user:Ahsan Ahsan Habib - 1144421

test:**R-18 Core Java mock test 2**  
R-18 Core Java mock test 2

start time:2013-08-20 03:38:27

end time:2013-08-20 04:28:25

test time:00:49:58

points:25.333 / 47.000 (54%)

correct:25 / 47 (53%)

comment:

1. **[0.000] (IP:281473913979142 | 03:28:27 | 04:28:11 | 59:44 | 4.248)**   
   What type of loop is implemented with a do statement?
   1. x   top-driven loop
   2. ® bottom-driven loop
   3. while loop
   4. off-by-one loop
2. **[0.000] (IP:281473913979142 | 03:29:06 | 03:30:46 | 01:40 | 100.346)**   
   Given:  
   11. int x = 3;  
   12. int y = 1;  
   13. if (x = y) {  
   14. System.out.println(“x = “ + x);  
   15. }  
   What is the result?
   1. The code runs with no output.
   2. x = 3
   3. ® Compilation fails.
   4. An exception is thrown at runtime.
   5. x   x = 1
3. **[1.000] (IP:281473913979142 | 03:30:46 | 03:31:00 | 00:14 | 13.227)**   
   The statement i++; is equivalent to
   1. x ® i = i + 1;
   2. i = i - 1;
   3. i = i + i;
   4. i - - ;
4. **[1.000] (IP:281473913979142 | 03:31:00 | 03:33:48 | 02:48 | 168.063)**   
   Given:  
   11. for (int i =0; i <3; i++) {  
   12. switch(i) {  
   13. case 0: break;  
   14. case 1: System.out.print(“one “);  
   15. case 2: System.out.print(“two “);  
   16. case 3: System.out.print(“three “);  
   17. }  
   18. }  
   19. System.out.println(“done”);  
   What is the result?
   1. x ® one two three two three done
   2. one two three done
   3. one two done
   4. Compilation fails.
   5. done
5. **[0.000] (IP: 281473913979142 | 03:33:48 | 03:36:52 | 03:04 | 184.635)**   
   Given:  
   11. int i = 0, j = 1;  
   12. if ((i++ == 1) && (j++ == 2)) {  
   13. i = 42;  
   14 }  
   15. System.out.println(“i = “ + i + “, j = “ + j);  
   What is the result?
   1. i = 1, j = 2
   2. ® i = 1, j = 1
   3. Compilation fails.
   4. i = 42, j = 1
   5. x   i = 42, j = 2
6. **[1.000] (IP:281473913979142 | 03:36:52 | 03:38:17 | 01:25 | 84.803)**   
   for (int i = 0; i <= 3;) {  
   System.out.println("i = " + i);  
   }
   1. The code does not run
   2. i = 0  
      i = 1  
      i = 2  
      i = 3
   3. x ® i = 0 infinitely
   4. The code does not compile
7. **[1.000] (IP:281473913979142 | 03:38:17 | 03:39:26 | 01:09 | 68.803)**   
   What does the following print?  
     
   int count = 10;  
   do  
   {  
   System.out.print( count +" ");  
   count-- ;   
   }  
   while ( count >= 5 );
   1. 9 8 7 6 5
   2. x ® 10 9 8 7 6 5
   3. 9 8 7 6 5 4
   4. 10 9 8 7 6 5 4
8. **[1.000] (IP:281473913979142 | 03:39:26 | 03:40:49 | 01:23 | 82.598)**   
   Examine the following code fragment:  
     
   int j = 1;  
   do  
   {  
   System.out.println( j );  
   j++ ;  
   }  
   while ( j <= 3 );   
     
   Which of the following for loops does the same thing?
   1. for ( int j=0; j < 4; j++ )  
      System.out.println( j );
   2. for ( int j=1; j < 3; j++ )  
      System.out.println( j );
   3. for ( int j=0; j <= 3; j++ )  
      System.out.println( j );
   4. x ® for ( int j=1; j <= 3; j++ )  
      System.out.println( j );
9. **[1.000] (IP:281473913979142 | 03:40:49 | 03:43:16 | 02:27 | 146.662)**   
   Examine the following code:   
   int count = 1;   
   while ( \_\_\_\_\_\_\_\_\_\_\_ )   
   {  
   System.out.print( count + " " );  
   count = count + 1;  
   }  
   System.out.println( );  
   What condition should be used so that the code writes out:   
   1 2 3 4 5 6 7 8
   1. count+1 <= 8
   2. x ® count < 9
   3. count != 8
   4. count < 8
10. **[1.000] (IP:281473913979142 | 03:43:16 | 03:44:12 | 00:56 | 55.891)**   
    What value is placed in var?  
      
    var = 12 > 9 ? 0 : 1;
    1. 12
    2. x ® 0
    3. 1
    4. 9
11. **[1.000] (IP:281473913979142 | 03:44:12 | 03:45:09 | 00:57 | 57.173)**   
    In a group of nested loops, which loop is executed the most number of times?
    1. cannot be determined without knowing the size of the loops
    2. x ® the innermost loop
    3. the outermost loop
    4. all loops are executed the same number of times
12. **[0.000] (IP:281473913979142 | 03:45:09 | 03:46:30 | 01:21 | 80.439)**   
    A continue statement causes execution to skip to
    1. the end of the program
    2. ® the next iteration of the loop
    3. x   the statement following the continue statement
    4. the first statement after the loop
13. **[1.000] (IP:281473913979142 | 03:46:30 | 03:46:54 | 00:24 | 24.519)**   
    What's wrong? for (int k = 2, k <= 12, k++)
    1. there should be a semicolon at the end of the statement
    2. x ® the commas should be semicolons
    3. the increment should always be ++k
    4. the variable must always be the letter i when using a for loop
14. **[1.000] (IP:281473913979142 | 03:46:54 | 03:47:58 | 01:04 | 63.376)**   
    What value is placed in choice?  
      
    int a=5, b=10, c=15 ;  
      
    choice = a>b && a>c ? a : (b > c ? b : c) ;
    1. x ® 15
    2. 5
    3. 0
    4. 10
15. **[0.000] (IP:281473913979142 | 03:47:58 | 03:48:34 | 00:36 | 36.13)**   
    What does the following print?  
      
    int count = 0;  
    do  
    {  
    System.out.print( count +" ");  
    count++ ;   
    }  
    while ( count < 6 );
    1. x   0 1 2 3 4 5 6
    2. ® 0 1 2 3 4 5
    3. 1 2 3 4 5
    4. 1 2 3 4 5 6
16. **[0.000] (IP:281473913979142 | 03:48:34 | 03:49:29 | 00:55 | 54.754)**   
    Given:  
    11. Float f = new Float(“12”);  
    12. switch (f) {  
    13. case 12: System.out.println(“Twelve”);  
    14. case 0: System.out.println(“Zero”);  
    15. default: System.out.println(“Default”);  
    16. }  
    What is the result?
    1. Twelve
    2. x   Twelve  
       Zero  
       Default
    3. ® Compilation fails.
    4. Zero
    5. Default
17. **[1.000] (IP:281473913979142 | 03:49:29 | 03:49:47 | 00:18 | 17.78)**   
    Which looping process checks the test condition at the end of the loop?
    1. for
    2. x ® do-while
    3. while
    4. no looping process checks the test condition at the end
18. **[0.000] (IP:0 | 03:49:47 | --:--:-- | --:-- | ------ )**   
    Given:  
    11. int i = 1,j = 10;  
    12. do{  
    13. if (i>j) {  
    14. continue;  
    15. }  
    16. j--;  
    17. } while (++i <6);  
    18. System.out.println(“i = “ +i+” and j = “+j);  
    What is the result?
    1. ® i = 6 and j = 5
    2. i = 5 and j = 5
    3. i = 5 and j = 6
    4. i = 6 and j = 4
    5. i = 6 and j = 6
19. **[1.000] (IP:281473913979142 | 03:53:27 | 03:54:11 | 00:44 | 44.7)**   
    Examine the following code:  
      
    int count = -2 ;   
    while ( count < 3 )   
    {  
    System.out.print( count + " " );  
    count = count + 1;   
    }  
    System.out.println( );  
      
    What does this code print on the monitor?
    1. -2 -1 1 2 3 4
    2. -2 -1 1 2 3
    3. x ® -2 -1 0 1 2
    4. -3 -4 -5 -6 -7
20. **[1.000] (IP:281473913979142 | 03:54:12 | 03:55:28 | 01:16 | 76.201)**   
    What value is assigned to discount ?  
      
    double discount;  
    char code = 'C' ;  
      
    switch ( code )  
    {  
    case 'A':  
    discount = 0.0;  
    break;  
      
    case 'B':  
    discount = 0.1;  
    break;  
      
    case 'C':  
    discount = 0.2;  
    break;  
      
    default:  
    discount = 0.3;  
    }
    1. 0.3
    2. 0.0
    3. 0.1
    4. x ® 0.2
21. **[1.000] (IP: 281473913979142 | 03:55:28 | 03:57:43 | 02:15 | 135.094)**   
    What will be the result of compiling and running the given program?  
    Select one correct answer.  
    1 class Q1  
    2 {  
    3 public static void main(String arg[])  
    4 {  
    5 int a[]={2,2};  
    6 int b=1;  
    7 a[b]=b=0;  
    8 System.out.println(a[0]);  
    9 System.out.println(a[1]);  
    10 }  
    11 }
    1. Program compiles correctly and print 0,2 when executed.
    2. Run time error at the line no. 5.
    3. Compile time error at the line no. 5.
    4. x ® Program compiles correctly and print 2,0 when executed.
22. **[0.000] (IP:281473913979142 | 03:57:43 | 03:59:32 | 01:49 | 109.177)**   
    Given:  
    11. boolean bool = true;  
    12. if(bool = false) {  
    13. System.out.println(“a”);  
    14. } else if (bool) {  
    15. System.out.println(“c”);  
    16. } else if (!bool) {  
    17. System.out.println(“c”);  
    18. } else {  
    19. System.out.println(“d”);  
    20. }  
    What is the result?
    1. b
    2. ® Compilation fails.
    3. a
    4. d
    5. x   c
23. **[0.000] (IP:281473913979142 | 03:59:32 | 04:00:51 | 01:19 | 78.139)**   
    What fact about a do loop is responsible for many program bugs?
    1. ® The body of a do loop is always executed at least once.
    2. x   The do is not a good choice for a counting loop.
    3. The do must be matched with a while.
    4. Using a do loop sometimes shortens a program.
24. **[1.000] (IP:281473913979142 | 04:00:51 | 04:02:16 | 01:25 | 84.969)**   
    What value is placed in awk?  
      
    int x = 5, y = 19;  
      
    awk = y-x > x-y ? y-x : x-y ;
    1. x ® 14
    2. 5
    3. 19
    4. -14
25. **[0.000] (IP:281473913979142 | 04:02:16 | 04:04:09 | 01:53 | 113.509)**   
    Given:  
    11. int i =1,j =10;  
    12. do {  
    13. if(i++> --j) {  
    14. continue;  
    15. }  
    16. } while (i <5);  
    17. System.out.println(“i = “ +i+ “and j = “+j);  
    What is the result?
    1. i = 6 and j = 4
    2. ® i = 5 and j = 6
    3. i = 5 and j = 5
    4. x   i = 6 and j = 5
    5. i = 6 and j = 6
26. **[1.000] (IP:281473913979142 | 04:04:09 | 04:07:47 | 03:38 | 217.24)**   
    What does the following print?  
      
    int row = 1;  
    do  
    {  
    int col = 1;  
    do  
    {  
    System.out.print( "\*" );  
    col++ ;   
    }  
    while ( col <= row );  
      
    System.out.println();  
    row++ ;  
    }  
    while ( row <= 3 );
    1. \*\*\*\*\*\*  
       \*\*\*\*\*  
       \*\*\*\*  
       \*\*\*  
       \*\*  
       \*
    2. x ® \*  
       \*\*  
       \*\*\*
    3. \*\*\*\*  
       \*\*\*\*  
       \*\*\*\*
    4. \*\*\*  
       \*\*  
       \*
27. **[0.000] (IP:281473913979142 | 04:07:47 | 04:08:30 | 00:43 | 43.353)**   
    What's wrong? while( (i < 10) && (i > 24))
    1. the while loop is an exit-condition loop
    2. x   the test condition is always true
    3. the logical operator && cannot be used in a test condition
    4. ® the test condition is always false
28. **[1.000] (IP:281473913979142 | 04:08:30 | 04:10:26 | 01:56 | 115.941)**   
    Given:  
    10. int i = 0;  
    11. for (; i <4; i += 2) {  
    12. System.out.print(i + “”);  
    13. }  
    14. System.out.println(i);  
    What is the result?
    1. Compilation fails.
    2. 0 2 4 5
    3. An exception is thrown at runtime.
    4. x ® 0 2 4
    5. 0 1 2 3 4
29. **[0.000] (IP:281473913979142 | 04:10:26 | 04:10:44 | 00:18 | 17.232)**   
    What are the branching statements in a programming language?
    1. x   Statements that evaluate boolean expressions.
    2. Statements that affect the execution of loops.
    3. Statements that are used to build classes.
    4. ® Statements like if that make choices.
30. **[1.000] (IP:281473913979142 | 04:10:44 | 04:11:08 | 00:24 | 24.575)**   
    Examine the following code  
    int count = 0;   
    while ( count <= 6 )   
    {  
    System.out.print( count + " " );  
    count = count + 2;   
    }  
    System.out.println( );  
    What does this code print on the monitor?
    1. 0 2 4 6 8
    2. 1 2 3 4 5 6
    3. 0 2 4
    4. x ® 0 2 4 6
31. **[0.000] (IP:281473913979142 | 04:11:08 | 04:11:32 | 00:24 | 23.576)**   
    What value is assigned to discount ?  
      
    double discount;  
    char code = 'C' ;  
      
    switch ( code )  
    {  
    case 'A':  
    discount = 0.0;  
      
    case 'B':  
    discount = 0.1;  
      
    case 'C':  
    discount = 0.2;  
      
    default:  
    discount = 0.3;  
    }
    1. x   0.2
    2. 0.1
    3. 0.0
    4. ® 0.3
32. **[0.000] (IP:281473913979142 | 04:11:32 | 04:11:50 | 00:18 | 18.107)**   
    What value is stored in num at the end of this looping?  
    for (num = 1; num <= 5; num++)
    1. 1
    2. 4
    3. ® 6
    4. x   5
33. **[1.000] (IP:281473913979142 | 04:11:50 | 04:12:43 | 00:53 | 52.078)**   
    Which statement makes sure that x is an even number?
    1. x = x%2 == 0 ? x+1 : x;
    2. x += 2\*x ;
    3. x = x%2 == 1 ? x++ : x;
    4. x ® x += x%2 == 0 ? 0 : 1 ;
34. **[0.333] (IP:281473913979142 | 04:12:43 | 04:16:37 | 03:54 | 233.968)**   
    Which statements about the output of the following program are true?  
    public class Logic {  
    public static void main(String args[]) {  
    int i = 0;  
    int j = 0;  
      
    boolean t = true;  
    boolean r;  
      
    r = (t && 0<(i+=1));  
    r = (t && 0<(i+=2));  
    r = (t && 0<(j+=1));  
    r = (t || 0<(j+=2));  
      
    System.out.println( i + “ ” + j );  
    }  
    }
    1. The second digit printed is 2.
    2. ® The first digit printed is 3.
    3. x   The second digit printed is 3.
    4. x   The first digit printed is 1.
    5. ® The second digit printed is 1.
    6. The first digit printed is 2.
35. **[0.000] (IP:281473913979142 | 04:16:37 | 04:28:25 | 11:48 | 6.548)**   
    Given:  
    11. int i = 1,j = 10;  
    12. do {  
    13. if(i>j) {  
    14. break;  
    15. }  
    16. j--;  
    17. } while (++i <5);  
    18. System.out.println(“i =” +i+” and j = “+j);  
    What is the result?
    1. i = 5 and j = 5
    2. i = 6 and j = 6
    3. x   i = 6 and j = 5
    4. i = 6 and j = 4
    5. ® i = 5 and j = 6
36. **[0.000] (IP:281473913979142 | 04:17:00 | 04:17:55 | 00:55 | 55.54)**   
    Is the do statement a necessary feature in Java?
    1. ® No--everything it does could be done with a while.
    2. x   Yes--some loops can only be implemented with a do.
    3. No--but it would be exremely difficult without it.
    4. Yes--without it one of the major control structures would be lost.
37. **[1.000] (IP:281473913979142 | 04:17:55 | 04:19:32 | 01:37 | 96.711)**   
    What does the following print?  
      
    int row = 1;  
    do  
    {  
    int col = 1;  
    do  
    {  
    System.out.print( "\*" );  
    col++ ;   
    }  
    while ( col <= 5 );  
      
    System.out.println();  
    row++ ;  
    }  
    while ( row <= 3 );
    1. \*\*\*  
       \*\*\*  
       \*\*\*  
       \*\*\*  
       \*\*\*
    2. x ® \*\*\*\*\*  
       \*\*\*\*\*  
       \*\*\*\*\*
    3. \*\*\*\*\*\*  
       \*\*\*\*\*\*  
       \*\*\*\*\*\*
    4. \*\*\*  
       \*\*\*  
       \*\*\*  
       \*\*\*
38. **[1.000] (IP:281473913979142 | 04:19:32 | 04:19:55 | 00:23 | 23.28)**   
    Each pass through a loop is called a/an
    1. enumeration
    2. pass through
    3. culmination
    4. x ® iteration
39. **[1.000] (IP:281473913979142 | 04:19:55 | 04:20:11 | 00:16 | 15.202)**   
    Which looping process is best used when the number of iterations is known?
    1. do-while
    2. x ® for
    3. while
    4. all looping processes require that the iterations be known
40. **[1.000] (IP:281473913979142 | 04:20:11 | 04:20:53 | 00:42 | 41.722)**   
    Examine the following code:   
    int count = 7;   
    while ( count >= 4 )   
    {  
    System.out.print( count + " " );  
    count = count - 1;   
    }  
    System.out.println( );  
    What does this code print on the monitor?
    1. 6 5 4 3
    2. 7 6 5 4 3
    3. 1 2 3 4 5 6 7
    4. x ® 7 6 5 4
41. **[1.000] (IP:281473913979142 | 04:20:53 | 04:21:35 | 00:42 | 42.886)**   
    What does the following print?  
      
    int count = 10;  
    do  
    {  
    System.out.print( count +" ");  
    count++ ;   
    }  
    while ( count < 6 );
    1. 6
    2. x ® 10
    3. It prints nothing
    4. 5
42. **[0.000] (IP:281473913979142 | 04:21:36 | 04:22:23 | 00:47 | 47.239)**   
    Given:  
    11. int i = 1,j = -1;  
    12. switch (i) {  
    13. case 0, 1:j = 1;  
    14. case 2: j = 2;  
    15. default; j = 0;  
    16. }  
    17. System.out.println(“j=”+j);  
    What is the result?
    1. j = -1
    2. ® Compilation fails.
    3. x   j = 0
    4. j = 2
    5. j = 1
43. **[0.000] (IP:281473913979142 | 04:22:23 | 04:24:31 | 02:08 | 26.216)**   
    What value is placed in sum?  
      
    double sum = 10.0, price=100;  
      
    sum += price>=100 ? price\*1.1 : price;
    1. x   110
    2. 90
    3. 100
    4. ® 120
44. **[0.000] (IP:281473913979142 | 04:24:31 | 04:24:51 | 00:20 | 19.379)**   
    Given:  
    11. for (int i =0; i < 4; i +=2) {  
    12. System.out.print(i + “”);  
    13. }  
    14. System.out.println(i);  
    What is the result?
    1. 0 1 2 3 4
    2. ® Compilation fails.
    3. An exception is thrown at runtime.
    4. 0 2 4 5
    5. x   0 2 4
45. **[0.000] (IP:281473913979142 | 04:24:51 | 04:25:33 | 00:42 | 42.507)**   
    Which statements about the output of the following program are true?  
    public class EqualTest {  
    public static void main(String args[]) {  
    String s1 = “YES”;  
    String s2 = “YES”;  
    if ( s1 == s2 ) System.out.println(“equal”);  
    String s3 = new String(“YES”);  
    String s4 = new String(“YES”);  
    if ( s3 == s4 ) System.out.println(“s3 eq s4”);  
    }  
    }
    1. ® “equal” is printed only.
    2. Nothing is printed.
    3. x   “equal” is printed, “s3 eq s4” is printed.
    4. “s3 eq s4” is printed only.
46. Examine the following code:  
      
    int count = 1;   
    while ( count < 5 )   
    {  
    System.out.print( count + " " );  
    }  
    System.out.println( );  
      
    What does this code print on the monitor?
    1. x ® 1 1 1 1 1 1 1 1 1 1 1 . . . .
    2. 2 3 4
    3. 1 2 3 4 5
    4. 1 2 3 4
47. Given:  
    1. public class SwitchTest {  
    2. public static void main(String[] args) {  
    3. System.out.println(“value = “ + switchIt(4));  
    4. }  
    5. public static int switchIt(int x) {  
    6. int j = 1;  
    7. switch (x) {  
    8. case 1: j++;  
    9. case 2: j++;  
    10. case 3: j++;  
    11. case 4: j++;  
    12. case 5: j++;  
    13. default: j++;  
    14. }  
    15. return j + x;  
    16. }  
    17. }  
    What is the result?
    1. x   value = 6
    2. ® value = 8
    3. value = 7
    4. value = 4
    5. value = 5
    6. value = 3