

The next two questions refer to this code segment

```
if (a != b && n / (a - b) > 90)
{
    /* statement 1 */
}
else
{
    /* statement 2 */
}
```

1. What will happen if `a == b` is true?
 - a) `/* statement 2 */` will be executed
 - b) `/* statement 1 */` will be executed
 - c) Neither `/* statement 1 */` nor `/* statement 2 */` will be executed.
 - d) A compile-time error will occur.
 - e) An exception will be thrown.
2. What will happen if `a == b` is false?
 - a) `/* statement 1 */` will be executed
 - b) `/* statement 2 */` will be executed
 - c) Either `/* statement 1 */` or `/* statement 2 */` will be executed.
 - d) A compile-time error will occur.
 - e) An exception will be thrown.
3. Look at the following poorly formatted program segment. If `a = 7` and `c = 6` before execution, which of the following represents the correct values of `c`, `d`, `p`, and `t` after execution? An undetermined value is represented with a question mark.

```
if (a == 6)
if (c == 6)
{
    c = 9;
    d = 9;
}
else
{
    t = 10;
    if (c == 6)
        c = 5;
}
else p = 9;
```

- a) `c = 6`, `d = ?`, `p = 9`, `t = ?`
- b) `c = 5`, `d = ?`, `p = ?`, `t = 10`
- c) `c = 6`, `d = ?`, `p = ?`, `t = ?`
- d) `c = 5`, `d = 9`, `p = ?`, `t = 10`
- e) `c = 9`, `d = 9`, `p = ?`, `t = ?`

4. Consider the following outline of a nested if-else structure which has more if clauses than else clauses. Which of the statements below is true regarding this structure?

```
if (condition1)
    if (condition2)
        statement1;
    else statement2;
```

- a) syntactically it is invalid to have more if clauses than else clauses
 - b) statement2 will only execute if condition1 is false and condition2 is false
 - c) statement2 will only execute if condition1 is true and condition2 is false
 - d) statement2 will only execute if condition1 is false, it does not matter what condition2 is
 - e) statement2 will never execute
5. What values are stored in x and y after execution of the following program segment?

```
int x = 30, y = 40;
if (x >= 0)
{
    if (x <= 100)
    {
        y = x * 3;
        if (y < 50)
            x /= 10;
    }
    else
        y = x * 2;
}
else
    y = -x;
```

- a) x = 30 y = 90
- b) x = 30 y = -30
- c) x = 30 y = 60
- d) x = 3 y = -3
- e) x = 30 y = 40

6. Of the following if statements, which one correctly executes three instructions if the condition is true, and executes no instructions if it is false?

a) `if (x < 0)`
 `a = b * 2;`
 `y = x;`
 `z = a - y;`

b) `{`
 `if (x < 0)`
 `a = b * 2;`
 `y = x;`
 `z = a - y;`
`}`

c) `if{ (x < 0)`
 `a = b * 2;`
 `y = x;`
 `z = a - y;`
`}`

d) `if (x < 0) {`
 `a = b * 2;`
 `y = x;`
 `z = a - y;`
`}`

e) b, c and d are all correct, but not a

7. Consider the following code that will assign a letter grade of A, B, C, D, or F depending on a student's test score.

```
if (score >= 90)
    grade = "A";
if (score >= 80)
    grade = "B";
if (score >= 70)
    grade = "C";
if (score >= 60)
    grade = "D";
else
    grade = 'F';
```

- a) This code will work correctly in all cases
- b) This code will work correctly only if `grade >= 60`
- c) This code will work correctly only if `grade < 60`
- d) This code will work correctly only if `grade < 70`
- e) This code will not work correctly under any circumstances

8. What is wrong, logically, with the following code?

```
if (x > 10)
    System.out.println("Large");
else if (x > 6 && x <= 10)
    System.out.println("Medium");
else if (x > 3 && x <= 6)
    System.out.println("Small");
else
    System.out.println("Very small");
```

- a) There is no logic error, but there is no need to have $(x \leq 10)$ in the second conditional or $(x \leq 6)$ in the third conditional
- b) There is no logic error, but there is no need to have $(x > 6)$ in the second conditional or $(x > 3)$ in the third conditional
- c) The logic error is that no matter what value x is, "Very small" is always printed out
- d) The logic error is that no matter what value x is, "Large" is always printed out
- e) There is nothing wrong with the logic at all

9. Given that a , b , and c are integers, consider the boolean expression

$(a < b) \ || \ !((c == a * b) \ \&\& \ (c < a))$

Which of the following will *guarantee* that the expression is true?

- a) $c < a$ is false.
- b) $c < a$ is true.
- c) $a < b$ is false.
- d) $c == a * b$ is true.
- e) $c == a * b$ is true, and $c < a$ is true.

10. Which of the following will evaluate to true only if boolean expressions A , B , and C are all false?

- a) $!A \ \&\& \ !(B \ \&\& \ !C)$
- b) $!A \ || \ !B \ || \ !C$
- c) $!(A \ || \ B \ || \ C)$
- d) $!(A \ \&\& \ B \ \&\& \ C)$
- e) $!A \ || \ !(B \ || \ !C)$

11. Assume that `x` and `y` are `int` variables with `x = 5`, `y = 3`, and `a` and `d` are `String` variables with `a = "a"` and `d = "A"`, and examine the following conditions:

Condition 1: `(x < y && x > 0)`

Condition 2: `(!a.equals(d) || x != 5)`

Condition 3: `!(true && false)`

Condition 4: `(x > y || a.equals("A") || !d.equals("A"))`

- a) All 4 Conditions are true
- b) Only Condition 2 is true
- c) Condition 2 and Condition 4 are true only
- d) Conditions 2, 3 and 4 are all true, Condition 1 is not
- e) All 4 Conditions are false

12. Assume that `x` and `y` have been declared and initialized with `int` values. Consider the following Java expression

```
(y > 10000) || (x > 1000 && x < 1500)
```

Which of the following is equivalent to the expression given above?

- a) `(y > 10000 || x > 1000) && (y > 10000 || x < 1500)`
- b) `(y > 10000 || x > 1000) || (y > 10000 || x < 1500)`
- c) `(y > 10000) && (x > 1000 || x < 1500)`
- d) `(y > 10000 && x > 1000) || (y > 10000 && x < 1500)`
- e) `(y > 10000 && x > 1000) && (y > 10000 && x < 1500)`

13. Consider the following code segment

```
int newNum = 0, temp;
int num = k;
while (num > 10)
{
    temp = num % 10;
    num /= 10;
    newNum = newNum * 10 + temp;
}
System.out.print(newNum);
```

Which is a true statement about the segment?

I If $100 \leq \text{num} \leq 1000$ initially, the final value of `newNum` must be in the range $10 \leq \text{num} \leq 100$

II There is no initial value of `num` that will cause an infinite `while` loop.

III If $\text{num} \leq 10$ initially, `newNum` will have a final value of 0.

- a) I only
- b) II only
- c) III only
- d) II and III only
- e) I, II, and III

14. Consider the following loop, where `n` is some positive integer.

```
for (int i = 0; i < n; i += 2)
{
    if (/* test */)
        /* perform some action */
}
```

In terms of `n`, which Java expression represents the maximum number of times that `/* perform some action */` could be executed?

- a) $n / 2$
- b) $(n + 1) / 2$
- c) n
- d) $n - 1$
- e) $(n - 1) / 2$

15. If `x` is an `int` where `x = 0`, what will `x` be after the following loop terminates?

```
while (x < 100)
    x *= 2;
```

- a) 2
- b) 64
- c) 100
- d) 128
- e) None of the above, this is an infinite loop

16. How many times will the following loop iterate?

```
int x = 10;
while (x > 0) {
    System.out.println(x);
    x--;
```

- a) 0 times
- b) 1 time
- c) 9 times
- d) 10 times
- e) 11 times

The next two questions refer to the following code segment:

```
int n = Keyboard.readInt();
int x = 1;
int y = 1;

// Point A

while (n > 2)
{
    x = x + y;

    // Point B

    y = x - y;
    n--;
}

// Point C

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

17. What is printed if the user types in a 6 for `n`?

- a) 1
- b) 5
- c) 6
- d) 8
- e) 13

18. What is true about the code?

- a) `x` will sometimes be 1 at // Point B
- b) `x` will never be 1 at // Point C
- c) `n` will never be greater than 2 at // Point A
- d) `n` will sometimes be greater than 2 at // Point C
- e) `n` will always be greater than 2 at // Point B

19. Given two `String` variables, `s1` and `s2`, to determine if they are the same length, which of the following conditions would you use?

- a) `(s1.equals(s2))`
- b) `(s1.length() .equals(s2))`
- c) `(s1.length() .equals(s2.length()))`
- d) `(s1.length() == s2.length())`
- e) `length(s1) == length(s2)`

20. Given a `String`, `s`, which is assumed to have at least one character in it, which of the following conditions would determine if the first character of the `String` is the same as the last character?

- a) `(s.substring(0,1).equals(s.substring(s.length(),s.length()+1)))`
- b) `(s.substring(1,2).equals(s.substring(s.length(), s.length()+1)))`
- c) `(s.substring(0,1).equals(s.substring(s.length()-1, s.length())))`
- d) `(s.substring(0,1).equals(s.substring(s.length()+1, s.length()+2)))`
- e) `(s.substring(0,1).equals(s.substring(last)))`

21. Consider the following code segment

```
int x = 1;
while ( /* missing code */ )
{
    System.out.print(x + " ");
    x = x + 2;
}
```

Consider the following possible replacements for `/* missing code */`

- I. `x < 6`
- II. `x != 6`
- III. `x < 7`

Which of the proposed replacements for `/* missing code */` will cause the code segment to print only the values 1 3 5?

- a) I only
- b) II only
- c) I and II only
- d) I and III only
- e) I, II, and III

22. Consider the following code segment

```
int x = 10, y = 0;
while (x > 5)
{
    y = 3;
    while (y < x)
    {
        y *= 2;
        if (y % x == 1)
            y += x;
    }
    x -= 3;
}
System.out.println(x + " " + y);
```

What will be output after execution of this code segment?

- a) 1 6
- b) 7 12
- c) -3 12
- d) 4 12
- e) -3 6

23. What will be output by this code segment?

```
for (int i = 5; i > 0; i--)  
{  
    for (int j = 1; j <= i; j++)  
        System.out.print(j*j + " ");  
    System.out.println();  
}
```

- a) 1
1 4
1 4 9
1 4 9 16
1 4 9 16 25
- b) 1 4 9 16 25
1 4 9 16
1 4 9
1 4
1
- c) 9 7 5 3 1
25 16 9 4
25 16 9
25 16
25
- d) 25
25 16
25 16 9
25 16 9 4
25 16 9 4 1
- e) 1 4 9 16 25
1 4 9 16 25
1 4 9 16 25
1 4 9 16 25
1 4 9 16 25

24. The following nested loop structure will execute the inner most statement (x++) how many times?

```
for(int j = 0; j < 100; j++)  
    for(int k = 100; k > 0; k--)  
        x++;
```

- a) 100
b) 200
c) 10,000
d) 20,000
e) 1,000,000