

1-) Switch parantezleri içinde double, decimal tipinde değişken kullanılmaz.

-Değişkenin tipi string veya nümerik olabilir. Cevap : B

Kaynak : <https://www.mustafaakbal.com.tr/c-switch-case-komutu.html>

3-) Burada John string variable değeri iken Jon ise string nesnesidir. Kodu derlediğimde false true sonucu çıkar. Cevap : C

4-) Kod derlenme hatası verir. Çünkü ordaki else - if kullanımı yanlıştır. Kullanımı doğru yapıldığında ise plan=2; olduğundan plan B olacaktır. Cevap : D

5-) Default statement değer almaz . Cevap : C

6-)İşlem önceliğini gözeterek uyguladığımızda thatNumber= 4 çıkar. Cevap : B

7-)Break ifadesi, tüm kalan dalları atlar. Cevap : B

8-) Ternary expressions genellikle if-then-else deyimlerinin yerine kullanılır. Cevap : C

Kaynak : <https://medium.com/@jraleman/ternary-operators-vs-if-else-statements-6c26f7d034f7>

9-) && işlemi boole ifadelerine uygulanır. candidateA ve candidateB sayıdır. Bu yüzden 4.satır derlenemez. Cevap : C

10-) 6%3 ifadesinden kalan olmadığı için if ifadesi sağlamaz. triceratops ++ ifadesi çağrılmaz.

triceratops-- yürütülür ve sonuç 2 elde edilir. Cevap : A

11-) if-then ifadesi tek bir ifade veya kod bloğu {} yürütebilir. Cevap : D

12-) Output Not enoughToo many olur . Cevap : D

13-) Case statement break statement ile sonlandırılabilir ama zorunlu değildir. Cevap : B

14-) Boolean değerleri ile kullanıldıklarında, && her iki operand true ise true ; aksi halde, false döndürür. Tabloyu && operatörü sağlar. Cevap : D

Kaynak :

https://developer.mozilla.org/tr/docs/Web/JavaScript/Reference/Operat%C3%B6rler/Mantıksal_Operatörler

15-) if () içerisindeki ifade boolean olmalı. Bu yüzden kod derlememez. Cevap : C

16-) Prefix operatörleri yeni değeri değerlendir ve değişkeni değiştirir. Postfix operatörleri değişkeni

değiştirir ama orijinal değeri döndürür. Cevap : B

17-) Outputa winner yazılacaktır. Long winner = lion+2*(tiger +lion) burada işlem önceliğine göre önce parantez içindeki tiger + lion (2+3)=5 olur . Bu 5 ,2 ile çarpılır. Ve en son lion(3) ile toplanır.3+2*5=13

Cevap : B

- 19-)Kod derlenmedi çünkü day ikinci ternary operations işlemde int yerine boole ifade olarak kullanılmalıydı. Cevap : D
- 20-) Eksik bir parantez olduğu için kod hata verir. Cevap : C
- 21-) İşlemleri yaparken soldan sağa doğru gitmeliyiz. İlk iki değer sayı olduğu için toplama gibi değerlendirilir ve 11 bulunur. Sonrasında “7” string ifadesiyle birleştirip “117” gibi olduğunu düşünmeliyiz. Sonrasında gelen iki ifade de string ile birleşeceğinden string gibi düşünülür. Bu yüzden 11789 olur. Cevap : B
- 22-) İki sayı arasındaki farkı bulmak için çıkarma – operatörü kullanılır. Bir sayı bölündüğünde kalanı bulmak için ise % operatörü kullanılır. Cevap : B
- 23-) int partA=3 olur. int partB=2 olur. int newDog=2+3*3=11 olur. Cevap : B
- 24-) eaten değeri önce 1 sonra 2 arttırıldı. Daha sonra 1 çıkarıldı. Output da 2 olur. Cevap : B,
- 25-) Ternary operatörlerinde her iki ifadeninde uyumlu veri türlerinde olması gerekir. Burda ise ilk string ve int türlerinde verilmiş. Bu yüzden kod derlenemez. Cevap : C
- 26-) İki String nesnesi == kullanarak true olarak değerlendirilirse aynı nesnedir. Aynı string nesnesi equals() şeklinde geri döner. Cevap : A
- 27-) myTestVariable öğesinin null olmadığı verildiği için ifade false olarak değerlendirilir. Cevap : B
- 28-) (sokaklar && kavşaklar> 1000) geçersiz bir ifade . Çünkü sokaklar bir boole ifade değil. Kod derlenemez. Cevap : D
- 29-) && operatörü varken ifadenin her iki tarafını da her zaman değerlendirir. Yalnızca sol tarafın doğru olduğu belirlenirse ifadenin sağ tarafını değerlendirir. Cevap : B
- 30-) Kodu derlediğimizde 11 5 sonucunu elde ederiz. Cevap : C
- 31-) String bob ile notBob “bob” a karşılık gelmektedir. Birbirlerine eşittirler. Bu yüzden true true olur. Cevap : A
- 32-) * ve ½ operatörlerinin önceliği aynıdır. $12+6*\%3\ 2=12+\%18\ 2=12+0=12$ Cevap : B
- 33-) XOR ^ operatörü, p ve q farklıysa true, aynı ise false olur. Cevap : C

35-) /,*,% operatörlerinin öncelik sırası aynıdır. Cevap : C

36-) ^ yalnızca Java'daki boole değerlerine uygulanır. Cevap : D

37-) Şekildeki küme X ve Y kümelerinin birleşimidir. Bunu da $x \cup y$ sağlar. Cevap : C

39-) 5.21e eşit büyük olması için \geq , 8.1den küçük olması için $<$ kullanılır. Cevap : C

40-) Turtle=30 olur. Turtle Hare den daha az olmadığından, son ifade Turtle kazanır!

bu da konsola gönderilir. Cevap : B

42-) Kodu derlediğimde output up oldu ama tam olarak yorumlayamadım. Cevap : A

43-) $||$ işlenenlerden biri doğru olduğunda operatör doğrudur. logical complement(!) operatörü bir boole değerini tersine çevirir veya döndürür. Cevap : D

44-) characters ≤ 4 ifadesi yanlış, bu nedenle ikinci üçlü işlem yürütülür. story > 1 ifadesi true. Bu yüzden movieRating 'in son değeri 2.0 olur. Cevap : A

45-) Herhangi bir JVM sınırlaması için, bir switch case'in herhangi bir sayıda case statement olabilir. Cevap : B

Kaynak : <https://performance.netbeans.org/howto/jvmswitches/>

47-)(!) logical complement operatörü sadece boole ifadelere uygulanabilir. Bu yüzden işlem yapılamayacaktır. Cevap : D

48-) $||$ işleminde işlenenlerden biri true ise sonuç true olur. Sonuç sadece işlenenlerden ikisi de false ise false olur. Bu yüzden iki ifade de true olur. Cevap : C

49-) çıkarma -, toplama +, bölme / ve çarpma * operatörleridir. Cevap : A

50-) P1 satırındaki ifadeler int değişkenlerdir. Ama atama yapılmaya .alşılan game String olarak verilmiştir. Bu yüzden derlenemez. Hata verir. Cevap : C

