    i--;
}while(i>10);
System.out.println(s);
```

2) Bir String' de tekrarlanan karakterleri yazdırmak için kod yazınız.

Örneğin; accessories ⇒ ces

1. Yol:

```
String s = "accessories";
String d = "";
for(Integer i=0; i<s.length(); i++){
    String c = s.substring(i,i+1);
    if(s.indexOf(c)!=s.lastIndexOf(c)){
        if(!d.contains(c)){
            d = d + c;
        }
    }
}
System.out.println(d);
```

2. Yol:

```
String s = "accessories";
String d = "";
Integer i=0;
while(i<s.length()){
    String c = s.substring(i,i+1);
    if(s.indexOf(c)!=s.lastIndexOf(c)){
        if(!d.contains(c)){
            d = d + c;
        }
    }
    i++;
}
System.out.println(d);
```

3. Yol:

```
String s = "accessories";
String d = "";
Integer i=0;
do{
    String c = s.substring(i,i+1);
    if(s.indexOf(c)!=s.lastIndexOf(c)){
        if(!d.contains(c)){
            d = d + c;
        }
    }
    i++;
}while(i<s.length());
System.out.println(d);
```

- 3) Bir String 'in palindrom olup olmadığını kontrol etmek için kod yazınız. Bir String, tersi ile aynıysa, buna palindrom denir. Örneğin; "anna", "123321" palindromlardır.

1. Yol:

```
String s = "runnur";
String r = "";
for(Integer i=s.length()-1; i>=0; i--){
    String c = s.substring(i,i+1);
    r = r + c;
}
if(s.equals(r)){
    System.out.println(s + " palindromdur");
}else{
    System.out.println(s + " palindrom degildir");
}
```

2. Yol:

```
String s = "runnur";
String r = "";
Integer i=s.length()-1;
while(i>=0){
    String c = s.substring(i,i+1);
    r = r + c;
    i--;
}
if(s.equals(r)){
    System.out.println(s + " palindromdur");
}else{
    System.out.println(s + " palindrom degildir");
}
```

3. Yol:

```
String s = "runnur";
String r = "";
Integer i=s.length()-1;
do{
    String c = s.substring(i,i+1);
    r = r + c;
    i--;
}while(i>=0);
if(s.equals(r)){
    System.out.println(s + " palindromdur");
}else{
    System.out.println(s + " palindrom degildir");
}
```

4) Benzersiz (Tekrarsız) rakamları bir tamsayı olarak yazdırmak için kod yazınız.

Örnek; 223878 \Rightarrow 37

1. Yol:

```
Integer num = 223878;
String s = String.valueOf(num);
String d = "";
for(Integer i=0; i<s.length(); i++){
    String c = s.substring(i,i+1);
    if(s.indexOf(c)==s.lastIndexOf(c)){
        d = d + c;
    }
}
System.out.println(d);
```

2. Yol:

```
Integer num = 223878;
String s = String.valueOf(num);
String d = "";
Integer i = 0;
while(i<s.length()){
    String c = s.substring(i,i+1);
    if(s.indexOf(c)==s.lastIndexOf(c)){
        d = d + c;
    }
    i++;
}
System.out.println(d);
```

3. Yol:

```
Integer num = 223878;
String s = String.valueOf(num);
String d = "";
Integer i = 0;
do{
    String c = s.substring(i,i+1);
    if(s.indexOf(c)==s.lastIndexOf(c)){
        d = d + c;
    }
    i++;
}while(i<s.length());
System.out.println(d);
```

Loops

5) Aşağıdaki görüntüyü bir for döngüsü kullanarak elde eden kodu yazınız.

A A A A A

A A A A A

A A A A A

1. Yol:

```
int rows = 3;
int columns = 5;
for(int i=1; i<=rows; i++){
    String s = "";
    for(Integer k=1; k<=columns; k++){
        s = s + "A ";
    }
    System.out.println(s);
}
```

2. Yol:

```
int rows = 3;
int columns = 5;
int i=1;
while(i<=rows){
    String s = "";
    for(Integer k=1; k<=columns; k++){
        s = s + "A ";
    }
    System.out.println(s);
    i++;
}
```

3. Yol:

```
int rows = 3;
int columns = 5;
int i=1;
do{
    String s = "";
    for(Integer k=1; k<=columns; k++){
        s = s + "A ";
    }
    System.out.println(s);
    i++;
}while(i<=rows);
```

6) Aşağıdaki görüntüyü bir for döngüsü kullanarak elde eden kodu yazınız.

A
A A
A A A
A A A A

1. Yol:

```
int rows = 4;
for(Integer i=1; i<=rows; i++){
    String s = "";
    for(Integer k=1; k<=i; k++){
        s = s + "A ";
    }
    System.out.println(s);
}
```

2. Yol:

```
int rows = 4;
int i=1;
while(i<=rows){
    String s = "";
    for(int k=1; k<=i; k++){
        s = s + "A ";
    }
    System.out.println(s);
    i++;
}
```

3. Yol:

```
int rows = 4;
int i=1;
do{
    String s = "";
    for(int k=1; k<=i; k++){
        s = s + "A ";
    }
    System.out.println(s);
    i++;
}while(i<=rows);
```

7) Aşağıdaki çarpım tablosunu oluşturmak için kod yazınız.

3x1=3 3x2=6 3x3=9 3x4=12 3x5=15 3x6=18 3x7=21 3x8=24 3x9=27 3x10=30

1. Yol:

```
int num = 3;
for(int i=1; i<11; i++){
    System.out.println(num + "x" + i + "=" + num*i);
}
```

2. Yol:

```
int num = 3;
int i=1;
while(i<11){
    System.out.println(num + "x" + i + "=" + num*i);
    i++;
}
```

3. Yol:

```
int num = 3;
int i=1;
do{
    System.out.println(num + "x" + i + "=" + num*i);
    i++;
}while(i<11);
```

- 8) 20'den 3'e kadar olan tek tamsayıları aynı satırda ardışık olacak şekilde aralarında boşluk bırakarak yazdırmak için gereken kodu yazınız.

1. Yol:

```
String s = "";
for(Integer i=20; i>2; i--){
    if(i%2!=0){
        s = s + i + " ";
    }
}
System.out.println(s);
```

2. Yol:

```
String s = "";
int i=20;
while(i>2) {
    if(i%2!=0){
        s = s + i + " ";
    }
    i--;
}
System.out.println(s);
```

3. Yol:

```
String s = "";
int i=20;
do{
    if(i%2!=0){
        s = s + i + " ";
    }
    i--;
}while(i>2);
System.out.println(s);
```



9) String içindeki tüm küçük harfleri yıldız işaretiyle yazdırmak için gereken kodu yazınız.

Örneğin; 'Ali Can?' ==> l*i*a*n*

1. Yol:

```
String s = "Ali Can?";  
s = s.replaceAll("[a-z]", "");  
String t = "";  
for(Integer i=0; i<s.length(); i++){  
    String r = s.substring(i,i+1);  
    t = t + r + '*';  
}  
System.out.println(t);
```

2. Yol:

```
String s = "Ali Can?";  
s = s.replaceAll("[a-z]", "");  
String t = "";  
Integer i=0;  
while(i<s.length()){  
    String r = s.substring(i,i+1);  
    t = t + r + '*';  
    i++;  
}  
System.out.println(t);
```

3. Yol:

```
String s = "Ali Can?";  
s = s.replaceAll("[a-z]", "");  
String t = "";  
Integer i=0;  
do{  
    String r = s.substring(i,i+1);  
    t = t + r + '*';  
    i++;  
}while(i<s.length());  
System.out.println(t);
```

- 10) Verilen ondalık sayının sadece ondalık kısmındaki rakamlarını, kendisinden önce bir yıldız işaretiyle yazdırmak için gereken kodu yazınız.**

Örneğin; 75.4238 \Rightarrow *4*2*3*8

1. Yol:

```
double num = 75.4238;
String s = String.valueOf(num);
int idxOfComma = s.indexOf(",");
String decimalPart = s.substring(idxOfComma + 1);
String t = "";
for(Integer i=0; i<decimalPart.length(); i++){
    String r = decimalPart.substring(i,i+1);
    t = t + "*" + r;
}
System.out.println(t);
```

2. Yol:

```
double num = 75.4238;
String s = String.valueOf(num);
int idxOfComma = s.indexOf(",");
String decimalPart = s.substring(idxOfComma + 1);
String t = "";
Integer i=0;
while(i<decimalPart.length()){
    String r = decimalPart.substring(i,i+1);
    t = t + '*' + r;
    i++;
}
System.out.println(t);
```

3. Yol:

```
double num = 75.4238;
String s = String.valueOf(num);
int idxOfComma = s.indexOf(",");
String decimalPart = s.substring(idxOfComma + 1);
String t = "";
Integer i=0;
do{
    String r = decimalPart.substring(i,i+1);
    t = t + '*' + r;
    i++;
}while(i<decimalPart.length());
System.out.println(t);
```

11) Bir String'i tersine çevirmek için kod yazınız. Örnek; Mark ==> kraM

1. Yol:

```
String t = "Mark";
String rev = "";
for(Integer i=t.length()-1; i>-1; i--){
    rev = rev + t.substring(i,i+1);
}
System.out.println(rev);
```

2. Yol:

```
String t = "Mark";
StringBuilder sb = new StringBuilder(t);
System.out.println(sb.reverse());
```

12) Aşağıdaki görüntüyü bir for döngüsü kullanarak elde eden kodu yazınız.

```
A A A A A A A
A X X X X X A
A X X X X X A
A X X X X X A
```

1. Yol:

```
int rows = 4, columns = 8;
for(int i=1; i<=rows; i++){
    String s = "";
    if(i==1 || i==rows){
        for(Integer k=1; k<=columns; k++){
            s = s + "A ";
        }
        System.out.println(s);
    }else{
        s = s + "A ";
        for(Integer m=2; m<columns; m++){
            s = s + "X ";
        }
        s = s + "A ";
        System.out.println(s);
    }
}
```

Loops

2. Yol:

```
int rows = 4, columns = 8, i = 1;
while(i<=rows){
    String s = "";
    if(i==1 || i==rows){
        for(Integer k=1; k<=columns; k++){
            s = s + "A ";
        }
        System.out.println(s);
    }else{
        s = s + "A ";
        for(Integer m=2; m<columns; m++){
            s = s + "X ";
        }
        s = s + "A ";
        System.out.println(s);
    }
    i++;
}
```

3. Yol:

```
int rows = 4, columns = 8, i = 1;
do{
    String s = "";
    if(i==1 || i==rows){
        for(Integer k=1; k<=columns; k++){
            s = s + "A ";
        }
        System.out.println(s);
    }else{
        s = s + "A ";
        for(Integer m=2; m<columns; m++){
            s = s + "X ";
        }
        s = s + "A ";
        System.out.println(s);
    }
    i++;
}while(i<=rows);
```

13) 3 ile 14 arasındaki tam sayıların toplamını bulmak için kodu yazınız.

1. Yol:

```
int sum = 0;
for(int i=3; i<15; i++){
    sum = sum+i;
}
System.out.println(sum);
```

2. Yol:

```
int sum = 0, i = 3;
while(i<15){
    sum = sum+i;
    i++;
}
System.out.println(sum);
```

3. Yol:

```
int sum = 0, i = 3;
do{
    sum = sum+i;
    i++;
}while(i<15);
System.out.println(sum);
```



14) 3 ile 9 arasındaki tam sayıların çarpımını bulmak için kodu yazınız.

1. Yol:

```
int m = 1;
for(int i=3; i<10; i++){
    m = m*i;
}
System.out.println(m);
```

2. Yol:

```
int m = 1, i = 3;
while(i<10){
    m = m*i;
    i++;
}
System.out.println(m);
```

3. Yol:

```
int m = 1, i = 3;
do{
    m = m*i;
    i++;
}while(i<10);
System.out.println(m);
```



15) Do-while döngüsünü kullanarak konsolda 'C' ile 'A' arasındaki karakterleri yazdırmak için kod yazınız.

1. Yol:

```
String result = "";
for(char i = 'C'; i>='A'; i--){
    result = result + i;
}
System.out.println(result);
```

2. Yol:

```
char ch = 'C';
String result = "";
while(ch>='A'){
    result = result + ch;
    ch--;
}
System.out.println(result);
```

3. Yol:

```
char ch = 'C';
String result = "";
do{
    result = result + ch;
    ch--;
}while(ch>='A');
System.out.println(result);
```

16) Bir String de, ilk 'm' karakterinden önceki tüm karakterleri yazınız.

1. Yol:

```
String s = "Christmas";
String result = "";
for(int i=0; i<s.length(); i++){
    String c = s.substring(i,i+1);
    if(c.equals("m")){
        break;
    }
    result = result + c;
}
System.out.println(result);
```

2. Yol:

```
String s = "Christmas";
String result = "";
int i=0;
while( i<s.length()){
    String c = s.substring(i,i+1);
    if(c.equals("m")){
        break;
    }
    result = result + c;;
    i++;
}
System.out.println(result);
```

3. Yol:

```
String s = "Christmas";
String result = "";
int i=0;
do{
    String c = s.substring(i,i+1);
    if(c.equals("m")){
        break;
    }
    result = result + c;;
    i++;
}while( i<s.length());
System.out.println(result);
```


17) Bir tamsayıdaki rakamların toplamını bulmak için kodu yazınız.

1. Yol:

```
int n = 753;
int sumOfDigits = 0;
for(int i = n; i>0; i=i/10){
    sumOfDigits = sumOfDigits + i%10;
}
System.out.println(sumOfDigits);
```

2. Yol:

```
int num = 753;
int sumOfDigits = 0;
while(num>0){
    sumOfDigits = sumOfDigits + num%10;
    num = num/10;
}
System.out.println(sumOfDigits);
```

3. Yol:

```
int num = 753;
int sumOfDigits = 0;
do{
    sumOfDigits = sumOfDigits + num%10;
    num = num/10;
}while(num>0);
System.out.println(sumOfDigits);
```

18) Bir String' de benzersiz(tekrarsız) karakterler yazdırmak için kod yazınız.

Örneğin; Hello ==> Heo

1. Yol:

```
String s = "Hello";
for(int i=0; i<s.length();i++){
    String c = s.substring(i,i+1);
    if(s.indexOf(c) == s.lastIndexOf(c)){
        System.out.print(c);
    }
}
```

2. Yol:

```
String s = "Hello";
int i = 0;
while(i<s.length()){
    String c = s.substring(i,i+1);
    if(s.indexOf(c) == s.lastIndexOf(c)){
        System.out.print(c);
    }
    i++;
}
```

3. Yol:

```
String s = "Hello";
int i = 0;
do{
    String c = s.substring(i,i+1);
    if(s.indexOf(c) == s.lastIndexOf(c)){
        System.out.print(c);
    }
    i++;
}while(i<s.length());
```

Loops

- 19) Bir String’deki boşluk ve noktalama işaretleri dışındaki toplam karakter sayısını bulunuz.**

```
String s = "Java is a strongly typed, object-oriented programming language.";
s = s.replaceAll("\\s","").replaceAll("\\p{Punct}", "");
System.out.println(s.length());
```

- 20) Aşağıdaki görüntüyü bir for döngüsü kullanarak elde eden kodu yazınız.**

```

* * * * *
* * * * *
* * * *
* * *
* *
*
*

int sayi=6;
String s = "";
for(int i=1; i<= sayi; i++){
    for(int k= sayi; k>=i; k--){
        s = s + "* ";
    }
    System.out.println(s);
    s = "";
}
```

- 21) Bir tamsayının benzersiz(tekrarsız) basamaklarının toplamını bulmak için kodu yazınız.**

Örnek:12133455 \Rightarrow 2+4=6

```
int p = 12133455;
String r = String.valueOf(p);
int sumOfUniqueDigits = 0;
for(int i=0; i<r.length(); i++){
    String c = r.substring(i,i+1);
    if(r.indexOf(c)==r.lastIndexOf(c)){
        sumOfUniqueDigits = sumOfUniqueDigits + Integer.valueOf(c);
    }
}
System.out.println(sumOfUniqueDigits);
```

22) 5 hariç 3'ten 9'a kadar olan tam sayıları yazdırmak için kod yazınız.

```
for(Integer i=3; i<10; i++){  
    if(i==5){  
        continue;  
    }  
    System.out.print(i + " ");  
}
```

23) Asal sayı olup olmadığını kontrol etmek için kullanıcıdan bir tam sayı girmesini isteyiniz. Örnek: kullanıcı 43 girerse çıktı "43 bir asal sayıdır", kullanıcı 120 girerse "120 asal değildir" olur, kullanıcı negatif tam sayılar girerse çıktı "Pozitif bir tam sayı giriniz" şeklinde kullanıcıyı yönlendiriniz.

```
Scanner scan = new Scanner(System.in);  
System.out.println("Asal olup olmadigini kontrol etmek icin pozitif bir tamsayi giriniz: ");  
int number = scan.nextInt();  
int count = 0;  
if(number>0) {  
    if (number == 1) {  
        System.out.println(number + " bir asal sayidir");  
    }else {  
        for(int i=2; i<number; i++) {  
            if(number%i==0) {  
                count++;  
            }  
        }  
        if(count==0) {  
            System.out.println(number + " bir asal sayidir ");  
        }else {  
            System.out.println(number + " bir asal sayi degildir ");  
        }  
    }  
}else {  
    System.out.println("Pozitif bir tam sayi giriniz ");  
}
```

- 24) Kullanıcıya kaç tane fibonacci sayısı görmek istediğini sorunuz. Ardından bu fibonacci sayılarını yazdıran kodu yazınız.**

Örnek: kullanıcı 6 girerse çıktı 1 1 2 3 5 8 olacaktır.

```
Scanner scan = new Scanner(System.in);
System.out.println("Kac tane fibonacci sayisi gormek istiyorsunuz ?");
int number = scan.nextInt();
int fibo1 = 1;
int fibo2 = 1;
int fibonacci = 0;
System.out.print(fibo1 + " ");
System.out.print(fibo2 + " ");
for (int i = 1; i < number-1; i++) {
    fibonacci = fibo1 + fibo2;
    fibo1 = fibo2;
    fibo2 = fibonacci;
    System.out.print(fibonacci + " ");
}
```

- 25) Bir sayı, her basamağının küpünün toplamına eşitse Armstrong sayısı olarak adlandırılır. Örneğin, 153, her basamağın küpüne eşit olan $153 = 1 + 125 + 27$ olduğundan bir Armstrong sayısıdır. Verilen numaranın Armstrong sayısı olup olmadığını kontrol etmek için bir kod yazınız.**

```
Scanner scan = new Scanner(System.in);
System.out.println("Armstrong sayisi olup olmadigini kontrol etmek icin bir sayi giriniz :");
int n = scan.nextInt();
int sumOfCubes = 0;
int temp = n;
int digit = 0;
while(n>0) {
    digit = n%10;
    sumOfCubes = sumOfCubes + digit*digit*digit;
    n = n/10;
}
if(temp==sumOfCubes) {
    System.out.println(temp + " Armstrong sayidir");
}else {
    System.out.println(temp + " Armstrong sayi degildir");
}
```

26) Kullanıcıdan bir String ve bir karakter alınız.

String'de verilen karakterin ilk olduğu index ile son tekrarlandığı index arasındaki karakter sayısını sayınız.

Boşluk karakterlerini saymayınız.

Kod, kullanıcının seçtiği karakter String 'de sadece bir kez varsa konsolda -1,

Kod, kullanıcının seçtiği karakter String 'de yoksa konsolda -1 döndürsün.

Örneğin; "Java is easy" - 'a' ==> 5

"Java is easy" - 'w' ==> -1

"Java is easy" - 'e' ==> -1

```
Scanner scan = new Scanner(System.in);
```

```
System.out.println("Bir String giriniz :");
```

```
String s = scan.nextLine();
```

```
System.out.println("Bir karakter giriniz :");
```

```
char ch = scan.next().charAt(0);
```

```
int firstOcclDx = s.indexOf(ch);
```

```
int lastOcclDx = s.lastIndexOf(ch);
```

```
int counter = 0;
```

```
if(firstOcclDx == lastOcclDx) {
```

```
    System.out.println(-1);
```

```
}else {
```

```
    for(int i=firstOcclDx+1; i<lastOcclDx; i++) {
```

```
        if(s.charAt(i)!=' '){
```

```
            counter++;
```

```
        }
```

```
    }
```

```
    System.out.println(counter);
```

```
}
```

27) Belirli bir arrayden, toplamı belirli bir sayı olan tüm çiftleri bulunuz.

Array {4, 6, 5, -10, 8, 5, 20} ve sayı 10 ise, çıktı 4+6=10, 5+5=10, -10+20=10 olacaktır.

```
Integer arr[] = {4, 6, 5, -10, 8, 5, 20};
int num = 10;
for(int i=0; i<arr.length; i++) {
    for(int j=i+1; j<arr.length; j++) {
        if(arr[i]+arr[j]==num) {
            System.out.println(arr[i] + " + " + arr[j] + " = " + num);
        }
    }
}
```

28) Belirli bir sayıdan küçük tüm asal sayıları yazdıran bir kod yazınız.

Örnek: kullanıcı 20 girerse çıktı 2, 3, 5, 7, 11, 13, 17, 19 olacaktır.

```
public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Pozitif bir sayı giriniz:");
    int n = scan.nextInt();
    printPrime(n);
}

private static boolean isPrime(int n){
    if (n <= 1) {
        return false;
    }
    for (int i = 2; i < n; i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}

public static void printPrime(int n){
    for (int i = 2; i <= n; i++){
        if(isPrime(i)) {
            System.out.print(i + " ");
        }
    }
}
```

- 29) Herhangi bir döngü kullanmadan 1'den 100'e kadar tam sayıları yazdıran bir program yazınız.**

```
public static void main(String[] args) {  
    printNumbers(100);  
}  
static void printNumbers(int n) {  
    if(n > 0) {  
        printNumbers(n - 1);  
        System.out.print(n + " ");  
    }  
    return;  
}
```

- 30) Her girdi satırı, tek boşluklarla ayrılmış bir veya birkaç sözcük içerir. Sözcükler değişmeden, her giriş satırındaki sözcüklerin sırasını tersine çevirmek için kod yazınız.**

Çıktınızın satırlarında sonunda veya başında boşluk olmamalıdır.

Örnek; Kemal Can Kuzu, Kuzu Can Kemal'e dönüşecektir.

```
public static void main(String[] args) {  
    String input = "Ali\n" +  
        "Veli Han\n" +  
        "Kemal Can Kuzu";  
    String rev = "";  
    String[] lines = input.split(System.getProperty("line.separator"));  
    for(String w : lines) {  
        String[] words = w.split(" ");  
        for (int i = words.length - 1; i >= 0; i--) {  
            if (i != 0) {  
                rev = rev + words[i] + " ";  
            } else {  
                rev = rev + words[i];  
            }  
        }  
        System.out.println(rev);  
        rev="";  
    }  
}
```


31) Her girdi satırı, tek boşluklarla ayrılmış bir veya birkaç sözcük içerir. Her giriş satırındaki ilk ve son sözcükler dışındaki sözcüklerin sırasını tersine çevirip sözcükleri değiştirmeden kodu yazınız.

Çıktınızın satırlarında sonunda veya başında boşluk olmamalıdır.

Örnek; Kemal Can Tan Han Kuzu, Kemal Han Tan Can Kuzu'ya dönüşecektir.

```
String input = "Ali\n" +  
    "Veli Can Han\n" +  
    "Kemal Can Tan Kuzu\n" +  
    "Kemal Can Tan Han Kuzu\n" +  
    "Kemal Can Tan Han Man Kuzu";  
  
String[] lines = input.split(System.getProperty("line.separator"));  
for(String w : lines) {  
    String[] words = w.split(" ");  
    String rev = words[0] + " ";  
    for (int i = words.length - 2; i >= 1; i--) {  
        rev = rev + words[i] + " ";  
    }  
    rev = rev + words[words.length-1];  
    System.out.println(rev);  
    rev="";  
}
```

1) Tamsayılardan oluşan bir arrayde ortadaki elemanı bulunuz.

Örnek: (12, 5, 8) ==> (5, 8, 12) ==> Çıktı=8

(12, 5, 8, 13) ==> (5, 8, 12, 13) ==> Çıktı=(8+12)/2 = 10

```
int[] a = new int[] {12, 5, 8, 13};
```

```
Arrays.sort(a);
```

```
if(a.length%2!=0){
```

```
    Integer ortadakiElemaninIndexi = a.length/2;
```

```
    System.out.println(a[ortadakiElemaninIndexi]);
```

```
}else{
```

```
    Integer ortadakiElemaninIndexi = a.length/2;
```

```
    Integer ortadakiEleman = (a[ortadakiElemaninIndexi] + a[ortadakiElemaninIndexi - 1])/2;
```

```
    System.out.println(ortadakiEleman);
```

```
}
```

2) String'lerden oluşan bir arrayde uzunluğu en küçük olan elemanları bulunuz.

Örnek: (Kemal, Jonathan, Mark, Angie, Veli) ==> Output is Mark, Veli

```
String[] b = new String[] {"Kemal", "Jonathan", "Mark", "Angie", "Veli"};
```

```
int minLength = b[0].length();
```

```
for(String w : b){
```

```
    minLength = Math.min(minLength, w.length());
```

```
}
```

```
for(String w : b){
```

```
    if(minLength == w.length()){
```

```
        System.out.println(w);
```

```
    }
```

```
}
```

3) Tamsayılardan oluşan bir arrayde en küçük pozitif elemanı ve en büyük negatif elemanı bulunuz.

Örnek: (-12, 18, -5, 23, -2) ==> En küçük pozitif 18, en büyük negatif -2

1. Yol:

```
int[] a = new int[]{-12, 18, -5, 23, -2};
Arrays.sort(a);
Integer minPositive = a[a.length-1];
Integer maxNegative = a[0];
for(Integer w : a){
    if(w>=0){
        minPositive = Math.min(minPositive, w);
    }
    if(w<0){
        maxNegative = Math.max(maxNegative, w);
    }
}
System.out.println("Minimum positive: " + minPositive);
System.out.println("Maximum negative: " + maxNegative);
```

2. Yol:

```
int[] a = new int[]{-12, 18, -5, 23, -2};
Arrays.sort(a);
for (Integer i=0;i<a.length;i++){
    if (a[i]<0 && a[i+1]>0) {
        System.out.println("Maximum negative: " + a[i]);
        System.out.println("Minimum positive: " + a[i+1]);
    }
}
```

- 4) String'lerden oluşan bir arrayde öğeler 'n' veya 'k' ile bitiyorsa, o öğelerin baş harflerini alınız.

Örnek: { "Kemal", "Jonathan", "Mark", "Jackson", "Ali" } ==> Çıktı JJM

```
String[] b = new String[]{"Kemal", "Jonathan", "Mark", "Angie", "Veli"};  
String initials = "";
```

```
for(String w : b){  
    if(w.endsWith("n") || w.endsWith("k")){  
        initials = initials + w.substring(0,1);  
    }  
}  
System.out.println(initials);
```

- 5) String'lerden oluşan bir arrayde kullanılan öğelerin toplam karakter sayısını bulunuz.

Örnek: { "Kemal", "Jonathan", "Mark", "Angie", "Veli" } ==> Çıktı 26

```
String[] b = new String[]{"Kemal", "Jonathan", "Mark", "Angie", "Veli"};  
int sum = 0;
```

```
for(String w : b){  
    sum = sum + w.length();  
}  
System.out.println("Total number of characters is " + sum);
```

6) Verilen bir String'de 'a' veya 'A' ile başlayan kelimeyi sayısını bulunuz.

```
String s = "Apex is an object oriented programming language";
String[] arr = s.split(" ");
int counter = 0;
for(String w : arr){
    if(w.startsWith("a") || w.startsWith("A")){
        counter++;
    }
}
System.out.println("The number of words which starts with a or A is : " + counter);
```

7) Verilen bir String'deki sesli harf sayısını bulunuz.

```
String s = "Apex is an object oriented programming language";
String[] arr = s.toLowerCase().split("");
int counter = 0;
for(String w : arr){
    switch(w){
        case "a":
        case "e":
        case "i":
        case "o":
        case "u":
            counter++;
    }
}
System.out.println("The number of vowels is " + counter);
```

8) İlk ve son karakterleri aynı olan array öğelerini bulmak için kod yazınız.

```
String[] arr = new String[]{"alabama", "pick", "sos", "sets", "pex"};
int size = arr.length;
for(Integer i=0; i<size; i++){
    String firstChar = arr[i].substring(0, 1);
    String lastChar = arr[i].substring(arr[i].length()-1);
    if(firstChar.equals(lastChar)){
        System.out.println(arr[i]);
    }
}
```

9) Verilen bir String arraydeki belirli bir öğenin var olup olmadığını bulmak için kod yazınız.

```
String[] arr = new String[]{"Apex", "is", "an", "object", "oriented", "programming", "language"};
String s = "object";
int counter = 0;
for(String w : arr){
    if(w.equalsIgnoreCase(s)){
        counter++;
    }
}
if(counter>0){
    System.out.println(s + " arrayde vardır");
}else{
    System.out.println(s + " arrayde yoktur");
}
```

10) Verilen bir String arraydeki öğelerin karakterlerinin toplamını bulmak için kod yazınız.

```
String[] arr = new String[]{"alabama", "pick", "sos", "sets", "pex"};
Integer sum = 0;
for(String w : arr){
    sum = sum + w.length();
}
System.out.println(sum);
```

11) Tamsayılardan oluşan arrayde bulunan sıfırları, array sonuna yerleştiren kod yazınız.

Örnek: (5, 0, 2, 0, 3) ==> (5, 2, 3, 0, 0)

```
Integer[] arr = new Integer[]{5, 0, 2, 0, 3};
Integer[] brr = new Integer[arr.length];
int firstIdx = 0;
int lastIdx = arr.length-1;
```

```
for(int i=0; i<arr.length; i++){
    if(arr[i]!=0){
        brr[firstIdx] = arr[i];
        firstIdx++;
    }else{
        brr[lastIdx]=0;
        lastIdx--;
    }
}
```

```
System.out.println(Arrays.toString(brr));
```

- 12)** Kullanıcıdan aldığınız tamsayılar ile bir array oluşturunuz ve bu arraydeki en küçük ve en büyük öğeler arasındaki farkı konsolda yazdırınız.

```
Scanner scan = new Scanner(System.in);
System.out.println("Array uzunlugunu giriniz : ");
int len = scan.nextInt();

int arr[] = new int[len];

System.out.println("Girilen array elemanlari : " + len);
for(int i=0; i<len; i++) {
    arr[i] = scan.nextInt();
}
System.out.println(Arrays.toString(arr));

Arrays.sort(arr);
System.out.println(Arrays.toString(arr));

int diffOfBigAndSmall = arr[arr.length-1] - arr[0];
System.out.println("En buyuk ve en kucuk oge arasındaki fark : " + diffOfBigAndSmall);
```


- 13) Kullanıcıdan 2 String girmesini isteyiniz. Stringlerin karakterleri ve karakter sayıları aynıysa konsola "Anagramdır" yazdırın. Aksi takdirde, konsolda "Anagram Değil" yazdırınız.**

Örneğin; "Mary" ve "army" ve "RAMY" Anagramlardır.

```
Scanner scan = new Scanner(System.in);
System.out.println("iki String giriniz :");
String s1 = scan.nextLine();
String s2 = scan.nextLine();
```

```
String a1[] = s1.toLowerCase().split("");
Arrays.sort(a1);
```

```
String a2[] = s2.toLowerCase().split("");
Arrays.sort(a2);
```

```
if (s1.length() != s2.length()) {
    System.out.println("Anagram Degil");
} else if (s1.isEmpty() || s2.isEmpty()) {
    System.out.println("Anagram Degil");
} else if (Arrays.equals(a1, a2)) {
    System.out.println("Anagramdir");
} else {
    System.out.println("Anagram Degil");
}
```

1) Bir tamsayı listesindeki tüm öğelerinin toplamını bulunuz.

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(31);
myList.add(7);
myList.add(13);
myList.add(10);
int sum = 0;
for(Integer w : myList){
    sum = sum + w;
}
System.out.println(sum);
```

2) Bir tamsayı listesinde 13 ögesinden önceki tüm liste öğelerinin toplamını bulunuz.

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(31);
myList.add(7);
myList.add(13);
myList.add(10);
int sum = 0;
for(Integer w : myList){
    if(w==13){
        break;
    }
    sum = sum + w;
}
System.out.println(sum);
```

3) Bir tamsayı listesindeki tüm çift sayıların çarpımını bulunuz.

1. Yol:

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(31);
myList.add(7);
myList.add(13);
myList.add(10);
int m = 1;
for(Integer w : myList){
    if(w%2!=0){
        continue;
    }
    m = m * w;
}
System.out.println(m);
```

2. Yol:

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(31);
myList.add(7);
myList.add(13);
myList.add(10);
int m = 1;
for(Integer w : myList){
    if(w%2==0){
        m = m * w;
    }
}
System.out.println(m);
```

4) Bir listedeki öğelerin azalan sırada olup olmadığını kontrol ediniz.

Örnek: (Sarı, Mavi, Kırmızı, Yeşil) ==> Çıktı: Azalan sırada değil

(Sarı, Kırmızı, Yeşil, Mavi) ==> Çıktı: Azalan sıradadır

```
List<String> e = new ArrayList<>();
e.add("Yellow");
e.add("Red");
e.add("Green");
e.add("Blue");
List<String> f = new ArrayList<>();
f.addAll(e);
Collections.sort(f);
System.out.println(e);
System.out.println(f);
int size = e.size();
int flag = 0;
for(int i=0; i<size; i++){
    if(f.get(i).equals(e.get(size-1-i))){
        flag++;
    }
}
if(flag == size){
    System.out.println("Azalan sıradadır");
}else{
    System.out.println("Azalan sırada değil");
}
```

5) Listede eleman olarak 15 varsa, tüm 15'leri 51 olarak değiştiriniz.

Örnek: (12, 11, 15, 34, 43) ==> Çıktı (12, 11, 51, 34, 43)

```
List<Integer> g = new ArrayList<>();
g.add(12);
g.add(11);
g.add(15);
g.add(34);
g.add(15);
g.add(43);
if(g.contains(15)){
    for(int w : g){
        if(w==15){
            int idx = g.indexOf(15);
            g.set(idx, 51);
        }
    }
    System.out.println(g);
}else{
    System.out.println("liste 15 elemanıni icermiyor");
}
```

6) Listede 15 veya 13 varsa, bu elemanları kaldırınız.

Örnek: (10, 31, 15, 13, 54) ==> Çıktı (10, 31, 54)

```
List<Integer> h = new ArrayList<>();
h.add(10);
h.add(31);
h.add(15);
h.add(13);
h.add(54);
h.add(13);
if(!h.contains(15) && !h.contains(13)){
    System.out.println("Liste 13 ve 15'i icermiyor");
}else {
    for (int i = 0; i < h.size(); i++) {
        if (h.get(i) == 15) {
            int idx15 = h.indexOf(15);
            h.remove(idx15);
            i--;
        }
        if (h.get(i) == 13) {
            int idx13 = h.indexOf(13);
            h.remove(idx13);
            i--;
        }
    }
    System.out.println(h);
}
```

7) Bir tamsayı listesinde en yakın 2 tamsayıyı bulunuz.

Örnek: (12, 31, 15, 13, 54) ==> Çıktı 12 ve 13'tür

```
List<Integer> h = new ArrayList<>();
h.add(12);
h.add(31);
h.add(15);
h.add(13);
h.add(54);
Collections.sort(h);
int minDiff = h.get(2) - h.get(1);
for(int i=1; i<h.size(); i++){
    minDiff = Math.min(minDiff, h.get(i)-h.get(i-1));
}
for(Integer i=1; i<h.size(); i++){
    if(h.get(i)-h.get(i-1) == minDiff){
        System.out.println(h.get(i) + " and " + h.get(i-1));
    }
}
```

8) Listedeki 7 ve 10 dışındaki her öğenin değerini 2'ser artırınız.

Örnek: (12, 31, 7, 13, 10) ==> Çıktı (14, 33, 7, 15, 10)

```
List<Integer> h = new ArrayList<>();
h.add(12);
h.add(31);
h.add(7);
h.add(13);
h.add(10);
for(Integer w : h){
    if(w==7 || w==10){
        continue;
    }
    h.set(h.indexOf(w), w+2);
}
System.out.println(h);
```

9) String bir listede verilen tüm fiyatların toplamını bulunuz.

Örnek: `List<String> myList = new List<String>{'$12.99', '$23.60', '$54.45'}; ==> 91.04`

```
List<String> myList = new ArrayList<>();
myList.add("$12.99");
myList.add("$23.60");
myList.add("$54.45");
double sum = 0;
for(String w : myList){
    Double price = Double.valueOf(w.replace("$", ""));
    sum = sum + price;
}
System.out.println(sum);
```

10) Bir String listesinde verilen en düşük ve en büyük fiyatın toplamını bulunuz.

Örnek: `List<String> myList = new List<String>{'$12.99', '$8.25', '$23.60', '$54.45'}; ==> 62.70`

```
List<String> myList = new ArrayList<>();
myList.add("$12.99");
myList.add("$23.60");
myList.add("$8.25");
myList.add("$54.45");
List<Double> decimalList = new ArrayList<>();
for(String w : myList){
    Double price = Double.valueOf(w.replace("$", ""));
    decimalList.add(price);
}
Collections.sort(decimalList);
Double sum = decimalList.get(0) + decimalList.get(decimalList.size()-1);
System.out.println(sum);
```


- 11) Döngüleri kullanarak tamsayılardan oluşan bir listenin tüm öğelerinin benzersiz (tekrarsız) olup olmadığını kontrol ediniz.**

```
List<Integer> myList = new ArrayList<>();
myList.add(10);
myList.add(31);
myList.add(15);
myList.add(7);
myList.add(15);
myList.add(23);
Integer counter = 0;
for(Integer w : myList){
    for(Integer k : myList){
        if(w==k){
            counter++;
        }
    }
}
if(counter==myList.size()){
    System.out.println("Tekrarlanan oge yoktur");
}else{
    System.out.println("En az 1 oge tekrarlanmistir");
}
```

- 12) Verilen herhangi bir arrayin Mountain Array olup olmadığını kontrol etmek için bir kod yazınız.**

Not:

Mountain Array ==> [0, 2, 5, 3, 1]

Bir array elemanları en büyük değerine kadar sürekli artan, en büyük değerinden sonra sürekli azalan değer alıyorsa Mountain Array' dir.

Mountain Array Değil ==> [5, 2, 7, 1, 4]

Bir array elemanları en büyük değerine kadar sürekli azalan, en büyük değerinden sonra sürekli artan değer alıyorsa Mountain Array değildir.

```
int a[] = { 0, 2, 5, 3, 1 };
List<Integer> list = new ArrayList<>();
Arrays.stream(a).forEach(t -> list.add(t));
int max = list.stream().reduce(Integer.MIN_VALUE, Integer::max);
int idxOfMax = list.indexOf(max);
List<Integer> part1 = new ArrayList<>();
list.stream().filter(t -> list.indexOf(t) <= idxOfMax).forEach(t -> part1.add(t));
List<Integer> sortedPart1 = new ArrayList<>();
part1.stream().forEach(t -> sortedPart1.add(t));
Collections.sort(sortedPart1);
List<Integer> part2 = new ArrayList<>();
list.stream().filter(t -> list.indexOf(t) >= idxOfMax).forEach(t -> part2.add(t));
List<Integer> reverseSortedPart2 = new ArrayList<>();
part2.stream().forEach(t -> reverseSortedPart2.add(t));
Collections.sort(reverseSortedPart2);
Collections.reverse(reverseSortedPart2);

if (part1.equals(sortedPart1) && part2.equals(reverseSortedPart2)) {
    System.out.println(list + " Mountain Arraydir");
} else {
    System.out.println(list + " Mountain Array degildir");
}
```

1) Bir listede kaç farklı öğe olduğunu gösteren kodu yazınız.

Örnek: {10, 31, 15, 7, 15, 7, 7} ==> 4

```
List<Integer> myList = new ArrayList<>();  
myList.add(10);  
myList.add(31);  
myList.add(15);  
myList.add(7);  
myList.add(15);  
myList.add(7);  
myList.add(7);  
  
Set<Integer> mySet = new HashSet<>(myList);  
System.out.println(mySet.size());
```

2) Set ve Liste arasındaki ortak öğeleri yazdırmak için kod yazınız.

```
List<Integer> myList = new ArrayList<>();  
myList.add(10);  
myList.add(31);  
myList.add(15);  
myList.add(7);  
myList.add(13);  
  
Set<Integer> mySet = new HashSet<>();  
mySet.add(10);  
mySet.add(7);  
mySet.add(35);  
mySet.add(13);  
  
mySet.retainAll(myList);  
System.out.println(mySet);
```

3) Bir String' de kullanılan farklı karakterleri yazdırmak için kod yazınız.

Örnek: 'Mississippi' ⇒ Misp

```
String s = "Mississippi";  
String[] arr = s.split("");  
List<String> characters = Arrays.asList(arr);  
  
Set<String> mySet = new HashSet<>(characters);  
System.out.println(mySet);
```

4) Bir liste ve bir set oluşturunuz. Set içerisinde var olan listenin bütün elemanlarını kaldıran bir kod yazınız.

```
List<Integer> myList = new ArrayList<>();  
myList.add(10);  
myList.add(31);  
myList.add(15);  
myList.add(7);  
myList.add(13);
```

```
Set<Integer> mySet = new HashSet<>();  
mySet.add(10);  
mySet.add(7);  
mySet.add(35);  
mySet.add(13);
```

```
mySet.removeAll(myList);  
System.out.println(mySet);
```

5) Bir listede tekrarlanan öğeler olup olmadığını kontrol etmek için kodu yazınız.

```
List<Integer> myList = new ArrayList<>();  
myList.add(10);  
myList.add(31);  
myList.add(15);  
myList.add(7);  
myList.add(13);  
  
Set<Integer> mySet = new HashSet<>(myList);  
  
if(myList.size()>mySet.size()){  
    System.out.println("Listede tekrarlanan öğeler var");  
}else{  
    System.out.println("Listede tekrarlanan öğeler yok");  
}
```

- 1) Aşağıda key olarak ürün adlarını ve value olarak ürün sayısını içeren bir map bulunmaktadır. Toplam ürün sayısını bulmak için kodu yazınız.

```
Map<String, Integer> product = new HashMap<>();  
product.put("Laptop", 12);  
product.put("TV", 53);  
product.put("Refrigerator", 12);  
product.put("Music System", 87);
```

```
Collection<Integer> numOfProducts = product.values();
```

```
Integer sum = 0;  
for(Integer w : numOfProducts){  
    sum = sum + w;  
}  
System.out.println(sum);
```

- 2) Aşağıda key olarak ürün adlarını ve value olarak ürün sayısını içeren bir map bulunmaktadır. Ürünler arasında " Laptop " olup olmadığını kontrol etmek için kodu yazınız.

```
Map<String, Integer> product = new HashMap<>();  
product.put("Laptop", 12);  
product.put("TV", 53);  
product.put("Refrigerator", 12);  
product.put("Music System", 87);
```

```
String expectedProduct = "Laptop";  
if(product.containsKey(expectedProduct)){  
    System.out.println(expectedProduct + " vardır");  
}else{  
    System.out.println(expectedProduct + " yoktur");  
}
```

- 3) Aşağıda key olarak ürün adlarını ve value olarak ürün sayısını içeren bir map bulunmaktadır. Ürün sayılarını artan sırada yazdırınız.

```
Map<String, Integer> product = new HashMap<>();
```

```
product.put("Laptop", 82);
```

```
product.put("TV", 53);
```

```
product.put("Refrigerator", 12);
```

```
product.put("Music System", 87);
```

```
product.put("Mobile Phone", 53);
```

```
Object[] productNumber = product.values().toArray();
```

```
Arrays.sort(productNumber);
```

```
System.out.println(Arrays.toString(productNumber));
```

- 4) Aşağıda key olarak ürün adlarını ve value olarak ürün sayısını içeren bir map bulunmaktadır. Ürün adlarını alfabetik sırayla yazdırınız.

```
Map<String, Integer> product = new HashMap<>();
```

```
product.put("Laptop", 82);
```

```
product.put("TV", 53);
```

```
product.put("Refrigerator", 12);
```

```
product.put("Music System", 87);
```

```
product.put("Mobile Phone", 53);
```

```
Set<String> productNamesAsSet = product.keySet();
```

```
List<String> productNamesAsList = new ArrayList<>(productNamesAsSet);
```

```
Collections.sort(productNamesAsList);
```

```
System.out.println(productNamesAsList);
```

**5) Bir String' deki kelimelerin kaç defa tekrarlandığını ifade eden kod yazınız.
(Büyük/küçük harfe duyarlı değil)**

```
String s = "Apex is easy. Type codes to learn apex. To earn money learn apex.";
```

```
s = s.replaceAll("\\p{Punct}","").toLowerCase();
```

```
Map<String, Integer> wordOccurence = new HashMap<>();
```

```
String[] words = s.split(" ");
```

```
for(String w : words){
```

```
    Integer numOfOccurence = wordOccurence.get(w);
```

```
    if(numOfOccurence == null ){
```

```
        wordOccurence.put(w, 1);
```

```
    }else{
```

```
        wordOccurence.put(w, numOfOccurence+1);
```

```
    }
```

```
}
```

```
System.out.println(wordOccurence);
```


6) Belirli bir listede tekrarlanan öğelerin sayısı nasıl kontrol edilir?

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(21);
myList.add(12);
myList.add(13);
myList.add(12);
myList.add(21);
myList.add(35);

Map<Integer, Integer> myMap = new HashMap<>();

for(Integer w : myList){
    Integer numOfOccurence = myMap.get(w);
    if(numOfOccurence==null){
        myMap.put(w, 1);
    }else{
        myMap.put(w, numOfOccurence+1);
    }
}

Collection<Integer> myValues = myMap.values();
Integer counter = 0;
for(Integer w : myValues){
    if(w>1){
        counter++;
    }
}

System.out.println("Tekrarlanan oge sayisi " + counter + " adettir");
```

ANSWER KEY

Introduction to Java

1-D	3-D	5-C	7-D	9-D
2-D	4-A	6-A	8-A-B-C	10-A-B-C

Variables

1-A	3-A	5-C-D	7-A	9-A-B-C
2-A	4-B	6-B	8-A-D	10-A-B

Variables Wrappers

1-B	3-B-C	5-A-B-C-D	7-A-B	9-A-B-C
2-D	4-A	6-A-D	8-C	10-C

Increment Decrement Concatenation

1-A	3-B	5-B	7-C	9-A-B-C
2-B	4-A	6-A-B	8-A	10-A-B-D

Scanner

1-D	3-A	5-C	7-A-B-C-D
2-B-C-D	4-B-C-D	6-D	

Type Casting

1-A	3-A-B	5-B	7-A	9-B
2-B	4-B	6-A	8-A	10-B

If Else - If Elseif

1-B	3-B	5-B	7-A	9-D
2-A	4-A	6-A	8-D	10-A-B-C

Nested If Ternary

1-B	3-A-B-C	5-D	7-D	9-A-B-C
2-A	4-A	6-D	8-D	10-A-B-C-D

Nested Ternary Switch

1-C-D	3-A-D	5-D	7-A	9-A
2-B	4-A-B-C-D	6-A-B-C	8-A	10-C

String Methods

1-C	3-C	5-A	7-C-D	9-A-B-C-D
2-C	4-B	6-D	8-A-B-C	10-D

String Methods For Loop

1-B-C-D	3-B	5-A	7-C	9-B-D
2-B-C-D	4-A-B-D	6-A-B-D	8-B	10-B-D

For Loops

1-B	3-A	5-A	7-A	9-A
2-B	4-A	6-D	8-A-C-D	10-A-B-C-D

While Loops

1-B	3-C	5-A-C	7-B	9-D
2-C	4-B	6-A-B	8-C	10-D

Do While Loop

1-A	3-D	5-A	7-C	9-B
2-B	4-A-B	6-B	8-C	10-A-B-C

Instance Local Variables Method Call

1-D	3-C	5-D	7-A-B-C
2-C	4-A	6-D	8-A-B-D-E

Constructors

1-D	3-A-B-C-D	5-A-B-E	7-D
2-C	4-C	6-A	8-A-B-D

Static Keyword

1-A	3-A-B-C-D	5-A	7-A	9-A
2-A	4-C-D	6-A	8-A	10-A

Static Block Arrays

1-C	3-A	5-A	7-A	9-A	11-B
2-A	4-B	6-A-C	8-A	10-C	

Arrays

1-C	3-A	5-B-C	7-A	9-C	11-D
2-A	4-B	6-A	8-C	10-C	

Multi Dimensional Arrays

1-B	3-D	5-A
2-D	4-A	

Array Lists

1-A	3-A	5-A	7-B	9-A	11-A
2-A	4-A	6-A	8-B-C-D	10-A	

For Each Loops

1-A	3-A	5-A	7-A	9-A-B-C
2-C	4-A	6-A	8-A	10-A

Date Time

1-A-B-C-D	3-A	5-B	7-A	9-A
2-A-B-C-D	4-A	6-D	8-A	10-C

Varargs Access Modifiers

1-A	3-B	5-A	7-A-B	9-A
2-A	4-A-B-C-D	6-A-D	8-A-D	10-B-D

String Builder

1-B	3-A	5-A-B	7-A	9-A
2-A-B-C-D	4-C	6-A	8-A-B	10-A-B-C-D

Encapsulation

1-A-B	3-C	5-A-B	7-A-B-C-D	9-B
2-A	4-D	6-B	8-A-B	

Inheritance

1-A	3-B	5-A	7-ABCD
2-ABCD	4-ABC	6-A	

Inheritance Constructor Call

1-A	3-B	5-ABC	7-C	9-AB
2-A	4-B	6-ABC	8-A	10-AB

Overriding

1-D	3-ABC	5-ABCD	7-D	9-ACD
2-A	4-ABC	6-ABC	8-AC	10-A

Overloading-Overriding

1-A	3-A	5-A	7-ABC	9-ABCD
2-A	4-A	6-C	8-ABCD	10-ABC

Exceptions - 1

1-BC	3-ABC	5-ACD	7-A	9-A
2-C	4-BC	6-B	8-D	

Exceptions - 2

1-D	3-A	5-D	7-A	9-D	11-A
2-B	4-B	6-C	8-A	10-ABCDE	

Abstract Classes

1-C	3-ACD	5-D	7-A	9-A
2-A	4-ABC	6-A	8-A	10-CD

Interface - 1

1-B	3-A	5-C	7-A
2-ABC	4-D	6-A	8-BD

Interface - 2

1-A	3-C	5-A	7-A	9-D
2-CD	4-B	6-A	8-A	10-ACE

Iterators

1-ABC	3-AB	5-ABCD	7-B
2-AB	4-A	6-ABCD	8-A

Collections - 1

1-A-B-C-D	3-A-B-C-D	5-C	7-A-B	9-A-B
2-C	4-A-B-C-D	6-A-D	8-A-B-C-D	10-A-B-C

Collections - 2

1-A	3-A-B-C	5-A-B-C-D	7-B	9-A
2-A	4-A	6-A-B	8-A-B-C-D	10-B-C