

### **Question 1-D**

The while loop has a condition that returns a boolean that controls the loop. Appears initially and is checked before going into the loop.

### **Question 2-B**

for loop is best known for using an index or counter.

### **Question 3-A**

do-while loop is guaranteed to have the body execute at least once.

### **Question 4-C**

for-each loop goes through each element, storing it in a variable.

### **Question 5- B**

The continue keyword is used to end the loop iteration immediately and resume execution at the next iteration

### **Question 6- A**

The break keyword is used to end the loop iteration immediately, skip any remaining executions of the loop, and resume execution immediately after the loop

### **Question 7-B**

for loop is having an initialization statement, condition statement, and update statement.

### **Question 8- C**

With for loop, we can loop through an array in ascending or descending order.

### **Question 9-A**

for-each loop can iterate through an array starting from index 0.

### **Question 10-A**

do-while loop has a condition that returns a boolean at the end of the loop.

### **Question 11-B**

“singer” variable is not a boolean. While loop requires a boolean condition.

### **Question 12-B**

It starts with the last index in the list and goes to the first index in the list.

### **Question 13-A**

The first time through the loop, the index is 0 and glass, is output. The break statement then skips all remaining executions on the loop and the main() method ends.

### Question 14-C

```
String letters = " ";  
while (letters.length() != 2)  
letters+="a";  
System.out.println(letters);           // print aa
```

### Question 15D

Since i never gets smaller, this code produces an infinite loop.

### Question 16-B

The break statement sends the execution to after the loop and 2 is output.

### Question 17-C

It is not in scope after the loop where it is referenced by the println().

### Question 18-D

for loop is allowed to have all three segments left blank. In fact, for( ; ; ) { } is an infinite loop.

### Question 19-C

It is not create an infinite loop using a for-each because it simply loop through an array or ArrayList. .

### Question 20-A

It starts with index 0 and goes to the last index in the list.

### Question 21-D

Parentheses are not allowed to surround a loop body though, so the code does not compile.

### Question 22-?

### Question 23-C

Break;/break letters; → the code flow follow the arrow in this diagram

### Question 24-B

The continue letters; statement does that.

### Question 25-C

That condition is false, so the loop body is never run. No output is produced.

### Question 26-C

for-each loop is allowed to be used with arrays and ArrayList objects. StringBuilder is not an allowed type for this loop.

### Question 27 –B

```
boolean balloonInflated = false;
do { if (!balloonInflated) {
    balloonInflated = true;
    System.out.print("inflate-"); }
} while (! balloonInflated);
System.out.println("done");           // inflate- done
```

### Question 28-D

The loop never ends . This is an infinite loop.

### Question 29-B

for loop, the segments are an initialization expression, a boolean conditional, and an update statement in that order.

### Question 30-B

The inner loop runs four times so count becomes 0. Then both loops end.

### Question 31-A

The boolean condition sees that 0 is not greater than 0. The outer loop ends and 1 is printed out.

### Question 32-D

for-each loop is only able to go through an array in ascending order. It is notable to control the order..

### Question 33-C

The break statement is not in the loop. Since break cannot be used at the top level of a method, the code does not compile.

### Question 34-C

Multiple update expressions are separated with a comma rather than a semicolon.

### Question 35-D

The program attempts to print args[3]. Since indexes are zero based in Java, it throws an ArrayIndexOutOfBoundsException.

### Question 36-B

Print runs after the loop, printing out shoelace once.

### Question 37-C

Line 27 refers to a loop label. While the label is still present, it no longer points to a loop. This causes the code to not compile.

### Question 38-C

Count increments for each element of the array. The code outputs 4.

### Question 39-C

The builder variable is aStringBuilder and not a boolean. So, the code does not compile  
\*\*do-while loop requires a boolean condition.

### Question 40-A

```
int count = 0;
do {
    do {
        count ;
    } while (count < 2);
    break;
} while (true);
System.out.println(count);           // print 2
```

### Question 41-C

break t; → it breaks out of both loops.

### Question 42-B

Since it checks the size of both arrays correctly, it prints the first two sets of elements.

### Question 43-B

Looping through the same list multiple times is allowed.

### Question 44-?

### Question 45-?

### Question 46-C

Option C goes through six iterations since the loop condition is at the end of the loop. Therefore it is not like the others.

**Question 47-D**

The loop condition keeps running with no opportunity for it to be set. Therefore, this is an infinite loop.

**Question 48-C**

The label of the loop is trying to use the keyword `for`. This is not allowed, so the code does not compile.

**Question 49-D**

This is an infinite loop. The `if` statement no longer runs, but the variable never changes state again, so the loop doesn't end.

**Question 50-B**

In a `for` loop, the type is only allowed to be specified once.