# Java Multiple Choice Questions

**Source Link:**
Java MCQs and Answers with Explanation | Java Quiz (freshersnow.com)

| | |
|---|---|
| **Name** | Java |
| **Exam Type** | MCQ (Multiple Choice Questions) |
| **Category** | **Technical Quiz** |
| **Mode of Quiz** | Offline |

## Top 53 Java Language MCQs with Answers | Practice Java Quiz

**1. Which of the following is not a primitive data type in Java?**

a) Boolean

b) Character

c) String

d) Byte

**Answer: c)** String

**Explanation:** String is a class in Java and not a primitive data type.

**2. What is the default value of an instance variable in Java?**

a) null

b) 0

c) false

d) undefined

**Answer: a)** null

**Explanation:** Instance variables are initialized to null by default.

**3. Which of the following is not a Java keyword?**

a) new

b) public

c) volatile

d) virtual

**Answer: d)** virtual

**Explanation:** virtual is not a Java keyword.

**4. What is the output of the following code?**

```
int x = 10;
int y = 20;
int z = x + y;
System.out.println(z);
```

a) 10

b) 20

c) 30

d) 40

**Answer: c)** 30

**Explanation:** The values of x and y are added and assigned to z, which is then printed.

**5. What is the output of the following code?**

```
int x = 10;
System.out.println(x++);
```

a) 10

b) 11

c) Compiler error

d) Undefined

**Answer: a)** 10

**Explanation:** The value of x is printed before it is incremented.

**6. What is the output of the following code?**

```
int x = 10;
System.out.println(++x);
```

a) 10

b) 11

c) Compiler error

d) Undefined

**Answer: b)** 11

**Explanation:** The value of x is incremented before it is printed.

**7. Which of the following is a valid Java identifier?**

a) 123abc

b) _abc123

c) abc-123

d) 123-abc

**Answer: b)** _abc123

**Explanation:** Java identifiers can start with a letter, underscore or dollar sign.

**8. Which of the following is a subclass of the Exception class?**

a) RuntimeException

b) Error

c) Throwable

d) IllegalArgumentException

**Answer: a)** RuntimeException

**Explanation:** RuntimeException is a subclass of the Exception class.

**9. Which of the following is a checked exception?**

a) NullPointerException

b) RuntimeException

c) ArrayIndexOutOfBoundsException

d) IOException

**Answer: d)** IOException

**Explanation:** IOException is a checked exception, which means that it must be either handled or declared.

**10. Which of the following is used to read input from the user in Java?**

a) System.out.println()

b) System.in.read()

c) Scanner

d) BufferedReader

**Answer: c)** Scanner

**Explanation:** Scanner is used to read input from the user in Java.

**11. Which of the following is not a collection interface in Java?**

a) List

b) Map

c) Queue

d) Array

**Answer: d)** Array

**Explanation:** Array is not a collection interface in Java.

**12. What is the output of the following code?**

String s1 = “Hello”;

String s2 = “World”;

String s3 = s1 + s2;

System.out.println(s3);

a) HelloWorld

b) Hello World

c) HelloWorld

d) Hello_World

**Answer: a)** HelloWorld

**Explanation:** The values of s1 and s2 are concatenated to form s3, which is then printed.

**13. Which of the following is a valid way to declare a multi-dimensional array in Java?**

a) int[] arr = new int[3][3];

b) int[][] arr = new int[3][3];

c) int[][] arr = new int[3,3];

d) int[] arr = new int[]{3,3};

**Answer: b)** int[][] arr = new int[3][3];

**Explanation:** To declare a multi-dimensional array in Java, we use the syntax int[][] arr = new int[3][3], where 3 is the size of both the first and second dimension of the array.

**14. What is the output of the following code?**

int[] arr = new int[5];

System.out.println(arr[0]);

a) 0

b) null

c) Compiler error

d) Undefined

**Answer: a)** 0

**Explanation:** The default value of an int array is 0.

**15. What is the output of the following code?**

int[] arr = new int[]{1, 2, 3, 4, 5};

for (int i = 0; i < arr.length; i++) {

System.out.println(arr[i]);

}

a) 1 2 3 4 5

b) 5 4 3 2 1

c) 1 3 5

d) 2 4

**Answer: a)** 1 2 3 4 5

**Explanation:** The for loop iterates over each element of the array and prints its value.

**16. What is the output of the following code?**

int[] arr = new int[]{1, 2, 3, 4, 5};

int sum = 0;

for (int i = 0; i < arr.length; i++) {

sum += arr[i];

}

System.out.println(sum);

a) 15

b) 10

c) 6

d) 3

**Answer: a)** 15

**Explanation:** The for loop iterates over each element of the array and adds its value to the variable sum, which is then printed.

**17. What is the output of the following code?**

int[] arr = new int[]{1, 2, 3, 4, 5};

System.out.println(arr[5]);

a) 0

b) null

c) Compiler error

d) ArrayIndexOutOfBoundsException

**Answer: d)** ArrayIndexOutOfBoundsException

**Explanation:** The array index 5 is out of bounds, since the array has only 5 elements.

**18. What is the output of the following code?**

int[] arr = new int[]{1, 2, 3, 4, 5};

System.out.println(arr.length);

a) 5

b) 6

c) Compiler error

d) Undefined

**Answer: a)** 5

**Explanation:** The length property of an array returns the number of elements in the array.

**19. What is the output of the following code?**

int x = 10;

if (x < 5) {

System.out.println("x is less than 5");

} else if (x > 20) {

System.out.println("x is greater than 20");

} else {

System.out.println("x is between 5 and 20");

}

a) x is less than 5

b) x is greater than 20

c) x is between 5 and 20

d) Compiler error

**Answer: c)** x is between 5 and 20

**Explanation:** Since x is between 5 and 20, the else block is executed and "x is between 5 and 20" is

**20. Which of the following is the correct syntax for a while loop in Java?**

a) while (x < 10) {

System.out.println("Hello");

}

b) do {

System.out.println("Hello");

} while (x < 10);

c) for (int i = 0; i < 10; i++) {

System.out.println("Hello");

}

d) while (x < 10) System.out.println("Hello");

**Answer: a)** while (x < 10) {

System.out.println("Hello");

}

**Explanation:** The while loop executes the block of code while the specified condition is true.

**21. What is the output of the following code?**

int x = 5;

while (x > 0) {

System.out.println(x);

x–;

}

a) 5 4 3 2 1

b) 1 2 3 4 5

c) 5 4 3 2

d) 1 2 3 4

**Answer: a)** 5 4 3 2 1

**Explanation:** The while loop decrements the value of x by 1 on each iteration, and prints its value until x becomes less than or equal to 0.

**22. What is the output of the following code?**

```
int x = 0;
do {
System.out.println(x);
x++;
} while (x < 5);
```

a) 0 1 2 3 4

b) 1 2 3 4 5

c) 0 1 2 3

d) 1 2 3 4

**Answer: a)** 0 1 2 3 4

**Explanation:** The do-while loop prints the value of x on each iteration until x becomes greater than or equal to 5.

**23. Which of the following is the correct syntax for a for loop in Java?**

a) for (int i = 0; i < 10; i++) {

System.out.println("Hello");

}

b) for (int i = 0; i < 10; i++)

System.out.println("Hello");

c) for (i = 0; i < 10; i++) {

System.out.println("Hello");

}

d) for (int i = 0; i < 10) {

System.out.println("Hello");

}

**Answer: a)** for (int i = 0; i < 10; i++) {

System.out.println("Hello");

}

**Explanation:** The for loop initializes a counter variable, tests a condition, and increments the counter variable on each iteration.

**24. What is the output of the following code?**

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

a) 12345

b) 1234

123

12

1

c) 11111

2222

333

44

5

d) 55555

4444

333

22

1

**Answer: b)** 1234

123

12

1

**Explanation:** The nested for loop prints the values of j up to i on each iteration of the outer loop. This results in the output shown above.

**25. What is the output of the following code?**

int x = 5;

switch (x) {

case 1:

System.out.println(“x is 1”);

break;

case 5:

System.out.println(“x is 5”);

break;

default:

System.out.println(“x is not 1 or 5”);

}

a) x is 1

b) x is 5

c) x is not 1 or 5

d) There is a syntax error in the code.

**Answer: b)** x is 5

**Explanation:** The switch statement tests the value of x against each case, and executes the code for the first matching case. In this case, the code for the case where x equals 5 is executed.

**26. Which of the following is the correct syntax for declaring an array of integers in Java?**

a) int[] myArray = {1, 2, 3, 4, 5};

b) int myArray[] = {1, 2, 3, 4, 5};

c) int[] myArray = new int[5];

d) int myArray[5] = {1, 2, 3, 4, 5};

**Answer: a)** int[] myArray = {1, 2, 3, 4, 5};

**Explanation:** The syntax for declaring an array of integers in Java is to use the int[] keyword followed by the name of the array, an equal sign, and a list of values enclosed in curly braces.

**27. What is the output of the following code?**

int[] myArray = {1, 2, 3, 4, 5};

for (int i = 0; i < myArray.length; i++) {

System.out.print(myArray[i] + ” “);

}

a) 1 2 3 4 5

b) 5 4 3 2 1

c) 1,2,3,4,5

d) 5,4,3,2,1

**Answer: a)** 1 2 3 4 5

**Explanation:** The for loop iterates through each element of the array and prints its value, followed by a space.

**28. What is the output of the following code?**

int[][] myArray = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

System.out.println(myArray[1][2]);

a) 1

b) 2

c) 3

d) 6

**Answer: d)** 6

**Explanation:** The two-dimensional array myArray has three rows and three columns. The expression myArray[1][2] accesses the element in the second row and third column, which has the value 6.

**29. Which of the following is the correct syntax for declaring a method in Java?**

a) void myMethod() {

System.out.println("Hello");

}

b) public void myMethod() {

System.out.println("Hello");

}

c) myMethod() {

System.out.println("Hello");

}

d) public myMethod() {

System.out.println("Hello");

}

**Answer: b)** public void myMethod() {

System.out.println("Hello");

}

**Explanation:** The syntax for declaring a method in Java is to use an access modifier (such as public or private), followed by the return type of the method (such as void or int), the name of the method, and any parameters in parentheses.

**30. What is the output of the following code?**

public static void myMethod(int x) {

System.out.println(x * 2);

}

myMethod(5);

a) 10

b) 5

c) 25

d) There is a syntax error in the code.

**Answer: a)** 10

**Explanation:** The method myMethod takes an integer parameter and prints out its value multiplied by 2. When the method is called with the argument 5, it prints out 10.

**31. Which of the following is the correct syntax for calling a method in Java?**

a) myMethod();

b) myMethod(int x);

c) myMethod(5);

d) myMethod(x);

**Answer: c)** myMethod(5);

**Explanation:** The syntax for calling a method in Java is to use the name of the method followed by any required parameters in parentheses.

**32. What is the output of the following code?**

int x = 5;

if (x > 10) {

System.out.println("x is greater than 10");

} else if (x > 0) {

System.out.println("x is greater than 0");

} else {

System.out.println(“x is less than or equal to 0”);

}

a) x is greater than 10

b) x is greater than 0

c) x is less than or equal to 0

d) There is a syntax error in the code.

**Answer: b)** x is greater than 0

**Explanation:** The if-else statement tests the value of x against each condition in turn. Since x is greater than 0 but not greater than 10, the code for the second condition is executed.

**33. What is the output of the following code?**

for (int i = 1; i <= 10; i++) {

if (i % 2 == 0) {

continue;

}

System.out.println(i);

}

a) 1 3 5 7 9

b) 2 4 6 8 10

c) 1 2 3 4 5 6 7 8 9 10

d) There is a syntax error in the code.

**Answer: a)** 1 3 5 7 9

**Explanation:** The for loop iterates from 1 to 10, skipping any even numbers using the continue statement. The odd numbers are printed to the console.

**34. What is the output of the following code?**

String str1 = “hello”;

String str2 = “world”;

System.out.println(str1 + str2);

a) “helloworld”

b) “hello world”

c) “hello” “world”

d) There is a syntax error in the code.

**Answer: a)** “helloworld”

**Explanation:** The two String variables str1 and str2 are concatenated using the + operator, which results in the combined string “helloworld”.

**35. What is the output of the following code?**

String str = “hello”;

System.out.println(str.charAt(1));

a) ‘h’

b) ‘e’

c) ‘l’

d) ‘o’

**Answer: b)** ‘e’

**Explanation:** The charAt() method returns the character at the specified index in the string. In this case, the second character (index 1) is ‘e’.

**36. What is the output of the following code?**

String str = “hello”;

System.out.println(str.indexOf(‘l’));

a) 0

b) 1

c) 2

d) 3

**Answer: c)** 2

**Explanation:** The indexOf() method returns the index of the first occurrence of the specified character in the string. In this case, the first ‘l’ in the string is at index 2.

**37. What is the output of the following code?**

String str = “hello”;

System.out.println(str.length());

a) 4

b) 5

c) 6

d) 7

**Answer: b)** 5

**Explanation:** The length() method returns the number of characters in the string. In this case, the string “hello” has five characters.

**38. What is the output of the following code?**

String str = “hello”;

System.out.println(str.substring(1, 3));

a) “he”

b) “el”

c) “lo”

d) “ell”

**Answer: d)** “el”

**Explanation:** The substring() method returns a new string that is a substring of the original string. The two arguments specify the starting and ending index of the substring, but the ending index is exclusive. In this case, the substring starts at index 1 (‘e’) and ends at index 3 (‘l’), but the ‘l’ is not included in the substring. Therefore, the result is “el”.

**39. What is the output of the following code?**

String str1 = “hello”;

String str2 = “world”;

System.out.println(str1.equals(str2));

a) true

b) false

c) It will result in a compile error.

d) It will result in a runtime error.

**Answer: b)** false

**Explanation:** The equals() method compares two strings to see if they have the same value. In this case, the strings “hello” and “world” are not the same, so the result is false.

**40. What is the output of the following code?**

String str1 = “hello”;

String str2 = “HELLO”;

System.out.println(str1.equalsIgnoreCase(str2));

a) true

b) false

c) It will result in a compile error.

d) It will result in a runtime error.

**Answer: a)** true

**Explanation:** The equalsIgnoreCase() method compares two strings to see if they have the same value, ignoring case differences. In this case, the strings “hello” and “HELLO” are the same, so the result is true.

**41. What is the output of the following code?**

String str = ” hello “;

System.out.println(str.trim());

a) “hello”

b) ” hello ”

c) ” hello”

d) ” hello ”

**Answer: a)** “hello”

**Explanation:** The trim() method removes leading and trailing white space from a string. In this case, the original string has leading and trailing white space, so the result is the trimmed string “hello”.

**42. What is the output of the following code?**

String str = “hello”;

str.toUpperCase();

System.out.println(str);

a) “hello”

b) “HELLO”

c) It will result in a compile error.

d) It will result in a runtime error.

**Answer: a)** “hello”

**Explanation:** The toUpperCase() method returns a new string that is the original string converted to uppercase. However, it does not modify the original string. In this case, the original string “hello” is printed, because it was not changed by the toUpperCase() method.

**43. What is the output of the following code?**

String str = “HELLO”;

str.toLowerCase();

System.out.println(str);

a) “HELLO”

b) “hello”

c) It will result in a compile error.

d) It will result in a runtime error.

**Answer: a)** “HELLO”

**Explanation:** The toLowerCase() method returns a new string that is the original string converted to lowercase. However, it does not modify the original string. In this case, the original string “HELLO” is printed, because it was not changed by the toLowerCase() method.

**44. What is the output of the following code?**

int[] nums = {1, 2, 3, 4, 5};

for (int i = 0; i < nums.length; i++) {

System.out.print(nums[i] + ” “);

}

a) “1 2 3 4 5”

b) “5 4 3 2 1”

c) “1, 2, 3, 4, 5”

d) “5, 4, 3, 2, 1”

**Answer: a)** “1 2 3 4 5”

**Explanation:** The for loop iterates through the elements of the nums array, using the index variable i. The System.out.print statement prints each element followed by a space, resulting in the output “1 2 3 4 5”.

**45. What is the output of the following code?**

int[] nums = {1, 2, 3, 4, 5};

for (int num : nums) {

System.out.print(num + ” “);

}

a) “1 2 3 4 5”

b) “5 4 3 2 1”

c) “1, 2, 3, 4, 5”

d) “5, 4, 3, 2, 1”

**Answer: a)** “1 2 3 4 5”

**Explanation:** The enhanced for loop iterates through the elements of the nums array, using the variable num to represent each element. The System.out.print statement prints each element followed by a space, resulting in the output “1 2 3 4 5”.

**46. What is the output of the following code?**

```
int[] nums = {1, 2, 3, 4, 5};
for (int i = nums.length – 1; i >= 0; i–) {
System.out.print(nums[i] + ” “);
}
```

a) “1 2 3 4 5”

b) “5 4 3 2 1”

c) “1, 2, 3, 4, 5”

d) “5, 4, 3, 2, 1”

**Answer: b)** “5 4 3 2 1”

**Explanation:** This for loop iterates through the elements of the nums array in reverse order, using the index variable i. The System.out.print statement prints each element followed by a space, resulting in the output “5 4 3 2 1”.

**47. What is the output of the following code?**

```
int[] nums = {1, 2, 3, 4, 5};
for (int i = 0; i < nums.length; i += 2) {
System.out.print(nums[i] + ” “);
}
```

a) “1 2 3 4 5”

b) “2 4”

c) “1 3 5”

d) “5 3 1”

**Answer: c)** “1 3 5”

**Explanation:** This for loop iterates through the elements of the nums array, using the index variable i, but only printing out the elements with even indexes. The System.out.print statement prints each element followed by a space, resulting in the output “1 3 5”.

**48. What is the output of the following code?**

```
int[] nums = {1, 2, 3, 4, 5};
int sum = 0;
for (int i = 0; i < nums.length; i++) {
sum += nums[i];
}
System.out.println(sum);
```

a) 1

b) 15

c) 3

d) 6

**Answer: b)** 15

**Explanation:** This for loop iterates through the elements of the nums array, using the index variable i, and adds each element to the variable sum. The System.out.println statement then prints out the final value of sum, which is 15.

**49. What is the output of the following code?**

int x = 5;

int y = 3;

System.out.println(x / y);

a) 1

b) 1.67

c) 1.5

d) 2

**Answer: a)** 1

**Explanation:** When dividing two integers in Java, the result is always an integer. In this case, the division of 5 by 3 is equal to 1 with a remainder of 2,

**50. What is the output of the following code?**

double x = 5;

double y = 3;

System.out.println(x / y);

a) 1

b) 1.67

c) 1.5

d) 2

**Answer: b)** 1.67

**Explanation:** When dividing two doubles in Java, the result is a double. In this case, the division of 5.0 by 3.0 is equal to 1.6666666666666667, which is printed to the console.

**51. What is the output of the following code?**

int x = 5;

int y = 3;

System.out.println(x % y);

a) 1

b) 1.67

c) 1.5

d) 2

**Answer: a)** 2

**Explanation:** The modulus operator (%) returns the remainder of integer division. In this case, 5 divided by 3 has a remainder of 2, which is printed to the console.

**52. What is the output of the following code?**

String name = “John”;

System.out.println(“Hello, ” + name + “!”);

a) Hello, John!

b) Hello, name!

c) Hello, !

d) John

**Answer: a)** Hello, John!

**Explanation:** The string concatenation operator (+) combines the string “Hello, ” with the value of the name variable (which is “John”) and the string “!”, resulting in the output “Hello, John!”.

**53. What is the output of the following code?**

String name = null;

System.out.println(“Hello, ” + name + “!”);

a) Hello, !

b) Hello, null!

c) Null Pointer Exception

d) Compilation Error

**Answer: b)** Hello, null!

**Explanation:** When a null reference is concatenated with a string, it is treated as the string “null”. So the output of this code is “Hello, null!”.

# Top 48 Java 8 MCQs and Answers | Java 8 Quiz

**1. What is the primary objective of the Java 8 release?**

a) Improved Security Features

b) Improved Performance

c) Functional Programming

d) Improved Collection Libraries

**Answer: c)** Functional Programming.

**Explanation:** Java 8 introduced new features and APIs to support functional programming, such as lambda expressions and streams.

**2. Which of the following is not a functional interface in Java 8?**

a) Consumer

b) Supplier

c) Runnable

d) Comparator

**Answer: d)** Comparator.

**Explanation:** Although it is often used with functional programming constructs such as lambda expressions, Comparator is not a functional interface because it has two abstract methods.

**3. What is a lambda expression in Java 8?**

a) A new type of class

b) A shorthand way of creating an anonymous function

c) A special type of method that can only be called once

d) A way to create a singleton object

**Answer: b)** A shorthand way of creating an anonymous function.

**Explanation:** A lambda expression is a concise way to represent an anonymous function that can be passed around as a value.

**4. What is a stream in Java 8?**

a) A type of input/output stream for reading and writing files

b) A sequence of objects that can be processed in parallel

c) A data structure that holds key-value pairs

d) A type of network socket

**Answer: b)** A sequence of objects that can be processed in parallel.

**Explanation:** Streams are a new type of collection in Java 8 that allows for efficient processing of large datasets in parallel.

**5. What is the difference between an intermediate operation and a terminal operation in a stream pipeline?**

a) Intermediate operations modify the stream, while terminal operations do not.

b) Intermediate operations are evaluated lazily, while terminal operations are evaluated eagerly.

c) Intermediate operations are executed sequentially, while terminal operations are executed in parallel.

d) Intermediate operations return a new stream, while terminal operations return a non-stream result.

**Answer: a)** Intermediate operations modify the stream, while terminal operations do not.

**Explanation:** Intermediate operations transform the stream into a new stream, while terminal operations produce a result or a side-effect.

**6. Which of the following is not a built-in functional interface in Java 8?**

a) Predicate

b) Function

c) BiFunction

d) Procedure

**Answer: d)** Procedure.

**Explanation:** Although it is often used in functional programming, Procedure is not a built-in functional interface in Java 8.

**7. Which of the following is not a valid lambda expression in Java 8?**

a) (int x, int y) -> x + y

b) () -> "Hello, World!"

c) (String s) -> System.out.println(s)

d) (int x) -> {return x*x;}

**Answer: c)** (String s) -> System.out.println(s).

**Explanation:** This lambda expression does not return a value, which violates the functional interface contract.

**8. What is the purpose of the Optional class in Java 8?**

a) To represent a value that may be null

b) To represent an empty collection

c) To represent a primitive value

d) To represent a stream of values

**Answer: a)** To represent a value that may be null.

**Explanation:** The Optional class provides a way to handle null values without resorting to null checks and exceptions.

**9. Which of the following is not a valid collector in Java 8?**

a) toList()

b) toSet()

c) toMap()

d) toQueue()

**Answer: d)** toQueue().

**Explanation:** Although queues are a common data structure, there is no built-in collector to create a queue from a stream in Java 8.

**10. What is the purpose of the peek() method in a stream pipeline?**

a) To modify the stream elements

b) To inspect the stream elements

c) To terminate the stream

d) To filter the stream elements

**Answer: b)** To inspect the stream elements.

**Explanation:** The peek() method allows you to perform a side-effect on each element of the stream without modifying the stream itself.

**11. What is the syntax for a method reference in Java 8?**

a) (args) -> ClassName::methodName

b) ClassName::methodName(args)

c) (args) -> methodName::ClassName

d) methodName::ClassName(args)

**Answer: b)** ClassName::methodName(args).

**Explanation:** Method references allow you to refer to an existing method as a lambda expression, without having to write out the full lambda syntax.

**12. What is the purpose of the reduce() method in a stream pipeline?**

a) To transform a stream into a new stream

b) To terminate the stream and return a single result

c) To filter the stream elements

d) To perform a side-effect on the stream elements

**Answer: b)** To terminate the stream and return a single result.

**Explanation:** The reduce() method combines the elements of a stream into a single result using a specified binary operator.

**13. What is the difference between a parallel stream and a sequential stream?**

a) A parallel stream is more efficient than a sequential stream.

b) A parallel stream processes elements in parallel, while a sequential stream processes elements sequentially.

c) A parallel stream is always unordered, while a sequential stream is always ordered.

d) A parallel stream is always lazy, while a sequential stream is always eager.

**Answer: b)** A parallel stream processes elements in parallel, while a sequential stream processes elements sequentially.

**Explanation:** Parallel streams can offer improved performance for certain types of processing, but they may also introduce additional overhead and synchronization issues.

**14. What is the purpose of the flatMap() method in a stream pipeline?**

a) To combine multiple streams into a single stream

b) To transform each element of a stream into a new stream

c) To remove duplicates from a stream

d) To sort the elements of a stream

**Answer: b)** To transform each element of a stream into a new stream.

**Explanation:** The flatMap() method allows you to replace each element of a stream with a new stream, which is then flattened into a single stream.

**15. What is the purpose of the forEach() method in a stream pipeline?**

a) To transform the stream into a new stream

b) To terminate the stream and return a single result

c) To perform a side-effect on each element of the stream

d) To filter the stream elements

**Answer: c)** To perform a side-effect on each element of the stream.

**Explanation:** The forEach() method allows you to perform an action on each element of the stream, without modifying the stream itself.

**16. What is the difference between a consumer and a supplier in Java 8?**

a) A consumer takes one argument and returns a value, while a supplier takes no arguments and returns a value.

b) A consumer takes no arguments and returns a value, while a supplier takes one argument and returns a value.

c) A consumer takes one argument and returns no value, while a supplier takes no arguments and returns a value.

d) A consumer takes no arguments and returns no value, while a supplier takes one argument and returns a value.

**Answer: c)** A consumer takes one argument and returns no value, while a supplier takes no arguments and returns a value.

**Explanation:** A consumer represents an operation that takes an input and performs a side-effect, while a supplier represents an operation that produces a value.

**17. What is the purpose of the map() method in a stream pipeline?**

a) To remove elements from the stream

b) To transform each element of the stream into a new element

c) To combine elements of the stream into a single element

d) To terminate the stream and return a single result

**Answer: b)** To transform each element of the stream into a new element.

**Explanation:** The map() method allows you to apply a function to each element of the stream, transforming the stream into a new stream of elements.

**18. Which of the following is true about the Stream API in Java 8?**

a) It is used to manipulate collections of data

b) It can only be used with parallel streams

c) It can only be used with sequential streams

d) It is a replacement for the java.util.Collection interface

**Answer: a)** It is used to manipulate collections of data.

**Explanation:** The Stream API provides a way to process collections of data in a declarative and functional style, without modifying the original collection.

**19. What is the difference between a predicate and a function in Java 8?**

a) A predicate takes an input and returns a Boolean, while a function takes an input and returns a value.

b) A predicate takes no input and returns a value, while a function takes an input and returns a value.

c) A predicate takes an input and returns no value, while a function takes no input and returns a value.

d) A predicate takes no input and returns no value, while a function takes an input and returns a value.

**Answer: a)** A predicate takes an input and returns a Boolean, while a function takes an input and returns a value.

**Explanation:** A predicate is a type of function that returns a Boolean value, indicating whether a condition is true or false.

**20. What is the purpose of the toArray() method in a stream pipeline?**

a) To transform the stream into an array

b) To filter the stream elements

c) To terminate the stream and return a single result

d) To perform a side-effect on each element of the stream

**Answer: a)** To transform the stream into an array.

**Explanation:** The toArray() method allows you to convert a stream into an array of elements.

**21. What is the difference between findAny() and findFirst() in Java 8?**

a) findAny() returns any element of the stream, while findFirst() returns the first element of the stream.

b) findAny() returns the first element of the stream, while findFirst() returns any element of the stream.

c) findAny() and findFirst() are identical methods.

d) findAny() and findFirst() can only be used with parallel streams.

**Answer: a)** findAny() returns any element of the stream, while findFirst() returns the first element of the stream.

**Explanation:** If the stream has an encounter order, findFirst() will return the first element, while findAny() may return any element. If the stream has no encounter order, both methods will behave the same way.

**22. What is the difference between a stateful and stateless operation in a stream pipeline?**

a) A stateful operation depends on the order of the stream elements, while a stateless operation does not.

b) A stateful operation modifies the state of the stream pipeline, while a stateless operation does not.

c) A stateful operation requires additional memory, while a stateless operation does not.

d) A stateful operation is always parallel, while a stateless operation is always sequential.

**Answer: b)** A stateful operation modifies the state of the stream pipeline, while a stateless operation does not.

**Explanation:** A stateful operation is one that depends on the state of the stream pipeline, such as sorting or distinct, while a stateless operation is one that does not depend on the state, such as map or filter.

**23. Which of the following is a valid method reference in Java 8?**

a) (args) -> System.out::println

b) (args) -> Math.abs

c) (args) -> Integer::valueOf

d) (args) -> String::split

**Answer: c)** (args) -> Integer::valueOf.

**Explanation:** A method reference is a shorthand syntax for a lambda expression that calls a method. In this case, the method reference Integer::valueOf refers to the static method valueOf() of the Integer class, which takes a String argument and returns an Integer object.

**24. What is the difference between peek() and forEach() in a stream pipeline?**

a) peek() performs a side-effect on each element of the stream, while forEach() terminates the stream and performs a side-effect on each element.

b) peek() terminates the stream and performs a side-effect on each element, while forEach() performs a side-effect on each element of the stream.

c) peek() and forEach() are identical methods.

d) peek() and forEach() can only be used with parallel streams.

**Answer: a)** peek() performs a side-effect on each element of the stream, while forEach() terminates the stream and performs a side-effect on each element.

**Explanation:** The peek() method allows you to perform a side-effect on each element of the stream, without modifying the stream. The forEach() method performs a side-effect on each element of the stream, and terminates the stream.

**25. Which of the following is true about the reduce() method in Java 8?**

a) It is used to perform a side-effect on each element of the stream.

b) It is used to transform each element of the stream into a new element.

c) It is used to terminate the stream and return a single result.

d) It is used to filter the stream elements.

**Answer: c)** It is used to terminate the stream and return a single result.

**Explanation:** The reduce() method allows you to reduce a stream of elements into a single result, using a binary operator. For example, you can use reduce() to find the sum or maximum value of a stream of integers.

**26. Which of the following is true about the collect() method in Java 8?**

a) It is used to terminate the stream and return a single result.

b) It is used to perform a side-effect on each element of the stream.

c) It is used to transform each element of the stream into a new element.

d) It is used to accumulate the elements of the stream into a collection.

**Answer: d)** It is used to accumulate the elements of the stream into a collection.

**Explanation:** The collect() method allows you to accumulate the elements of a stream into a collection, using a Collector. For example, you can use collect() to create a list, set, or map from a stream of elements.

**27. Which of the following is true about the Optional class in Java 8?**

a) It is used to represent a value that may be absent.

b) It is used to represent a collection of values.

c) It is used to represent a stream of values.

d) It is used to represent a function that takes an optional parameter.

**Answer: a)** It is used to represent a value that may be absent.

**Explanation:** The Optional class is a container object that may or may not contain a non-null value. It is used to avoid NullPointerExceptions and make code more readable.

**28. Which of the following is not a functional interface in Java 8?**

a) Runnable

b) Comparable

c) Callable

d) Predicate

**Answer: b)** Comparable.

**Explanation:** Although Comparable is an interface in Java, it is not considered a functional interface because it has more than one abstract method. A functional interface is an interface that has exactly one abstract method.

**29. Which of the following interfaces is not a subtype of the Function interface in Java 8?**

a) UnaryOperator

b) BiFunction

c) Supplier

d) Predicate

**Answer: c)** Supplier.

**Explanation:** The Supplier interface is not a subtype of the Function interface, but it is a functional interface that represents a supplier of results.

**30. What is the purpose of the Stream.Builder interface in Java 8?**

a) To provide a builder for creating Stream objects.

b) To provide a builder for creating Collection objects.

c) To provide a builder for creating Comparator objects.

d) To provide a builder for creating Function objects.

**Answer: a)** To provide a builder for creating Stream objects.

**Explanation:** The Stream.Builder interface is used to create and modify a stream of elements, by adding or removing elements to the stream.

**31. Which of the following is true about parallel streams in Java 8?**

a) They can improve the performance of some operations on large data sets.

b) They can only be used with sequential streams.

c) They always perform better than sequential streams, regardless of the data set size.

d) They can only be used with certain types of data, such as arrays.

**Answer: a)** They can improve the performance of some operations on large data sets.

**Explanation:** Parallel streams can improve the performance of some operations on large data sets by dividing the work into multiple threads that can run concurrently on different cores of the CPU.

**32. Which of the following is true about the Stream.concat() method in Java 8?**

a) It creates a new stream that is the concatenation of two streams.

b) It creates a new stream that is the intersection of two streams.

c) It creates a new stream that is the union of two streams.

d) It creates a new stream that is the difference of two streams.

**Answer: a)** It creates a new stream that is the concatenation of two streams.

**Explanation:** The Stream.concat() method creates a new stream that contains all the elements of two given streams, in the order they are encountered.

**33. Which of the following is true about the Files.walk() method in Java 8?**

a) It returns a stream of all the files in a directory tree.

b) It returns a stream of all the directories in a directory tree.

c) It returns a stream of all the files and directories in a directory tree.

d) It returns a stream of all the files and directories in a directory.

**Answer: c)** It returns a stream of all the files and directories in a directory tree.

**Explanation:** The Files.walk() method returns a stream of all the files and directories in a directory tree, starting from a given path.

**34. Which of the following is true about the StringJoiner class in Java 8?**

a) It is used to join two strings into a single string.

b) It is used to concatenate two strings into a single string.

c) It is used to create a comma-separated list of strings.

d) It is used to format strings.

**Answer: c)** It is used to create a comma-separated list of strings.

**Explanation:** The StringJoiner class is used to create a sequence of strings separated by a delimiter, such as a comma, and optionally with a prefix and suffix.

**35. Which of the following is true about method references in Java 8?**

a) They are a way to create new instances of classes.

b) They are a way to invoke constructors of classes.

c) They are a way to refer to methods or constructors of classes.

d) They are a way to define anonymous inner classes.

**Answer: c)** They are a way to refer to methods or constructors of classes.

**Explanation:** Method references are a shorthand way of referring to an existing method or constructor of a class, without the need to create a lambda expression.

**36. Which of the following interfaces represents a sequence of primitive double values in Java 8?**

a) DoubleConsumer

b) DoubleFunction

c) DoubleSupplier

d) DoubleStream

**Answer: d)** DoubleStream.

**Explanation:** The DoubleStream interface represents a sequence of primitive double values in Java 8, and provides various methods for performing operations on these values.

**37. Which of the following interfaces represents a supplier of objects in Java 8?**

a) Supplier

b) Consumer

c) Function

d) Predicate

**Answer: a)** Supplier.

**Explanation:** The Supplier interface represents a supplier of objects, and has a single abstract method get() that returns an object of the specified type.

**38. Which of the following is true about the Optional class in Java 8?**

a) It is used to represent a value that may or may not be present.

b) It is used to represent an empty collection.

c) It is used to represent a nullable object.

d) It is used to represent a default value.

**Answer: a)** It is used to represent a value that may or may not be present.

**Explanation:** The Optional class is used to represent a value that may or may not be present, and provides methods for working with this value.

**39. Which of the following interfaces represents a function that accepts a long-valued argument and produces a result of the specified type in Java 8?**

a) LongPredicate

b) LongSupplier

c) LongFunction

d) LongConsumer

**Answer: c)** LongFunction.

**Explanation:** The LongFunction interface represents a function that accepts a long-valued argument and produces a result of the specified type.

**40. Which of the following is true about the Stream.collect() method in Java 8?**

a) It returns a new stream that is the result of applying a reduction operation to the elements of the stream.

b) It performs a mutable reduction operation on the elements of the stream.

c) It performs an immutable reduction operation on the elements of the stream.

d) It performs a filter operation on the elements of the stream.

**Answer: b)** It performs a mutable reduction operation on the elements of the stream.

**Explanation:** The Stream.collect() method performs a mutable reduction operation on the elements of the stream, using a Collector to accumulate the elements into a result container.

**41. Which of the following interfaces represents a function that accepts a double-valued argument and produces a result of the specified type in Java 8?**

a) DoublePredicate

b) DoubleSupplier

c) DoubleFunction

d) DoubleConsumer

**Answer: c)** DoubleFunction.

**Explanation:** The DoubleFunction interface represents a function that accepts a double-valued argument and produces a result of the specified type.

**42. Which of the following is true about the Stream.sorted() method in Java 8?**

a) It sorts the elements of the stream in natural order.

b) It sorts the elements of the stream using the specified Comparator.

c) It returns a new stream that is the result of applying a reduction operation to the elements of the stream.

d) It performs a filter operation on the elements of the stream.

**Answer: b)** It sorts the elements of the stream using the specified Comparator.

**Explanation:** The Stream.sorted() method sorts the elements of the stream using the specified Comparator, or in natural order if no Comparator is provided.

**43. Which of the following interfaces represents a function that accepts a boolean-valued argument and produces a result of the specified type in Java 8?**

a) Predicate

b) Supplier

c) Function

d) Consumer

**Answer: c)** Function.

**Explanation:** The Function interface represents a function that accepts an argument of one type and produces a result of another type, and has a single abstract method apply() that takes an argument of the specified type and returns a result of the specified type.

**44. Which of the following interfaces represents a consumer of long-valued arguments in Java 8?**

a) LongSupplier

b) LongPredicate

c) LongFunction

d) LongConsumer

**Answer: d)** LongConsumer.

**Explanation:** The LongConsumer interface represents a consumer of long-valued arguments, and has a single abstract method accept() that takes a long value as input and returns no result.

**45. Which of the following is true about the Stream.flatMap() method in Java 8?**

a) It returns a new stream that is the result of applying a reduction operation to the elements of the stream.

b) It performs a mutable reduction operation on the elements of the stream.

c) It performs an immutable reduction operation on the elements of the stream.

d) It returns a new stream that is the result of applying a function to each element of the stream and flattening the result.

**Answer: d)** It returns a new stream that is the result of applying a function to each element of the stream and flattening the result.

**Explanation:** The Stream.flatMap() method returns a new stream that is the result of applying a function to each element of the stream and flattening the result into a single stream.

**46. Which of the following interfaces represents a consumer of double-valued arguments in Java 8?**

a) DoubleSupplier

b) DoublePredicate

c) DoubleFunction

d) DoubleConsumer

**Answer: d)** DoubleConsumer.

**Explanation:** The DoubleConsumer interface represents a consumer of double-valued arguments, and has a single abstract method accept() that takes a double value as input and returns no result.

**47. Which of the following is true about the Optional.orElse() method in Java 8?**

a) It returns the value of the Optional if present, or throws an exception if not.

b) It returns the value of the Optional if present, or the specified default value if not.

c) It returns true if the Optional contains a value, and false otherwise.

d) It returns the value of the Optional if present, or an empty Optional if not.

**Answer: b)** It returns the value of the Optional if present, or the specified default value if not.

**Explanation:** The Optional.orElse() method returns the value of the Optional if present, or the specified default value if not.

**48. Which of the following interfaces represents a function that accepts an int-valued argument and produces a result of the specified type in Java 8?**

a) IntPredicate

b) IntFunction

c) IntConsumer

d) IntSupplier

**Answer: b)** IntFunction.

**Explanation:** The IntFunction interface represents a function that accepts an int-valued argument and produces a result of the specified type, and has a single abstract method apply() that takes an int value as input and returns a result of the specified type.

# Top 70 Java 9 MCQs and Answers | Java 9 Quiz

**1. What is the module system introduced in Java 9?**

a) A new package management system

b) A new type of class loader

c) A new concurrency library

d) A new garbage collector

**Answer: a)** A new package management system

**Explanation:** Java 9 introduced a module system that allows you to create a new kind of Java artifact called a module. A module is a group of related packages, and it provides better encapsulation and control over dependencies.

**2. Which keyword is used to declare a module in Java 9?**

a) package

b) module

c) import

d) export

**Answer: b)** module

**Explanation:** The keyword “module” is used to declare a module in Java 9.

**3. What is the purpose of the “exports” keyword in a module-info.java file?**

a) To specify which packages should be included in the module

b) To specify which packages should be accessible to other modules

c) To specify which classes should be included in the module

d) To specify which classes should be accessible to other modules

**Answer: b)** To specify which packages should be accessible to other modules

**Explanation:** The “exports” keyword is used in the module-info.java file to specify which packages should be accessible to other modules.

**4. What is the purpose of the “requires” keyword in a module-info.java file?**

a) To specify which packages should be included in the module

b) To specify which packages should be accessible to other modules

c) To specify which modules are required by this module

d) To specify which modules require this module

**Answer: c)** To specify which modules are required by this module

**Explanation:** The “requires” keyword is used in the module-info.java file to specify which modules are required by this module.

**5. Which of the following is a benefit of using modules in Java 9?**

a) Improved performance

b) Better encapsulation

c) Easier development

d) All of the above

**Answer: b)** Better encapsulation

**Explanation:** One of the key benefits of using modules in Java 9 is better encapsulation, which allows you to control which packages are accessible to other modules.

**6. Which interface is used to represent an immutable set of characters in Java?**

a) CharSequence

b) String

c) StringBuilder

d) StringBuffer

**Answer: b)** String

**Explanation:** The String class in Java represents an immutable set of characters.

**7. Which method is used to concatenate two strings in Java?**

a) append()

b) concat()

c) join()

d) merge()

**Answer: b)** concat()

**Explanation:** The concat() method is used to concatenate two strings in Java.

**8. Which of the following is NOT a type of exception in Java?**

a) Checked exceptions

b) Unchecked exceptions

c) Runtime exceptions

d) Compilation exceptions

**Answer: d)** Compilation exceptions

**Explanation:** Compilation exceptions are not a type of exception in Java. They are errors that occur during the compilation process.

**9. What is the purpose of the "finally" block in a try-catch-finally statement in Java?**

a) To catch and handle exceptions

b) To specify the code that should be executed after the try or catch block

c) To specify the code that should be executed before the try or catch block

d) None of the above

**Answer: b)** To specify the code that should be executed after the try or catch block

**Explanation:** The "finally" block is used to specify the code that should be executed after the try or catch block, regardless of whether an exception was thrown or not.

**10. Which keyword is used to throw an exception in Java?**

a) throw

b) catch

c) try

d) finally

**Answer: a)** throw

**Explanation:** The "throw" keyword is used to explicitly throw an exception in Java.

**11. Which method is used to catch any type of exception in Java?**

a) catchAll()

b) catch(Exception e)

c) catch(Throwable t)

d) catch(Exception e, Throwable t)

**Answer: c)** catch(Throwable t)

**Explanation:** The catch(Throwable t) method is used to catch any type of exception in Java.

**12. Which keyword is used to indicate that a method can throw an exception?**

a) catch

b) throw

c) throws

d) try

**Answer: c)** throws

**Explanation:** The "throws" keyword is used to indicate that a method can throw an exception.

**13. Which of the following is NOT a type of loop in Java?**

a) for loop

b) while loop

c) do-while loop

d) until loop

**Answer: d)** until loop

**Explanation:** There is no "until" loop in Java.

**14. Which of the following is used to exit a loop in Java?**

a) break

b) continue

c) return

d) exit

**Answer: a)** break

**Explanation:** The “break” keyword is used to exit a loop in Java.

**15. Which of the following is used to skip an iteration in a loop in Java?**

a) break

b) continue

c) return

d) skip

**Answer: b)** continue

**Explanation:** The “continue” keyword is used to skip an iteration in a loop in Java.

**16. Which of the following is a data structure in Java that stores elements in a fixed-size array?**

a) ArrayList

b) LinkedList

c) ArrayDeque

d) PriorityQueue

**Answer: c)** ArrayDeque

**Explanation:** The ArrayDeque class in Java is a data structure that stores elements in a fixed-size array.

**17. Which of the following is a data structure in Java that stores elements in a resizable array?**

a) ArrayList

b) LinkedList

c) ArrayDeque

d) PriorityQueue

**Answer: a)** ArrayList

**Explanation:** The ArrayList class in Java is a data structure that stores elements in a resizable array.

**18. Which of the following is a data structure in Java that stores elements in a linked list?**

a) ArrayList

b) LinkedList

c) ArrayDeque

d) PriorityQueue

**Answer: b)** LinkedList

**Explanation:** The LinkedList class in Java is a data structure that stores elements in a linked list.

**19. Which method is used to add an element to the end of an ArrayList in Java?**

a) add()

b) addFirst()

c) addLast()

d) push()

**Answer: a)** add()

**Explanation:** The add() method is used to add an element to the end of an ArrayList in Java.

**20. Which method is used to remove an element from an ArrayList in Java?**

a) remove()

b) removeFirst()

c) removeLast()

d) pop()

**Answer: a)** remove()

**Explanation:** The remove() method is used to remove an element from an ArrayList in Java.

**21. Which of the following is a keyword used to define a lambda expression in Java?**

a) function

b) lambda

c) -> (arrow)

d) function() (parentheses)

**Answer: c)** -> (arrow)

**Explanation:** The "->" (arrow) operator is used to define a lambda expression in Java.

**22. Which of the following is NOT a type of method reference in Java?**

a) Reference to a static method

b) Reference to an instance method of an object of a particular class

c) Reference to an instance method of an arbitrary object of a particular class

d) Reference to an instance variable of an object

**Answer: d)** Reference to an instance variable of an object

**Explanation:** There is no type of method reference in Java that refers to an instance variable of an object.

**23. Which interface is used to define functional interfaces in Java?**

a) Function

b) Predicate

c) Supplier

d) Runnable

**Answer: a)** Function

**Explanation:** The Function interface is used to define functional interfaces in Java.

**24. Which of the following is NOT a method of the Function interface in Java?**

a) apply()

b) andThen()

c) compose()

d) then()

**Answer: d)** then()

**Explanation:** There is no method called "then()" in the Function interface in Java.

**25. Which interface is used to represent a supplier of results in Java?**

a) Function

b) Predicate

c) Supplier

d) Runnable

**Answer: c)** Supplier

**Explanation:** The Supplier interface is used to represent a supplier of results in Java.

**26. Which interface is used to represent a predicate (a function that returns a boolean value) in Java?**

a) Function

b) Predicate

c) Supplier

d) Runnable

**Answer: b)** Predicate

**Explanation:** The Predicate interface is used to represent a predicate (a function that returns a boolean value) in Java.

**27. Which interface is used to represent a function that takes two arguments and returns a result in Java?**

a) BiFunction

b) BinaryOperator

c) UnaryOperator

d) Consumer

**Answer: a)** BiFunction

**Explanation:** The BiFunction interface is used to represent a function that takes two arguments and returns a result in Java.

**28. Which interface is used to represent a binary operator (a function that takes two arguments and returns a result of the same type) in Java?**

a) BiFunction

b) BinaryOperator

c) UnaryOperator

d) Consumer

**Answer: b)** BinaryOperator

**Explanation:** The BinaryOperator interface is used to represent a binary operator (a function that takes two arguments and returns a result of the same type) in Java.

**29. Which interface is used to represent a unary operator (a function that takes one argument and returns a result of the same type) in Java?**

a) BiFunction

b) BinaryOperator

c) UnaryOperator

d) Consumer

**Answer: c)** UnaryOperator

**Explanation:** The UnaryOperator interface is used to represent a unary operator (a function that takes one argument and returns a result of the same type) in Java.

**30. Which interface is used to represent a function that does not return a result in Java?**

a) Function

b) Predicate

c) Consumer

d) Supplier

**Answer: c)** Consumer

**Explanation:** The Consumer interface is used to represent a function that does not return a result in Java.

**31. Which method is used to create a stream from an array in Java?**

a) stream()

b) arrayStream()

c) createStream()

d) fromArray()

**Answer: a)** stream()

**Explanation:** The stream() method is used to create a stream from an array in Java.

**32. Which method is used to create a stream from a collection in Java?**

a) stream()

b) collectionStream()

c) createStream()

d) fromCollection()

**Answer: a)** stream()

**Explanation:** The stream() method is used to create a stream from a collection in Java.

**33. Which method is used to filter elements in a stream based on a given predicate in Java?**

a) map()

b) flatMap()

c) filter()

d) reduce()

**Answer: c)** filter()

**Explanation:** The filter() method is used to filter elements in a stream based on a given predicate in Java.

**34. Which method is used to transform elements in a stream in Java?**

a) map()

b) flatMap()

c) filter()

d) reduce()

**Answer: a)** map()

**Explanation:** The map() method is used to transform elements in a stream in Java.

**35. Which method is used to concatenate the elements of a stream into a single string in Java?**

a) reduce()

b) collect()

c) join()

d) concatenate()

**Answer: c)** join()

**Explanation:** The join() method is used to concatenate the elements of a stream into a single string in Java.

**36. Which method is used to perform a reduction operation on the elements of a stream in Java?**

a) reduce()

b) collect()

c) join()

d) map()

**Answer: a)** reduce()

**Explanation:** The reduce() method is used to perform a reduction operation on the elements of a stream in Java.

**37. Which method is used to convert a stream of objects into an array in Java?**

a) toArray()

b) toList()

c) toSet()

d) toMap()

**Answer: a)** toArray()

**Explanation:** The toArray() method is used to convert a stream of objects into an array in Java.

**38. Which method is used to convert a stream of objects into a list in Java?**

a) toArray()

b) toList()

c) toSet()

d) toMap()

**Answer: b)** toList()

**Explanation:** The toList() method is used to convert a stream of objects into a list in Java.

**39. Which method is used to convert a stream of objects into a set in Java?**

a) toArray()

b) toList()

c) toSet()

d) toMap()

**Answer: c)** toSet()

**Explanation:** The toSet() method is used to convert a stream of objects into a set in Java.

**40. Which method is used to convert a stream of objects into a map in Java?**

a) toArray()

b) toList()

c) toSet()

d) toMap()

**Answer: d)** toMap()

**Explanation:** The toMap() method is used to convert a stream of objects into a map in Java.

**41. Which class is used to represent date and time values in Java?**

a) Date

b) Time

c) DateTime

d) LocalDateTime

**Answer: d)** LocalDateTime

**Explanation:** The LocalDateTime class is used to represent date and time values in Java.

**42. Which class is used to format date and time values in Java?**

a) DateFormat

b) SimpleDateFormat

c) DateTimeFormatter

d) DateFormatter

**Answer: c)** DateTimeFormatter

**Explanation:** The DateTimeFormatter class is used to format date and time values in Java.

**43. Which class is used to perform file input and output operations in Java?**

a) File

b) FileInputStream

c) FileOutputStream

d) FileWriter

**Answer: a)** File

**Explanation:** The File class is used to perform file input and output operations in Java.

**44. Which class is used to read data from a file in Java?**

a) File

b) FileInputStream

c) BufferedReader

d) FileReader

**Answer: b)** FileInputStream

**Explanation:** The FileInputStream class is used to read data from a file in Java.

**45. Which class is used to write data to a file in Java?**

a) File

b) FileOutputStream

c) BufferedWriter

d) FileWriter

**Answer: b)** FileOutputStream

**Explanation:** The FileOutputStream class is used to write data to a file in Java.

**46. Which class is used to read text from a file in Java?**

a) File

b) FileInputStream

c) BufferedReader

d) FileReader

**Answer: d)** FileReader

**Explanation:** The FileReader class is used to read text from a file in Java.

**47. Which class is used to write text to a file in Java?**

a) File

b) FileWriter

c) FileOutputStream

d) PrintWriter

**Answer: b)** FileWriter

**Explanation:** The FileWriter class is used to write text to a file in Java.

**48. Which class is used to represent a socket for communication between two computers over a network in Java?**

a) ServerSocket

b) Socket

c) DatagramSocket

d) MulticastSocket

**Answer: b)** Socket

**Explanation:** The Socket class is used to represent a socket for communication between two computers over a network in Java.

**49. Which class is used to represent a server socket for communication between two computers over a network in Java?**

a) ServerSocket

b) Socket

c) DatagramSocket

d) MulticastSocket

**Answer: a)** ServerSocket

**Explanation:** The ServerSocket class is used to represent a server socket for communication between two computers over a network in Java.

**50. Which class is used to represent a datagram socket for communication between two computers over a network in Java?**

a) ServerSocket

b) Socket

c) DatagramSocket

d) MulticastSocket

**Answer: c)** DatagramSocket

**Explanation:** The DatagramSocket class is used to represent a datagram socket for communication between two computers over a network in Java.

**51. Which class is used to represent a multicast socket for communication between multiple computers over a network in Java?**

a) ServerSocket

b) Socket

c) DatagramSocket

d) MulticastSocket

**Answer: d)** MulticastSocket

**Explanation:** The MulticastSocket class is used to represent a multicast socket for communication between multiple computers over a network in Java.

**52. Which class is used to create a thread in Java?**

a) Thread

b) Runnable

c) Executor

d) Callable

**Answer: a)** Thread

**Explanation:** The Thread class is used to create a thread in Java.

**53. Which interface is used to create a thread in Java?**

a) Thread

b) Runnable

c) Executor

d) Callable

**Answer: b)** Runnable

**Explanation:** The Runnable interface is used to create a thread in Java.

**54. Which class is used to execute a task asynchronously in Java?**

a) Thread

b) Runnable

c) Executor

d) Callable

**Answer: c)** Executor

**Explanation:** The Executor class is used to execute a task asynchronously in Java.

**55. Which interface is used to execute a task asynchronously and return a result in Java?**

a) Thread

b) Runnable

c) Executor

d) Callable

**Answer: d)** Callable

**Explanation:** The Callable interface is used to execute a task asynchronously and return a result in Java.

**56. Which class is used to schedule a task for execution after a specified delay in Java?**

a) Timer

b) TimerTask

c) ScheduledExecutorService

d) ScheduledThreadPoolExecutor

**Answer: a)** Timer

**Explanation:** The Timer class is used to schedule a task for execution after a specified delay in Java.

**57. Which class is used to represent a task that can be scheduled for execution by a Timer in Java?**

a) Timer

b) TimerTask

c) ScheduledExecutorService

d) ScheduledThreadPoolExecutor

**Answer: b)** TimerTask

**Explanation:** The TimerTask class is used to represent a task that can be scheduled for execution by a Timer in Java.

**58. Which interface is used to schedule a task for execution at a fixed rate in Java?**

a) TimerTask

b) ScheduledExecutorService

c) ScheduledThreadPoolExecutor

d) ScheduledFuture

**Answer: b)** ScheduledExecutorService

**Explanation:** The ScheduledExecutorService interface is used to schedule a task for execution at a fixed rate in Java.

**59. Which class is used to represent a thread pool that can schedule tasks for execution at a fixed rate in**

**Java?**

a) TimerTask

b) ScheduledExecutorService

c) ScheduledThreadPoolExecutor

d) ScheduledFuture

**Answer: c)** ScheduledThreadPoolExecutor

**Explanation:** The ScheduledThreadPoolExecutor class is used to represent a thread pool that can schedule tasks for execution at a fixed rate in Java.

**60. Which interface is used to represent the result of an asynchronous computation in Java?**

a) Future

b) CompletableFuture

c) Callable

d) Executor

**Answer: a)** Future

**Explanation:** The Future interface is used to represent the result of an asynchronous computation in Java.

**61. Which class is used to represent a CompletableFuture in Java?**

a) Future

b) CompletableFuture

c) Callable

d) Executor

**Answer: b)** CompletableFuture

**Explanation:** The CompletableFuture class is used to represent a CompletableFuture in Java.

**62. Which method is used to read a line of text from the standard input in Java?**

a) System.in.read()

b) System.in.read(buffer)

c) System.console().readLine()

d) Scanner.nextLine()

**Answer: d)** Scanner.nextLine()

**Explanation:** The Scanner class is used to read input from the standard input in Java, and the nextLine() method is used to read a line of text.

**63. Which class is used to represent an HTTP connection in Java?**

a) HttpURLConnection

b) URLConnection

c) HttpClient

d) HttpsURLConnection

**Answer: a)** HttpURLConnection

**Explanation:** The HttpURLConnection class is used to represent an HTTP connection in Java.

**64. Which class is used to represent a connection to a URL in Java?**

a) HttpURLConnection

b) URLConnection

c) HttpClient

d) HttpsURLConnection

**Answer: b)** URLConnection

**Explanation:** The URLConnection class is used to represent a connection to a URL in Java.

**65. Which class is used to send HTTP requests and receive HTTP responses in Java?**

a) HttpURLConnection

b) URLConnection

c) HttpClient

d) HttpsURLConnection

**Answer: c)** HttpClient

**Explanation:** The HttpClient class is used to send HTTP requests and receive HTTP responses in Java.

**66. Which class is used to represent an HTTPS connection in Java?**

a) HttpURLConnection

b) URLConnection

c) HttpClient

d) HttpsURLConnection

**Answer: d)** HttpsURLConnection

**Explanation:** The HttpsURLConnection class is used to represent an HTTPS connection in Java.

**67. Which method is used to convert a String to an int in Java?**

a) Integer.parseInt()

b) Integer.valueOf()

c) Double.parseDouble()

d) Double.valueOf()

**Answer: a)** Integer.parseInt()

**Explanation:** The Integer.parseInt() method is used to convert a String to an int in Java.

**68. Which method is used to convert a String to an Integer object in Java?**

a) Integer.parseInt()

b) Integer.valueOf()

c) Double.parseDouble()

d) Double.valueOf()

**Answer: b)** Integer.valueOf()

**Explanation:** The Integer.valueOf() method is used to convert a String to an Integer object in Java.

**69. Which method is used to convert a String to a double in Java?**

a) Integer.parseInt()

b) Integer.valueOf()

c) Double.parseDouble()

d) Double.valueOf()

**Answer: c)** Double.parseDouble()

**Explanation:** The Double.parseDouble() method is used to convert a String to a double in Java.

**70. Which method is used to convert a String to a Double-object in Java?**

a) Integer.parseInt()

b) Integer.valueOf()

c) Double.parseDouble()

d) Double.valueOf()

**Answer: d)** Double.valueOf()

**Explanation:** The Double.valueOf() method is used to convert a String to a Double object in Java.

## Top 70 Java 10 MCQs and Answers | Java 10 Quiz

**1. What is the major feature added in Java 10?**

A) JShell

B) Local Variable Type Inference

C) Lambda Expressions

D) Enhanced for loop

**Answer: B**

**Explanation:** Local Variable Type Inference is the major feature added in Java 10. It allows you to declare a variable without specifying its type explicitly.

**2. Which of the following is not a primitive data type in Java?**

A) boolean

B) byte

C) char

D) string

**Answer: D**

**Explanation:** string is not a primitive data type in JavA) It is a class in the Java standard library.

**3. What is the purpose of the 'final' keyword in Java?**

A) To indicate that a variable cannot be changed after initialization

B) To indicate that a class cannot be inherited

C) To indicate that a method cannot be overridden

D) All of the above

**Answer: D**

**Explanation:** The 'final' keyword can be used to indicate that a variable, class, or method cannot be changed, inherited, or overridden respectively.

**4. What is the default value of a boolean variable in Java?**

A) true

B) false

C) null

D) 0

**Answer: B**

**Explanation:** The default value of a boolean variable in Java is false.

**5. What is the output of the following code snippet?**

go

int x = 5;

int y = x++;

System.out.println(x + " " + y);

A) 5 5

B) 6 6

C) 5 6

D) 6 5

**Answer: C**

**Explanation:** The post-increment operator '++' first assigns the value of x to y and then increments x. Therefore, the value of x becomes 6 and the value of y remains 5.

**6. Which of the following is an example of an unchecked exception in Java?**

A) NullPointerException

B) IOException

C) ClassNotFoundException

D) SQLException

**Answer: A**

**Explanation:** NullPointerException is an example of an unchecked exception in JavA) Unchecked exceptions are not checked at compile-time and are thrown at runtime.

**7. Which of the following is an example of a checked exception in Java?**

A) ArrayIndexOutOfBoundsException

B) IllegalArgumentException

C) RuntimeException

D) FileNotFoundException

**Answer: D**

**Explanation:** FileNotFoundException is an example of a checked exception in JavA) Checked exceptions are checked at compile-time and must be caught or declared in the method signature.

**8. Which of the following is not a valid access modifier in Java?**

A) public

B) private

C) protected

D) internal

**Answer: D**

**Explanation:** internal is not a valid access modifier in JavA) The valid access modifiers are public, private, and protecteD)

**9. What is the difference between a class and an object in Java?**

A) A class is a blueprint for creating objects, whereas an object is an instance of a class

B) A class is an instance of an object, whereas an object is a blueprint for creating classes

C) A class and an object are the same thing

D) None of the above

**Answer: A**

**Explanation:** A class is a blueprint for creating objects, whereas an object is an instance of a class.

**10. What is the purpose of the 'this' keyword in Java?**

A) To refer to the current object

B) To create a new object

C) To refer to a static variable

D) None of the above

**Answer: A**

**Explanation:** The 'this' keyword is used to refer to the current object in JavA)

**11. What is the output of the following code snippet?**

python

Copy code

```
int x = 5;
int y = 2
go
System.out.println(x / y);
```

A) 2.5

B) 2

C) 2.0

D) Error

**Answer: B**

**Explanation:** Integer division in Java always returns an integer. Therefore, the result of x / y is B)

**12. What is the output of the following code snippet?**

javascript

```
String s = “Hello, World!”;
System.out.println(s.substring(7));
```

A) Hello,

B) World!

C) Hello, World!

D) World

**Answer: B**

**Explanation:** The substring method in Java returns the substring of a string starting from the specified index. Therefore, s.substring(7) returns “World!”.

**13. Which of the following is not a loop in Java?**

A) for loop

B) while loop

C) do-while loop

D) if-else loop

**Answer: D**

**Explanation:** if-else is not a loop in JavA) It is a conditional statement.

**14. What is the output of the following code snippet?**

python

```
int[] arr = {1, 2, 3, 4, 5};
for (int i : arr) {
if (i == 3) {
break;
}
System.out.print(i + ” “);
}
```

A) 1 2 3

B) 1 2

C) 1 2 4 5

D) 1 2 4

**Answer: B**

**Explanation:** The break statement in Java terminates the loop immediately. Therefore, the loop terminates when i equals 3 and the output is 1 B)

**15. What is the output of the following code snippet?**

python

```
int[] arr = {1, 2, 3, 4, 5};
for (int i : arr) {
```

```
if (i == 3) {
continue;
}
System.out.print(i + ” “);
}
```

A) 1 2 3

B) 1 2

C) 1 2 4 5

D) 1 2 4

**Answer: C**

**Explanation:** The continue statement in Java skips the current iteration of the loop and moves to the next iteration. Therefore, the loop skips 3 and the output is 1 2 4 5.

**16. What is the output of the following code snippet?**

go

```
int x = 5;
int y = 2;
if (x > y) {
System.out.println(“x is greater than y”);
} else {
System.out.println(“x is less than or equal to y”)
}
```

A) x is greater than y

B) x is less than y

C) x is equal to y

D) x is less than or equal to y

**Answer: A**

**Explanation:** The if-else statement in Java executes the if block if the condition is true and the else block if the condition is false. Therefore, the output is “x is greater than y”.

**17. What is the output of the following code snippet?**

go

```
int x = 5;
int y = 2;
int z = (x > y) ? x : y;
System.out.println(z);
```

A) 5

B) 2

C) 7

D) Error

**Answer: A**

**Explanation:** The ternary operator in Java is a shorthand for the if-else statement. It evaluates the condition and returns the value of the first expression if the condition is true and the value of the second expression if the condition is false. Therefore, the output is 5.

**18. What is the output of the following code snippet?**

go

```
int x = 5;
int y = 2;
int z = x % y;
System.out.println(z);
```

A) 2.5

B) 2

C) 2.0

D) 1

**Answer: D**

**Explanation:** The modulus operator in Java returns the remainder of the division of the first operand by the second operanD) Therefore, the output is A)

**19. What is the output of the following code snippet?**

go

int x = 5;

int y = 2;

int z = x++;

System.out.println(z);

A) 5

B) 6

C) 2

D) Error

**Answer: A**

**Explanation:** The post-increment operator in Java increments the value of the variable after the expression has been evaluateD) Therefore, the output is 5.

**20. What is the output of the following code snippet?**

go

int x = 5;

int y = 2;

int z = ++x + y–;

System.out.println(z);

A) 7

B) 8

C) 9

D) Error

**Answer: C**

**Explanation:** The pre-increment operator in Java increments the value of the variable before the expression has been evaluateD) Therefore, ++x evaluates to 6. The post-decrement operator in Java decrements the value of the variable after the expression has been evaluateD) Therefore, y– evaluates to B) The expression ++x + y– evaluates to 6 + 2 = 8. Therefore, the output is 9.

**21. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = “Hello”;

System.out.println(s1 == s2);

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The == operator in Java compares the references of two objects. In this case, both s1 and s2 refer to the same string literal “Hello”. Therefore, the output is true.

**22. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = new String(“Hello”);

System.out.println(s1 == s2);

A) true

B) false

C) Error

D) None of the above

**Answer: B**

**Explanation:** The == operator in Java compares the references of two objects. In this case, s1 refers to the string literal “Hello” and s2 refers to a new string object created with the new operator. Therefore, the output is false.

**23. What is the output of the following code snippet?**

vbnet

String s1 = “Hello”;

String s2 = new String(“Hello”);

System.out.println(sA)equals(s2));

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The equals method in Java compares the contents of two objects. In this case, both s1 and s2 contain the same string “Hello”. Therefore, the output is true.

**24. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = new String(“Hello”);

System.out.println(sA)compareTo(s2));

A) 0

B) 1

C) -1

D) Error

**Answer: A**

**Explanation:** The compareTo method in Java compares the contents of two strings lexicographically. In this case, both s1 and s2 contain the same string “Hello”. Therefore, the output is 0.

**25. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = new String(“Hello”);

System.out.println(sA)substring(0, 3));

System.out.println(sB)substring(0, 3));

A) Hel Hel

B) Hello Hel

C) Hel Hello

D) Error

**Answer: A**

**Explanation:** The substring method in Java returns the substring of a string starting from the specified index to the end of the string. Therefore, sA)substring(0, 3) and sB)substring(0, 3) both return “Hel”.

**26. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = “World”;

System.out.println(sA)concat(s2));

A) HelloWorld

B) Hello World

C) Error

D) None of the above

**Answer: A**

**Explanation:** The concat method in Java concatenates two strings. Therefore, sA)concat(s2) returns “HelloWorld”.

**27. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = “Hello”;

System.out.println(sA)compareTo(s2));

A) 0

B) 1

C) -1

D) Error

**Answer: A**

**Explanation:** The compareTo method in Java compares the contents of two strings lexicographically. In this case, both s1 and s2 contain the same string “Hello”. Therefore, the output is 0.

**28. What is the output of the following code snippet?**

javascript

String s1 = “hello”;

String s2 = “HELLO”;

System.out.println(sA)equalsIgnoreCase(s2));

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The equalsIgnoreCase method in Java compares the contents of two strings ignoring their case. In this case, s1 contains “hello” and s2 contains “HELLO”. Therefore, the output is true.

**29. What is the output of the following code snippet?**

javascript

String s1 = ”   Hello   “;

System.out.println(sA)trim());

A) Hello

B) Hello

C) Hello

D) Error

**Answer: A**

**Explanation:** The trim method in Java removes leading and trailing white spaces from a string. Therefore, sA)trim() returns “Hello”.

**30. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)replace(‘l’, ‘L’));

A) HeLLo

B) Hello

C) HELLO

D) Error

**Answer: A**

**Explanation:** The replace method in Java replaces all occurrences of a character in a string with another character. Therefore, sA)replace(‘l’, ‘L’) returns “HeLLo”.

**31. What is the output of the following code snippet?**

String s1 = “Hello”;

System.out.println(sA)indexOf(‘l’));

A) 2

B) 3

C) 4

D) Error

**Answer: A**

**Explanation:** The indexOf method in Java returns the index of the first occurrence of a character in a string. In this case, the first ‘l’ occurs at index B) Therefore, the output is B)

**32. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)lastIndexOf(‘l’));

A) 2

B) 3

C) 4

D) Error

**Answer: C**

**Explanation:** The lastIndexOf method in Java returns the index of the last occurrence of a character in a string. In this case, the last ‘l’ occurs at index D) Therefore, the output is D)

**33. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)charAt(1));

A) H

B) e

C) l

D) Error

**Answer: B**

**Explanation:** The charAt method in Java returns the character at the specified index in a string. In this case, the character at index 1 is ‘e’. Therefore, the output is ‘e’.

**34. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)length());

A) 4

B) 5

C) 6

D) Error

**Answer: B**

**Explanation:** The length method in Java returns the number of characters in a string. In this case, the string “Hello” has 5 characters. Therefore, the output is 5.

**35. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)startsWith(“He”));

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The startsWith method in Java checks if a string starts with a specified prefix. In this case, the string “Hello” starts with the prefix “He”. Therefore, the output is true.

**36. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)endsWith(“lo”));

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The endsWith method in Java checks if a string ends with a specified suffix. In this case, the string “Hello” ends with the suffix “lo”. Therefore, the output is true.

**37. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)substring(2));

A) Hel

B) Hello

C) llo

D) Error

**Answer: C**

**Explanation:** The substring method in Java returns a substring of a string starting from the specified index. In this case, sA)substring(2) returns “llo”.

**38. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)substring(2, 4));

A) He

B) Hel

C) ll

D) Error

**Answer: C**

**Explanation:** The substring method in Java with two arguments returns a substring of a string starting from the first index and ending at the second index (exclusive). In this case, sA)substring(2, 4) returns "ll".

**39. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

System.out.println(sA)toUpperCase());

A) HELLO

B) Hello

C) hello

D) Error

**Answer: A**

**Explanation:** The toUpperCase method in Java converts a string to uppercase. In this case, sA)toUpperCase() returns "HELLO".

**40. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

System.out.println(sA)toLowerCase());

A) hello

B) Hello

C) HELLO

D) Error

**Answer: A**

**Explanation:** The toLowerCase method in Java converts a string to lowercase. In this case, sA)toLowerCase() returns "hello".

**41. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

String s2 = "Hello";

System.out.println(s1 == s2);

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** In Java, when two string literals are created with the same value, they refer to the same object in memory. Therefore, s1 and s2 both refer to the same "Hello" string object. The == operator checks if two variables refer to the same object in memory, so s1 == s2 returns true.

**42. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

String s2 = new String("Hello");

System.out.println(s1 == s2);

A) true

B) false

C) Error

D) None of the above

**Answer: B**

**Explanation:** In this case, s1 is a string literal and s2 is a string object created using the new keyworD) When a string object is created using the new keyword, a new object is created in memory even if the value is the same as an existing string. Therefore, s1 and s2 do not refer to the same object in memory, so s1 == s2 returns false.

**43. What is the output of the following code snippet?**

vbnet

String s1 = “Hello”;

String s2 = new String(“Hello”);

System.out.println(sA)equals(s2));

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The equals method in Java checks if two strings have the same value. In this case, s1 and s2 both have the value “Hello”, so sA)equals(s2) returns true.

**44. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = “hello”;

System.out.println(sA)equalsIgnoreCase(s2));

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The equalsIgnoreCase method in Java checks if two strings have the same value ignoring case. In this case, s1 has the value “Hello” and s2 has the value “hello”, but because the case is ignored, sA)equalsIgnoreCase(s2) returns true.

**45. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = “World”;

System.out.println(sA)concat(s2));

A) HelloWorld

B) Hello World

C) Error

D) None of the above

**Answer: A**

**Explanation:** The concat method in Java concatenates two strings together. In this case, sA)concat(s2)

**46. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = “World”;

System.out.println(s1 + s2);

A) HelloWorld

B) Hello World

C) Error

D) None of the above

**Answer: A**

**Explanation:** In Java, the + operator can be used to concatenate two strings. In this case, s1 + s2 returns “HelloWorld”.

**47. What is the output of the following code snippet?**

java

String s1 = “Hello”;

int i = 2;

System.out.println(sA)charAt(i));

A) l

B) e

C) Error

D) None of the above

**Answer: A**

**Explanation:** The charAt method in Java returns the character at a specified index in a string. In this case, sA)charAt(2) returns ‘l’.

**48. What is the output of the following code snippet?**

java

String s1 = “Hello”;

int i = 10;

System.out.println(sA)charAt(i));

A) IndexOutOfBoundsException

B) Error

C) null

D) None of the above

**Answer: A**

**Explanation:** If the index specified in the charAt method is greater than or equal to the length of the string, an IndexOutOfBoundsException is thrown.

**49. What is the output of the following code snippet?**

java

String s1 = “Hello”;

int i = 2;

System.out.println(sA)indexOf(‘l’, i));

A) 2

B) 3

C) -1

D) Error

**Answer: B**

**Explanation:** The indexOf method in Java returns the index of the first occurrence of a specified character in a string, starting from a specified index. In this case, sA)indexOf(‘l’, 2) returns 3 because the first occurrence of ‘l’ after index 2 is at index C)

**50. What is the output of the following code snippet?**

java

String s1 = “Hello”;

int i = 2;

System.out.println(sA)lastIndexOf(‘l’, i));

A) 2

B) 3

C) -1

D) Error

**Answer: A**

**Explanation:** The lastIndexOf method in Java returns the index of the last occurrence of a specified character in a string, starting from a specified index. In this case, sA)lastIndexOf('l', 2) returns 2 because the last occurrence of 'l' before index 2 is at index B)

**51. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

System.out.println(sA)length());

A) 5

B) 6

C) Error

D) None of the above

**Answer: A**

**Explanation:** The length method in Java returns the length of a string, which is the number of characters in the string. In this case, sA)length() returns 5.

**52. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

System.out.println(sA)substring(1));

A) ello

B) Hello

C) H

D) Error

**Answer: A**

**Explanation:** The substring method in Java returns a new string that is a substring of the original string, starting from a specified index. If no end index is specified, the substring extends to the end of the original string. In this case, sA)substring(1) returns "ello".

**53. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

System.out.println(sA)substring(1, 3));

A) el

B) He

C) ll

D) Error

**Answer: A**

**Explanation:** The substring method in Java can also take a second parameter, which specifies the end index of the substring (exclusive). In this case, sA)substring(1, 3) returns "el".

**54. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

String s2 = sA)replace('l', 'z');

System.out.println(s2);

A) Hezzo

B) Helzo

C) H

D) Error

**Answer: A**

**Explanation:** The replace method in Java returns a new string that is the original string with all occurrences of a specified character replaced by another character. In this case, sA)replace('l', 'z') returns "Hezzo".

**55. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

System.out.println(sA)toLowerCase());

A) hello

B) HELLO

C) Hello

D) Error

**Answer: A**

**Explanation:** The toLowerCase method in Java returns a new string that is the original string with all uppercase letters converted to lowercase. In this case, sA)toLowerCase() returns "hello".

**56. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

System.out.println(sA)toUpperCase());

A) HELLO

B) hello

C) Hello

D) Error

**Answer: A**

**Explanation:** The toUpperCase method in Java returns a new string that is the original string with all lowercase letters converted to uppercase. In this case, sA)toUpperCase() returns "HELLO".

**57. What is the output of the following code snippet?**

vbnet

String s1 = "Hello";

String s2 = "hello";

System.out.println(sA)equals(s2));

A) true

B) false

C) Error

D) None of the above

**Answer: B**

**Explanation:** The equals method in Java checks whether two strings are equal in content. In this case, s1 and s2 have different cases, so sA)equals(s2) returns false.

**58. What is the output of the following code snippet?**

javascript

String s1 = "Hello";

String s2 = "hello";

System.out.println(sA)equalsIgnoreCase(s2));

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The equalsIgnoreCase method in Java checks whether two strings are equal in content, ignoring differences in case. In this case, s1 and s2 have different cases, but sA)equalsIgnoreCase(s2) returns true because the comparison is case-insensitive.

**59. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)indexOf(‘l’));

A) 2

B) 3

C) 4

D) None of the above

**Answer: A**

**Explanation:** The indexOf method in Java returns the index of the first occurrence of a specified character in a string. In this case, sA)indexOf(‘l’) returns 2, which is the index of the first “l” in the string.

**60. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)indexOf(‘z’));

A) -1

B) 0

C) Error

D) None of the above

**Answer: A**

**Explanation:** If the specified character is not found in the string, the indexOf method returns -A) In this case, sA)indexOf(‘z’) returns -1 because the character “z” is not in the string.

**61. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)charAt(2));

A) H

B) e

C) l

D) None of the above

**Answer: C**

**Explanation:** The charAt method in Java returns the character at a specified index in a string. In this case, sA)charAt(2) returns “l”, which is the character at index B)

**62. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

String s2 = “World”;

System.out.println(sA)concat(s2));

A) HelloWorld

B) Hello World

C) Error

D) None of the above

**Answer: A**

**Explanation:** The concat method in Java concatenates two strings together, returning a new string that is the combination of the two. In this case, sA)concat(s2) returns “HelloWorld”.

**63. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)endsWith(“lo”));

A) true

B) false

C) Error

D) None of the above

**Answer: A**

**Explanation:** The endsWith method in Java checks whether a string ends with a specified substring. In this case, sA)endsWith(“lo”) returns true because the string “Hello” ends with “lo”.

**64. What is the output of the following code snippet?**

javascript

String s1 = ”   Hello   “;

System.out.println(sA)trim());

A) Hello

B) Hello

C) Error

D) None of the above

**Answer: A**

**Explanation:** The trim method in Java returns a new string that is the original string with all leading and trailing whitespace removeD) In this case, sA)trim() returns “Hello”.

**65. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)isEmpty());

A) false

B) true

C) Error

D) None of the above

**Answer: A**

**Explanation:** The isEmpty method in Java checks whether a string is empty, i.e. has a length of zero. In this case, s1 is not empty, so sA)isEmpty() returns false.

**66. What is the output of the following code snippet?**

javascript

String s1 = “Hello”;

System.out.println(sA)startsWith(“He”));

A) true

B) false

**Answer: A**

**Explanation:** The startsWith method in Java checks whether a string starts with a specified substring. In this case, sA)startsWith("He") returns true because the string "Hello" starts with "He".

**67. What is the output of the following code snippet?**

```
javascript
String s1 = "Hello";
System.out.println(sA)substring(2));
```

A) Hello

B) llo

C) Error

D) None of the above

**Answer: B**

**Explanation:** The substring method in Java returns a substring of a string, starting from a specified index. In this case, sA)substring(2) returns "llo", which is the substring of s1 starting from index B)

**68. What is the output of the following code snippet?**

```
javascript
String s1 = "Hello";
System.out.println(sA)substring(1, 4));
```

A) Hello

B) Hel

C) ell

D) None of the above

**Answer: B**

**Explanation:** The substring method in Java can also take two arguments, specifying a start index and an end index for the substring. The substring returned includes the character at the start index, but not the character at the end index. In this case, sA)substring(1, 4) returns "Hel", which is the substring of s1 starting from index 1 and ending at index C)

**69. What is the output of the following code snippet?**

```
javascript
String s1 = "Hello";
System.out.println(sA)replace('l', 'z'));
```

A) Hezzo

B) Hezzz

C) Hello

D) None of the above

**Answer: A**

**Explanation:** The replace method in Java replaces all occurrences of a specified character in a string with another character. In this case, sA)replace('l', 'z') returns "Hezzo", which is the string "Hello" with all "l" characters replaced with "z".

**70. What is the output of the following code snippet?**

```
javascript
String s1 = "Hello";
System.out.println(sA)replaceAll("l", "z"));
```

A) Hezzo

B) Hezzz

C) Hello

D) None of the above

**Answer: A**

**Explanation:** The replaceAll method in Java replaces all occurrences of a specified substring in a string with another substring. In this case, sA)replaceAll(“l”, “z”) returns “Hezzo”, which is the string “Hello” with all “l” characters replaced with “z”.