

- 1-) Bir try bloğu catch ya da final bloğu ya da her ikisini içermelidir. think()
methodu bir try bloğunu bildirir, ancak ek blok da bildirmez. Bu nedenle, kod
derlenmez. Cevap : D
- 2-) Doğru blok sırası try, catch, and finally. Cevap : B
- 4-) Exception ve RuntimeException Java uygulamalarında sık sık yakalanırken recommended hata
yakalanmaz. Cevap : A
- 5-) Kod derlenemez . Cevap : D
- 6-) ClassCastException, ArrayIndexOutOfBoundsException ve IllegalArgumentException
denetlenmeyen istisnalar ve herhangi bir zamanda atılabilir. IOException, kullanıldığında işlenmesi
veya bildirilmesi gereken kontrol edilmiş bir istisnadır. Cevap : B
- 7-) throws, throw . Cevap : A
- 8-) IOException, bir istisna alt sınıfıdır, bu nedenle ilgili herhangi bir yakalamada önce görünmelidir.
IOException'dan önce görünecekse, IOException bloğu atılan herhangi bir IOException zaten ele
alındığı için erişilemeyen kod olarak kabul edilir. Cevap : B
- 10-) P3 Satırı derlenmez. Cevap : C
- 12-) Kod sorunsuz çalışır ve outputu 1345 olur. Cevap : A
- 13-) Bir if-then ifadesinden farklı olarak, tek bir ifade, son olarak bir deyim parantez {} gerektirir.
Cevap : A
- 14-) Catch blokları yanlış sırada kullanıldığından kod derlenmez. Cevap : C
- 15-) Bir try ifadesi de hem yakalama hem de son olarak engelleme olarak kullanılabilir. Cevap : C
- 17-) Yanlış parantez kullanımından kaynaklı bir hata vardır. (parantez ifadesini { ile değiştirdiğimizde
kod doğru çalışır ve 4 outputunu verir. Cevap : D
- 19-) Bu sınıfların üçü de java.lang paketine aittir. Cevap : D
- 20-) Catch bloğunda değişken tip eksik olduğu için kod derlenmiyor. catch (Exception e) gibi bir ad
girilmeli. Cevap : C
- 21-) checked exceptions, java.lang.Error. Cevap : B
- 24-) main () yönteminde compute () yöntemine yapılan çağrı nedeniyle kod derlenmez. Cevap : C
- 25-) Listenin değeri null ise bir NullPointerException atanabilir. Aynı şekilde,
listenin değeri bir dizi ise bir ArrayIndexOutOfBoundsException oluşturulabilir. bir ClassCastException
listesi varsa atılır. Cevap : D
- 26-) StackOverflowError, NullPointerException Cevap : B
- 28-) Bu kod derlenmez çünkü catch ve finally bloklar yanlış. Cevap : D
- 29-) Finally bir bloğa sahip olmak için bir try ifadesi gerekli değildir. Ayrıca try ifadesinde herhangi bir sayıda catch
bloğu olabilir. Cevap : A
- 30-) Kod compile hatası verdi. Cevap : C

32-) m1 satırı bloğundan dolayı kod hata verir. Cevap : A

33-) ClassCastException özel durumu . Cevap : A

34-) Java'daki tüm istisnalar Throwable'dan devralınarak Seçenek C'yi doğru yanıt haline getirir. Cevap : C

36-) RuntimeException, ClassCastException'dan önce görünecekse, ClassCastException bloğuna ulaşamayan kod olarak kabul edilir, çünkü atılan ClassCastException RuntimeException catch bloğu tarafından zaten ele alınmıştır. Cevap : A

38-) throw exception dan dolayı hata verir. Kaldırdığımızda beat çıktısı verir. Cevap : C

39-) NullPointerException denetlenmeyen bir özel durumdur. Cevap : D

40-) Nesneyi oluşturmak için yeni anahtar kelime gerekli. Bu nedenle, kod derlemez. Cevap : D

42-) A, B ve C seçeneklerinin her biri yöntemin geçersiz geçersiz kılmalarıdır, çünkü dönüş türü boşlukla kovaryant olmalıdır. Bu nedenle, Seçenek D doğrudur. Cevap : D

43-) Kod hata verir. catch (Error) ifadesini **catch** (Exception **e**) olarak değiştirdiğimizde hata kalkar ve output unable! Olur. Cevap : D

45-) Hem IllegalArgumentException hem de ClassCastException inheritleri

RuntimeException, ancak ikisi de diğerinin alt sınıfı değildir. Bu nedenle listelenebilirler. Cevap : C

46-) RuntimeException sınıfı bir arabirim değil ve uygulanamıyor. Bu nedenle derlenmez. Cevap : D

48-) Belleğin bitmesi genellikle Java'da kurtarılamaz. Cevap : D

49-) Catch bloğunun içerisini FileNotFoundException **e1** yaptığımızda kod doğru çalışır. Output Failed olur. Cevap : C

50-) x1 satırında olan bir hatadan dolayı kod hata vermektedir. Cevap : B