1.Which type of loop is best known for its boolean condition that controls entry to the loop?

A. do-while loop
B. for (traditional)
C. for-each
D. while

D. A while loop has a condition that returns a boolean that controls the loop. It appears at the beginning and is checked before entering the loop. Therefore, Option D is correct. A traditional for loop also has a boolean condition that is checked before entering the loop. However, it is best known for having a counter variable, making Option B incorrect. Option A is incorrect because the boolean condition on a do-while loop is at the end of the loop. Option C is incorrect because there is no condition as part of the loop construct.

2.Which type of loop is best known for using an index or counter?

A. do-while loop
B. for (traditional)
C. for-each
D. while

B. A traditional for loop is best known for having a loop variable counting up or down as the loop progresses. Therefore, Option B is correct. Options A and D are incorrect because do-while and while loops are known for their boolean conditions. Option C is incorrect because the for-each loop iterates through without an index.

3.Which type of loop is guaranteed to have the body execute at least once?

A. do-while loop
B. for (traditional)
C. for-each
D. while

A. A do-while loop checks the loop condition after execution of the loop body. This ensures it always executes at least once, and Option A is correct. Option B is incorrect because there are loops you can write that do not ever enter the loop body, such as for (int i=0;i<1;i++).

Similarly, Option D is incorrect because a while loop can be written where the initial loop condition is false. Option C is incorrect because a for-each loop does not enter the loop body when iterating over an empty list.

4.What keyword is used to end the current loop iteration and proceed execution with the next iteration of that loop?

A. break
B. continue
C. end
D. skip

B. The continue keyword is used to end the loop iteration immediately and resume execu- tion at the next iteration. Therefore, Option B is correct. Option A is incorrect because the break statement causes execution to proceed after the loop body. Options C and D are incorrect because these are not keywords in Java.

5.What does the following code output?

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

A. aa
B. aaa
C. The loops complete with no output.
D. This is an infinite loop.

A. Immediately after letters is initialized, the loop condition is checked. The variable letters is of length 0, which is not equal to 2 so the loop is entered. In the loop body, letters becomes length 1 with contents "a". The loop index is checked again and now 1 is not equal to 2. The loop is entered and letters becomes length 2 and contains "aa". Then the loop index is checked again. Since the length is now 2, the loop is completed and aa is output. Option A is correct.

6.What keyword is used to proceed with execution immediately after a loop?

A. break
B. continue
C. end
D. skip

A. The break keyword is used to end the loop iteration immediately, skip any remaining exe- cutions of the loop, and resume execution immediately after the loop. Therefore, Option A is correct. Option B is incorrect because execution proceeds at the next execution of the current loop for continue. Options C and D are incorrect because these are not keywords in Java.

7. What does the following code output?

```
int singer = 0;
while (singer)
System.out.println(singer++);
```

A. 0
B. The code does not compile.
C. The loops complete with no output.
D. This is an infinite loop.

B. A while loop requires a boolean condition. While singer is a variable, it is not a boolean. Therefore, the code does not compile, and Option B is correct.

8. What does the following code output?

```
int singer = 0;
while (singer > 0)
System.out.println(singer++);
```

A. 0
B. The code does not compile.
C. The loops completes with no output.
D. This is an infinite loop.

C. A while loop checks the boolean condition before entering the loop. In this code, that condition is false, so the loop body is never run. No output is produced, and Option C is correct.

9. What does the following code output?

```
String letters = "";
while (letters.length() != 3)
letters+="ab";
System.out.println(letters);
```

A. ab
B. abab
C. The loop completes with no output.
D. This is an infinite loop.

D.Immediately after letters is initialized, the loop condition is checked. The variable letters is of length 0, which is not equal to 3, so the loop is entered. In the loop body, letters becomes length 2 and contains "ab". The loop index is checked again and now 2 is not equal to 3. The loop is entered and letters becomes length 4 with contents "abab". Then the loop index is checked again. Since the length 4 is not equal to 3, the loop body is entered again. This repeats for 6, 8, 10, etc. The loop never ends, and Option D is correct.

10. What is the result of the following?

```
int k = 0;
for (int i = 10; i > 0; i--) {
while (i > 3)
{i -= 3;}
k += 1;
}
System.out.println(k);
```

A. 1
B. 2
C. 3
D. 4

A. On the first iteration of the outer loop, i starts out at 10. The inner loop sees that 10 > 3 and subtracts 3, making the 7 the new value of i. Since 7 > 3, we subtract 3 again, making i set to 4. Yet again 4 > 3, so i becomes 1. Then k is finally incremented to 1. The outer loop decrements i i, making it 0. The boolean condition sees that 0 is not greater than 0. The outer loop ends and 1 is printed out. Therefore, Option A is correct.

11.Which of the following is equivalent to this code snippet given an array of String objects?

```
for (int i=fun.length-1; i>=0; i--)
System.out.println(fun[i]);
```

A. for (String f = fun) System.out.println(f);

B. for (String f : fun) System.out.println(f);

C. for (String f fun) System.out.println(it);

D. None of the above

D. Options A and C do not compile as they do not use the correct syntax for a for-each loop. The for-each loop is only able to go through an array in ascending order. It is not able to control the order, making Option C incorrect. Therefore, Option D is the answer.

12.What is the output of the following?

```
StringBuilder builder = new StringBuilder();
String str = new String("Leaves growing");
do {
System.out.println(str);
}
while (builder);
System.out.println(builder);
```

A. Leaves growing

B. This is an infinite loop.

C. The code does not compile.

D. The code compiles but throws an exception at runtime.

C. A do-while loop requires a boolean condition. The builder variable is a StringBuilder and not a boolean. The code does not compile, and Option C is correct.

13.What is the result of the following code?

```
6:   int count = 0;
7:   do {
8:   do {
9:   count++;
10:  } while (count < 2);
11:  break;
12: } while (true);
13: System.out.println(count);
```

A. 2
B. 3
C. The code does not compile.
D. This is an infinite loop

A. At first this code appears to be an infinite loop. However, there is a break statement. On line 6, count is set to 0. On line 9, it is changed to 1. Then the condition on line 10 runs. count is less than 2 so the inner loop continues. Then count is set to 2 on the next iteration of the inner loop. The loop condition on line 10 runs again and this time is false. The inner loop is completed. Then line 11 of the outer loop runs and sends execution to after the loop on line 13. At this point count is still 2, so Option A is correct.

14. Which of the following best describes the flow of execution in this for loop if beta always returns false?

```
for (alpha; beta; gamma)
{
delta;
}
```

A. alpha
B. alpha, beta
C. alpha, beta, gamma
D. None of the above

B.The initializer, which is alpha, runs first. Then Java checks the condition, which is beta, to see if loop execution should start. Since beta returns false, the loop is never entered, and Option B is correct.

15.Which of the following best describes the flow of execution in this for loop if the loop body is run exactly once?

```
for (alpha; beta; gamma)
{
delta;
}
```

A. alpha, delta, gamma, beta
B. alpha, beta, delta, gamma, beta
C. alpha, delta, gamma, alpha, beta
D. alpha, beta, delta, gamma, alpha, beta

B. The initializer, which is alpha, runs first. Then Java checks the condition, which is beta, to see if loop execution should start. Then the loop body, which is delta, runs. After the loop execution, the updater, which is gamma, runs. Then the loop condition, which is beta, is checked again. Therefore, Option B is correct.

16. What is the output of the following?

```
public class Shoelaces
{
public static void main(String[] args) {
String tie = null;
while (tie == null);
tie = "shoelace";
System.out.print(tie);
}
}
```

A. null
B. shoelace
C. shoelaceshoelace
D. None of the above

D. The first time the loop condition is checked, the variable tie is null. However, the loop body is empty due to the semicolon right after the condition. This means the loop condition keeps running with no opportunity for tie to be set. Therefore, this is an infinite loop, and Option D is correct.

17.What is the output of the following application?

```
package stocks;
public class Bond {
private static int price = 5;
public boolean sell() {
if(price<10)
{ price++; return true;
}
else if(price>=10) {
return false;
}
}
public static void main(String[] cash) {
new Bond().sell();
new Bond().sell();
new Bond().sell();
System.out.print(price);
}
}
```

A. 5
B. 6
C. 8
D. The code does not compile.