

1. Cevap E. StringBuilder sınıfı referansı henüz değeri atanmadığı için boştur. for döngüsü içindeki ilk satırda indexOf() metodunda -1 döneceği için bu nedenden dolayı kod derlense de çalışma zamanında java.lang.StringIndexOutOfBoundsException hatası fırlatır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/StringIndexOutOfBoundsException.html>
2. Cevap C ve E. Verilenlerden C ve E seçenekleri aynı operatör seviyesindedir.
Kaynak: <https://introcs.cs.princeton.edu/java/11precedence/>
<http://pages.cs.wisc.edu/~willb/cs302/java-operator-precedence.pdf>
http://www.cs.bilkent.edu.tr/~guvenir/courses/CS101/op_precedence.html
3. Cevap B, C ve F. JavaBean'leri argümanlı getter, argümanlı setter ve argümanlı is metotlarından oluşur. Bu nedenle bu yapıya metot deklarasyonları B, C ve F seçenekleridir.
Kaynak: <https://www.geeksforgeeks.org/javabean-class-java/>
4. Cevap A ve E. Verilen kodun ekran çıktısı verilen satırında derleme hatası vardır. crossword dizisinin boyutunu size() metodu ile değil, length özelliği ile alırız. Bu nedenle kodun derlenmesi için bir satırın düzelmesi gerekir. A seçeneği doğrudur. B ve C seçenekleri doğru değildir. crossword dizisinin length değeri 10'dur. Bu nedenle her iki boyut için de döngü 10 iterasyon yapar ve "x" karakterinin unicode değeri olan 120 değeri atanır. Dizideki elemanların yarısı 120 değeri ile yüklenir, diğer yarısı 0 değeri ile yüklenir. Bu nedenle E seçeneği doğru D ve F seçenekleri yanlıştır.
5. Cevap B ve D. Bir uygulama dosya sisteminin geçici olarak kullanılamaz hale gelmesi ya da kullanıcı hatalı değeri girmesi gibi durumları çözebilir. Bu tip durumlar Error fırlatmayı gerektirmez. Error sınıfı Throwable sınıfının bir subclass'ıdır. Olması beklenen durum, bir uygulamanın asla Error fırlatmamasıdır. Ancak fırlatabilir ve bu durumda bu hata handle edilebilir. Bu bilgilere göre A, C ve E seçenekleri yanlıştır. B ve D seçenekleri doğrudur.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/Error.html>
6. Cevap A, C ve D. Verilen import ifadelerine göre, A seçeneğindeki forest.Bird sınıfına ilk import ifadesi sayesinde paket ismi yazmadan erişilebilir. B seçeneğinde savana.sand paketinde yer alan Wave sınıfına paket ismi yazmadan erişilemez. Import ifadeleri recursive olarak alt paketler içindeki sınıfları dahil etmeyeceğinden dolayı Wave sınıfına paket ismi yazmadan erişilemez. C seçeneğindeki jungle.tree paketindeki Huicungo sınıfına ikinci import ifadesi sayesinde paket ismi yazmadan erişilebilir. D seçeneğindeki java.lang.Object sınıfına, java dosyalarının default import edildiği için hiçbir import ifadesi olmasa dahi paket ismi yazmadan erişilebilir. E ve F seçeneklerindeki Sloth sınıfına ve bonobo paketine de gerekli import ifadesi olmadığı için paket ismi yazmadan erişilemez.
7. Cevap C. Mutable değiştirilebilir, immutable değiştirilemez objelerdir. ArrayList bir obje seti sakladığı için değişebilir. StringBuilder her yeni bir nesne oluştuğunda değişebildiği için mutable'dır. Ancak String sınıfı ve LocalDateTime sınıfının bazı özellikleri değiştirilmez ve bunlar immutable'dır. Cevap 2'dir.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>
<https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html>
<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDateTime.html>
8. Cevap C. Verilen kod parçasında builder isimli StringBuilder referansına "Leaves growing" ifadesi eklenmiş ve do-while döngüsünde ilk 5 karakteri silinmektedir. builder referansının boyutu 5'ten büyük olduğu için ikinci iterasyonda 5 tane daha karakter silinir. Son olarak "wing" ifadesi elde kalır. Bu değer length() değeri 5'ten küçüktür ve iterasyona yeniden girmez ve bu ifade ekran çıktısı olur.

9. Cevap D. Verilen kodda ceiling isimli String değişkende “up” değeri yüklenmiştir. floor isimli referansın gösterdiği değer “up”tır. wall isimli referansa, floor isimli referansın gösterdiği değer atanmıştır. Bu nedenle, wall ve ceiling’in değerleri eşitken, wall ve floor referanslarının gösterdiği değer aynıdır. Bunlara göre kod, “false false true” şeklinde çıktı üretecektir.
10. Cevap C. Verilen kod çalışma zamanında ilk üç ifade false, diğer üç ifade true olacaktır. İlk satırda “LOL” == “lol” kontrolü ile false değeri ekrana basılır. İkinci satırda false dönmektedir. Çünkü, toUpperCase() metodu String döner ve String bir immutable tiptir. Bu nedenle heapte farklı noktada bir değer oluşur. Bundan dolayı false döner. Üçüncü satırda “LOL” ve “lol” değerleri equals() metodunda kontrol edilince false döner. Dördüncü satırda değerler toUpperCase() metodu ile “LOL” yapılır ve “LOL”=”LOL” true döner. Beşinci ve altıncı satırlarda da equalsIgnoreCase() metodu ile değerler eşitlenerek kontrol edilir ve true değeri basılır. Toplamda 3 tane false, 3 tane true değeri ekrana basılır.
11. Cevap A, B ve C. Verilen kodda inner ve outer etiketlerine zıplanmamıştır. Bu nedenle 15, 16, 17 ve 21 numaralı satırların syntax’i bozmadan çıkarılmaları varyanslarında kod aynı çıktıyı üretmektedir. Yani 15 ve 17’yi aynı anda, 15, 16 ve 21’i aynı anda çıkarmak “xxxxx” sonucu üretecektir. E ve D seçenekleri uygulanırsa kod derlenmez hale gelmektedir.
12. Cevap B ve C. Long veri tipinde noktalı gösterim olmadığından dolayı B ve C seçenekleri derlenmez.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/Long.html>
13. Cevap A. Verilen time değişkenine saat değeri 1, dakika değeri 11 atanmaktadır. while döngüsü kontrolü false döneceği için döngüye girmez. Bu nedenle herhangi birşey basılmaz.
14. Cevap D. Verilen kod derlenir ancak çalışma zamanında NullPointerException fırlatır. Çünkü game dizisi deklare edilirken boyutu verilmemiştir.
15. Cevap C ve E. Verilen kodda B, D ve F seçeneklerindeki ifadeler derlenmez. ArrayList ham bir veri tipidir. Hangi wrapper sınıfı toplayacağını ifade etmek gerekmektedir. A seçeneğindeki deklarasyon derlenir ancak warning verecektir. Nedeni ise ArrayList’e String değerlerle doldurulacağını söylenmemiş olmasıdır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
16. Cevap B ve D. Verilen kod işletildiğinde shoe1 ve shoe3 değişkenleri “flip flop”, shoe2 değişkeni “croc” değeri ile yüklenir. Bu nedenle shoe1 ve shoe3 değişkenlerinden birisi garbage collection için uygundur. Ancak çöp toplanması gc tarafından garanti edilmez. Bu nedenle B ve D seçenekleri doğrudur.
Kaynak: <https://www.geeksforgeeks.org/garbage-collection-java/>
17. Cevap C. Verilen kodda Fish.getFish() metodu BubbleException fırlattığı için main() metodunda throws keyword’ü ile metot imzasına eklenmelidir. main() metodunda getFish() metot çağırısı exception fırlatacağı için bu tek satırdan dolayı kod derlenmez.
18. Cevap A. Verilen kod negatif sayıyı arayarak ekrana basan lambda ifadesi kullanan bir örnektir. Bu nedenle sadece -5 değeri için if kontrolüne girer ve ekrana -5 değerini basar. Yani tek satır basılır.
19. Cevap F. Bir try ifadesi ya bir catch bloğuna ya da finally bloğuna sahiptir. Ya da her ikisine sahiptir. Bu bilgiyi kapsayan bir seçenek olmadığından cevap seçeneklerden hiçbirisi değildir. I ve III numaralı maddelerde verilen anahtar sözcükler finalize try ifadesi için kullanılan bir anahtar kelime değildir. throws anahtar kelimesi ise metot imzalarında exceptionları deklare etmek için kullanılır.
20. Cevap A. Verilen kodda result değişkeni önce 8 olduğundan while döngüsüne girer. Sonra 1 artar ve içteki do-while döngüsüne girer. İçteki döngüde değeri 5 olana kadar azalır. Sonrasında loop etiketine kadar döngü kırılır ve ekrana 5 değeri basılır.

21. Cevap C. Verilen kodda teeth değişkeni Alligator sınıfının static bir üyesidir. Burada snap() metodu içindeki argüman olan teeth ile karıştırılmaması gerekir. Instance variable olan teeth her kurucu metod çağrısında 1 artar. Bu nedenle ekrana önce 1 sonra 2 yazılarak uygulama sonlanır.
22. Cevap A. Verilen kodda concat metodu() ile "b" ve "lack" ifadeleri birleştirilmektedir. Ancak değişkene atanmadığı için witch değişkeni değeri "b" olarak ekran çıktısı olacaktır.
23. Cevap A, C ve F. Java 8 ile birlikte interface metotları abstract metotlara ilave olarak default ve static metotlar da olabilmektedir. interface metotları erişim belirleyici verilmezse public olur. Zaten private ya da protected olamazlar. Ayrıca final olmaları durumunda override edilemeyecekleri için final anahtar kelimesi de kullanılamaz.
24. Cevap D. Verilen kodda while döngüsündeki "tie=null" kontrolü sonucu true yada false değildir. Bu nedenle bir yazım hatası vardır ve kod derlenmez. == operatörü çözüm üretilirse ekran çıktısı shoelace olacaktır.
25. Cevap B ve F. Eğer bir .java uzantılı kod dosyası package deklarasyonu içermiyorsa default paket içinde konumlanmaktadır. A seçeneğindeki bilgi ile ilgisi yoktur. Dolayısıyla paket deklarasyonu içermesi de gerekmez. Bu nedenle A ve D seçenekleri yanlıştır. Boş bir java kaynak kod dosyası hiçbir import ifadesi yazılmasa dahi java.lang.* paketini import eder, java.util.* paketini import etmez. Dolayısıyla java.lang.Object sınıfı da import edilmiş olacaktır. B ve F doğru, C yanlıştır. Bir java dosyası import ifadesi içermek zorunda değildir. E seçeneği de yanlıştır.
26. Cevap D. m1 numaralı satırdan dolayı kod derlenmez. Çünkü, Tree sınıfı Plant ve Living interface'lerini uygulamaktadır. Ancak hangisinin grow() metodunu override edeceği hususu belirsizdir ve m1 satırında derleme hatası verecektir. Hangi interface'in grow() metodu override edilirse buna bağlı olarak ekran çıktısı "Grow!" ya da " Growing!" olacaktır.
27. Cevap D. Verilen kodda profit\$\$\$ değişkeni main() metodunda yerel bir değişken olduğundan dolayı default değeri yoktur. Bu değişken initialize edilmeden kod derlenmez. Instance variable ya da sınıf değişkeni haline getirilirse default değeri false olacaktır ve kod derlenerek ekran çıktısı üretecektir. Derleme hatası ise ekran çıktısı alınan satırda olduğundan dolayı doğru cevap D seçeneğidir.
28. Cevap A. Verilen boşluklara sırayla post-decrement operatörü x—ve pre-increment operatörü ++x gelecektir. Post-increment/post-decrement operatörleri değerlere işlem yapar ve orijinal değer ile dönerken, pre-increment/pre-decrement operatörleri değerlere işlem yapar ve yeni değerle döner.
29. Cevap B, C, E. Seçenek A, D ve F tarafından beyan edilen yapıcılar sorunsuz derlenir. Seçenek B derlenmez. Aynı sınıfta bir üst kurucuya veya kurucuya çağrı olmadığından, derleyici kurucunun ilk satırı olarak argümansız bir super () çağrısı ekler. Big'un bağımsız değişken yapıcısı olmadığından, noargument yapıcısı Trouble () derlenmez. C seçeneği de derlenmez çünkü super () ve this () aynı kurucuda çağrılmaz. Super () deyimini kaldırılırsa, yinelemeli bir yapıcı çağrısı olacağından yine de derlenmeyeceğini unutmayın. Son olarak, Seçenek E derlenmez. Bir String ve ardından uzun bir değer alabilen eşleşen bir yapıcı yoktur. Derin giriş argümanı bu kurucuda bir int olsaydı, o zaman Seçenek D'de kullanılan kurucu ile eşleşir ve sorun olmadan derlenirdi.
30. Cevap E ve F. main() metodu static olduğu için static değişkenlere erişebilecektir. E ve F seçeneklerindeki deklarasyonlar insert code kısmına eklenirse 100 sonucu ekran çıktısı olarak basılacaktır.
31. Cevap B ve E. Üçlü kontrol yapısında boolean şarta bağlı olarak bu değer true ise sol taraf, değilse sağ taraf icra edilir. A seçeneği yanlıştır. Bir switch ifadesi en fazla bir default case tanımlanmasına izin verir. B seçeneği doğrudur. if-then yapısı sadece 0 ya da daha fazla else if

bloğu ve en fazla bir tane else bloğu inşa edilmesine izin verir. C seçeneği yanlıştır. Java'da | operatörü bitset or operatörüdür ve değerin sağ ve solundaki değeri tüm bitlerini mantıksal OR işlemine sokarak yeni değer elde edilmesini sağlarken || operatörü ise boolean bir sonuç döndürür. Dolayısıyla aynı değerleri üretmezler ve D seçeneği yanlıştır. ! operatörü ise nümerik ifadelerle uygulanamazlar. E seçeneği doğrudur.

32. Cevap C. Java'da deleteCharAt() metodu indeks değerini verilen değeri StringBuilder referansından silerken referansı günceller. delete() metodunda 1, 1 parametreleri ile hiçbir değer silinmeden bırakılmaktadır. Buna göre önce 0. indeks değeri silinir. Sonraki satırda 1. indeksten 1. indekse kadar olan –yani hiçbir- değer silinmez. Ekran çıktısı “ed” olacaktır.

Kaynak: <https://www.geeksforgeeks.org/stringbuffer-deletecharat-method-in-java/>
https://www.tutorialspoint.com/java/lang/stringbuilder_delete.htm#:~:text=StringBuilder,.e nd%2C%20no%20changes%20are%20made.

33. Cevap A ve D. Java isimlendirme kurallarına göre A ve D seçeneklerindeki metot isimleri geçerlidir.

Kaynak: <https://www.javatpoint.com/java-naming-conventions>

34. Cevap B ve C. A seçeneğinde kalıttan ziyade kapsüllemenin özelliği verilmiştir. Kapsülleme yöntemi, farklı bir sınıfın verilerine ulaşmak için static metot kullanmaktan daha iyi bir yöntemdir. A seçeneği yanlıştır. Miras alınmayan herhangi bir sınıftan java dosyası yaratılmasını sağlamadığı için D seçeneği yanlıştır. Kalıtım kullanmak kodu daha karmaşık değil, daha açık ve anlaşılır hale getirir. E seçeneği yanlıştır. Kalıtım alt-üst sınıf hiyerarşisi kurarak, metot overriding'e olanak sağlarken ortak kullanılan metot ve özelliklerin alt sınıflarda otomatik olarak implemente edilmesini sağladığı için kolaylık sağlar. B ve C seçenekleri doğrudur.

35. Cevap E. Java paketleri “.” karakteri ile ayırır. Fonksiyonel programlamayı ve polimorfizmi destekler ve Java nesneye yönelik bir dildir. Bu nedenle verilenlerin hepsi doğrudur.

Kaynak: https://www.tutorialspoint.com/functional_programming/functional_programming_polymorphism.htm

36. Cevap C. Verilen kodda listing isimli dizi 2 boyutludur ve ilk boyutu 3, ikinci boyutu 2 elemanlıdır. Dolayısıyla listing.length değeri 3 iken, listing[0].length değeri 2 olacaktır.

37. Cevap A, B ve E. Java byte, char, short ve int türü bile bunların wrapper sınıfları olan Character, Byte, Short ve Integer sınıflarına switch ifadesine argüman olma desteği verir. Bu nedenle A, B ve E seçeneklerindeki Character, Byte ve String türleri switch ifadesi için geçerlidir.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>

38. Cevap D. k2 numaralı satırdan dolayı kod derlenmez. Çünkü d isimli değişken deklare edilmemiştir. Hata giderilirse ekran çıktısı true olacaktır.

39. Cevap B, C ve E. Verilenlerden B, C ve E seçeneklerindeki yorum satırları Java için geçerli birer yorum satırıdır.

Kaynak: https://www.w3schools.com/java/java_comments.asp

40. Cevap A ve F. Verilen kodda Grass sınıfı üyeleri static olduğundan dolayı static import yapılması gerekmektedir. Bu nedenle A veya F seçeneğindeki gibi impor yapılarak Deer sınıfı derlenir hale getirilmelidir.

Kaynak: <https://www.geeksforgeeks.org/static-import-java/>
<https://docs.oracle.com/javase/8/docs/technotes/guides/language/static-import.html>

41. Cevap D. Verilen kod derlenir ancak çalışma zamanında UnsupportedOperationException hatası fırlatır. Bunun anlamı, java.util.asList() metodu argüman olarak aldığı listeyi sabit boyutlu bir dizi haline getirir. Değerlerin değiştirilmesine izin verir ancak add() ve remove() metotları ile eleman ekleme/çıkarma işlemlerine izin vermez. Bu nedenle bu exception fırlatılır.

Kaynak: javadoc java.util.Arrays.asList()

42. Cevap A ve D. Verilen kodda A ve D seçeneklerinde verilen değişiklikler uygulandığında kod derlenmeye devam etmektedir. Değişken ismi “_” ile başlayabilir ve int bir değişkenin değeri içinde “_” kullanılabilir.
43. Cevap B. Verilen kodda main() metodu komutu satırı argümanı metot içinde initialize edilerek değer yüklenmiş ve bu değerler sort() metodu ile sıralanmaktadır. Sonrasında ise B seçeneğindeki çıktıyı üreterek sonlanacaktır.
44. Cevap D. Java’da bir sınıfın üyelerine aynı sınıf, alt sınıf, aynı ve farklı paketlerden erişebilmek için public olarak imzalanması gerekmektedir. static yapılırsa bir sınıfın örnek referansı olmadan erişime açık hale gelecektir. Bu nedenle boşluklara public ve static gelecektir. final kelimesi bir erişim belirleyici değildir. Bu nedenle soruyla ilgisi yoktur. Aynı şekilde class, instance ve default şeklinde erişim belirleyicileri yoktur ve B, C ve E seçenekleri de yanlış olacaktır.
45. Cevap A. Verilen kod örneğinde exams listesi boyutu 2’dir. İç içe iki for döngüsü toplamda $2 \times 2 = 4$ iterasyon yaparak çalışır. Ancak her print işlemini aynı satıra basacağından dolayı toplamda sadece bir satır ekran çıktısı vardır. Bu nedenle cevap A seçeneğidir.
46. Cevap C ve D. javac komutu .java uzantılı text kaynak kod dosyasını .class uzantılı byte koda derler. Bu nedenle C ve D seçenekleri doğrudur.
47. Cevap C. java.lang.Integer.parseInt() metodu ilkel int değeri döner. Bu nedenle int tipinde ya da daha geniş bir ilkel tipe veya int türünün wrapper classı olan Integer’a atanabilir. Bu nedenle verilenlerden three ve four değişkenleri için olan satırlar derlenebilir.
Kaynak: javadoc java.lang.Integer.parseInt(String s)
48. Cevap B, D ve F. 3 değerini dönen drive() metotları sırayla double, car ve short türünden argüman alan metotlardır. short ve byte türü için aynen uygulanır. Argüman float olduğunda ise floatı kapsayabilecek olan double türünden argüman alan drive() metodu çağrılır ve 3 değeri basılır. A seçeneği boolean olduğu için zaten bir literal atanamaz ve kod derlenmez.
49. Cevap A. Verilen koddaki main() metodu bir Java uygulamasına giriş için geçerli bir main() metodu değildir. main() metodu static olmalıdır. Bu nedenle kod sorunsuz şekilde derlenir, ancak run edilemez. Eğer geçerli bir main() metodu inşa edilirse sırayla false, true, true, false, false değerlerini alt alta ekrana basacaktır.
50. Cevap D. Verilen kodda swing() metoduna 2 tane 0 değeri argüman olarak geçilmektedir. Metoda girildiğinde try bloğunda işletilen kodda beats[2] denilerek ArrayIndexOutOfBoundsException fırlatılmaktadır. Bu exception RuntimeException’ın subclassıdır ve hata ilk catch bloğunda yakalanır. Burada ekrana 2 değeri basılır. Peşinden finally bloğu icra edilerek 4 değeri basılır. Daha sonra main() metoduna dönülerek 5 değeri ekrana basılır ve uygulama sonlanır.
Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/lang/ArrayIndexOutOfBoundsException.html>
51. Cevap E. Verilen kodda ekran çıktısı alınan satırda ArrayIndexOutOfBoundsException hatası fırlatılır. İlk iterasyonda drinks listesinde sondan ilk eleman “cup” değeri ekrana basılsa da sonraki iterasyonda container değişkeni azaltılması gerekirken artırılmaktadır. Bu nedenle 2. iterasyonda drinks.get(2) çağrısı yapıldığında hata fırlatılır.
52. Cevap A, E, F. Geçerli bir main() metodu public static olarak imzalanır ve String türünden bir dizi argüman alır. Bu dizi gösterimi “[]” ile ya da “...” ile belirtilebilir ve final olarak imzalanabilir. Bu tanımlara uyan geçerli main() metodu deklarasyonları A, E ve F seçeneklerindeki gibidir.
53. Cevap C, D ve E. Verilen kodda 10, 11 ve 12 numaralı satırların kaldırılması durumunda kod derlenecektir. case ifadesinde sabit ya da final olarak tanımlanan int, char, byte veya short primitiveleri ya da bunları kapsayan sınıflar ile beraber String kullanılabilir. Bu nedenle 10 ve 12 numaralı satırlar derlenmez. 11 numaralı satırdaki default ya da 14 numaralı satırdaki

default tan birisi silinmelidir. switch ifadesinde default bir tane olmalıdır. 14 numaralı satır seçeneklerde olmadığı için 11 numaralı satır silinmelidir.

54. Cevap A, B ve C. Verilen kodda Friend arayüzündeki getName() interface metodu protected yapılmıştır. public olması gerektiği için h1 satırında derleme hatası vardır. Buna bağlı olarak Friend interface'ini uygulayan Cat sınıfının getName() metodu da public olarak erişimi belirlenmelidir. Benzer şekilde Dog sınıfının getName() metodu da public olacaktır. Bu nedenle h1, h2 ve h3 satırlarından dolayı kod derlenmez.

55. Cevap B, E ve F. java.lang.Error ve java.lang.RuntimeException sınıfı ve subclassları unchecked exceptionlardır. Seçeneklerde verilenlerden ArithmeticException, IllegalArgumentException ve RuntimeException hataları unchecked exception'dır.

Kaynak: <https://www.geeksforgeeks.org/checked-vs-unchecked-exceptions-in-java/>
<https://docs.oracle.com/javase/8/docs/api/java/lang/package-tree.html>

56. Cevap F. Verilen kodda printMessage() metodu çağrılmadan önce space değişkeni 5, ship değişkeni üçlü kontrol ifadesinden sonra 10 değeri ile yüklenir. Metoda çağrı yapıldığında ship 10 olduğu için kontrole girer ve "Goodbye" değeri yazılır. Sonraki if kontrolünde ship 10 olduğu için else bloğuna girerek "See you again" yazılır. Sonuç olarak 2 satır yazılır ve bu seçenek olmadığı için cevap F seçeneği olacaktır.

57. Cevap B, C ve D. Verilen kodun derlenmesi için B, C veya D seçeneğindeki şekillerde erişim belirleyicileri verilebilir. Aşağıdaki tabloya göre belirlenecektir.

	default	private	protected	public
Same Class	Yes	Yes	Yes	Yes
Same package subclass	Yes	No	Yes	Yes
Same package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

Kaynak: <https://www.geeksforgeeks.org/access-modifiers-java/>

58. Cevap B. Verilen kod q1 numaralı satırdan dolayı derlenmez. Somut bir sınıf içinde soyut metot implemente edilemez. Bu nedenle bu metot somut hale getirilmelidir.

Kaynak: <https://www.geeksforgeeks.org/difference-between-abstract-class-and-concrete-class-in-java/>

59. Cevap C. Verilen kodun 6 ve 8. satırlarında derleme hatası vardır. Tarih bilgisi formatlama için DateTimeFormatter sınıfının ofPattern() metodu kullanılır. 6. satırda ise LocalDate.of() metodu ile initialization yapılır. Bunlar yapıldıktan sonra da main() metodu geçerli bir giriş noktası haline getirilmelidir.

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html#method.summary>

60. Cevap A ve E. Bir try-catch-finally yapısında try bloğu, 0 ya da daha fazla catch bloğu ile beraber 0 ya da 1 tane finally bloğu içerir. finally bloğu varsa catch bloğu olmak zorunda değildir. Ancak ikisinden en az biri olmalıdır. Buna göre doğru bilgi A ve E seçeneklerinde yer alan bilgilerdir.

61. Cevap B. Verilen kodda operatör önceliklerine göre değişkenlere değer atanmaktadır. fish değişkenine 4, mammals değişkenine 9 değeri atanır ve ikisi toplandıktan sonra "" ifadesi ile birleştirilerek String bir değer olarak "13" değeri ekrana basılır.

62. Cevap A, C ve E. Bir nesne açık bir casting olmadan bir üst sınıfa ya da devralınan interface'e dönüştürülebilir. Bir nesnenin bir örnek referansına atanması nesneyi değiştirmez. Bunlardan dolayı A, C ve E seçenekleri doğrudur. Bir nesne ClassCastException fırlatılmadan bir üst sınıfa ya da interface'e atanabilir. F seçeneği de yanlıştır.

63. Cevap F. Java'da `binarySearch()` metodunda arama yapmak için dizi öncelik sıralanmalıdır. Sıralı olmayan dizide arama sonucu garanti edilmez. Bu nedenle cevap F seçeneğidir.
Kaynak: <https://www.geeksforgeeks.org/arrays-binarysearch-java-examples-set-1/>
64. Cevap B. Verilen kodun icrası sonrası `shoe1` ve `shoe3` değişkenleri "croc" değeri ile yüklüdür. `shoe2` değişkeninde ise "sandal" değeri yüklüdür. Bu nedenle bir nesne garbage collection için uygundur.
65. Cevap B. Metot deklarasyonunda exception imzalamak için `throws` anahtar sözcüğü kullanılır. Bir hata fırlatıldığında bu hata için implemente edilen `catch` bloğu hatayı handle eder. Devam eden bir processte hata fırlatmak için ise `throw` anahtar sözcüğü kullanılır. Bu nedenle boşluklara sırayla `throws`, `catch` ve `throw` gelecektir.
66. Cevap B ve E. `nycTourLoops` dizisi 3 elemanlı, `times` dizisi ise 2 elemanlıdır. Eleman sayılarının farklı olmasından dolayı `for` döngüsünün 3. İterasyonunda `java.lang.ArrayIndexOutOfBoundsException` hatası fırlatır. Bunun öncesinde ilk iki iterasyonda `Downtown Day`, `Uptown Night` değerlerini basar. Sonuç olarak 2 satır basılır ve sonra exception alınır. Bu nedenle cevap B ve E seçeneğidir.
67. Cevap E. Sanal bir metodun `override` yeteneği sağlaması, Java'nın çokbiçimliliğine örnektir. Bu nedenle cevap E seçeneğidir.
`non-final`, `non-static`, `non-private`
Virtual'dan kasıt, bir metodun nasıl icra edileceğinin `compile-time`'da değil, `runtime`'da belirlenmesidir.
`final` olursa, `constant`'tır değiştirilemez. `private` ise başka classtan erişilmez. `static` metotlar `class` metottur. `static` metotlar, `override` değil, `hide` olur.
68. Cevap E. `seasons()` metoduna gönderilen argümanların 1. indeksinde yer alan değer "Fall" dur ve `length` değeri 4'tür. Sonraki satır `names[4]` değeri ekrana basılmak istenmektedir. `names` dizisinin boyutu 4 olduğundan dolayı 4. indeks dizi dışıdır ve `java.lang.ArrayIndexOutOfBoundsException` hatası fırlatılır. Cevap E seçeneği olacaktır.
69. Cevap D. Verilen kodun 8, 11 ve 15. satırlarında derleme hatası vardır. 8. satırda superclass'a çağrı yapan tek argümanlı bir yapıcı metot olmadığı için derleme hatası vardır. 11. satırda superclass'ın kurucusuna yapılan çağrıda parametre geçilmiştir. Bu kurucu metodun parametresi olmadığı için derleme hatası vardır. Son olarak 15. satırda `mn` isimli referansın `Drum` sınıfına cast edilmesi gerekmektedir.
70. Cevap B. Verilen kod "6, LONG" şeklinde ekran çıktısı üretir.
71. Cevap D. `main()` metoduna `throws Exception` imzası gereklidir. `openDrawbridge()` metodunda `OpenDoorException` yakalayan `catch` bloğu içinde yer alan `catch` bloğu ile aynı değişken ismi kullanılmıştır. Değişken isimleri düzenlenir. Ayrıca `catch` bloklarında yakalanan exception tipleri yer değiştirmelidir. Toplamda 3 tane derleme hatası vardır. Cevap D seçeneğidir.
72. Cevap D. Sınıf değişkenlerini sarmalayarak metotlar vasıtasıyla değiştirilmesi ve sınıf tasarımının geliştirilmesi Java'nın kalıtım ve kapsülleme özellikleriyle sağlanmaktadır. Bu nedenle boşluklara kalıtım ve kapsülleme gelecektir.
73. Cevap E. Verilen kodda `String` ifadesi küçük harfle başladığı için derleme zamanı hatası vardır. Eğer bu hata giderilirse `false` ve `true` değerleri ekrana basılacaktır.
74. Cevap A. Verilen kodda `main()` metoduna parametre olarak geçilen değerlerle bir dizi oluşmakta ve bu dizi sıralanarak ilk eleman dizide aranmaktadır. Doğal olarak 0. indekste bulacağı için 0 değerini `result` değişkenine atar ve ekrana basar. Ekran çıktısı 0'dır.
75. Cevap B, C ve D. Java'da gelişmiş `for` döngüsü bir dizi veya listede index sıralı olarak dolaşır. Gelişmiş bir `for` döngüsü `do-while`, `for` ve `while` döngülerine evrilebilir. Bu nedenle B, C ve D seçenekleri doğrudur. Ancak, bir `do-while` döngüsü en az bir kez çalışacağı için gelişmiş `for` döngüsü gibi yazılamayabilir. A seçeneği yanlıştır. Geleneksel `for` döngüsü boolean bir şarta

bağlı olduğu için gelişmiş for döngüsünü aynen yerine getirmeyebilir. E seçeneği yanlıştır. Benzer şekilde while döngüsü de koşul şartından dolayı gelişmiş for döngüsü olarak yazılamayabilir. F seçeneği de yanlıştır.

76. Cevap E. Verilen kodda derleme hatası vardır. LocalDate sınıfı sadece yıl, ay ve gün bilgisi özellikleri taşımaktadır. LocalDate.of() metodu 3 tane parametre almaktadır. Bu nedenle parametre eklenerek bu hata giderilir. LocalDate.getHour() ve LocalDate.getHour() şeklinde metotları yoktur. Bu nedenle, bu kısımlar ya LocalDate' göre evrilmelidir ya da LocalDateTime sınıfı vasıtasıyla bu işlevler yerine getirilmelidir.
77. Cevap C. Verilen kodda iki tane obje garbage collection için uygundur. balls scores dizileri 1 elemanlıdır ve dolayısıyla null değeri atandıktan sonra toplamda 2 obje çöpe atılabilir.
78. Cevap B. Verilen kodda date nesne referansına 01.01.2017 tarihi atanmaktadır. while döngüsünde date.getMonth() değeri JANUARY olacağından dolayı döngüye girer ve APRIL olana kadar azalır ve son olarak date referansında 01.04.2016 tarihi set edilmiş olarak 9. İterasyonda döngüde çıkar ve count değişkeni 1 yapılarak ekrana 1 değeri yazılır.
79. Cevap D. Verilen kodda 10, 11 ve 12. satırlar yazım hatalarından dolayı derlenmez. Override edilen metodun erişim belirleyici daha kısıtlayıcı olamaz. Bu nedenle 10, 11 ve 12 satırlardan kod derlenmez.
80. Cevap B ve E. default anahtar kelimesi switch yapısında hiçbir case'i ifade etmeyen durumları işletmek için ve Java 8 ile birlikte gelen interface'lerde gövdeli metot yazımında kullanılmaktadır. Bu nedenle B ve E seçeneklerindeki bilgi doğrudur. Erişim belirleyicisi olarak ya da rezerve kelime olduğu için metot ismi ve değişken ismi olarak default kullanamayız. A , B ve F seçenekleri yanlıştır. Bir interface'de abstract metoda default denilirse gövdeli olmak durumundadır. Bu durumda da abstract metot olmayacaktır. Bu nedenle D seçeneği de yanlıştır.