Online Training Weekly Quiz-1 Backend Freshers

Points: 160/230

1. Name *

Adarsh.K

2. Email ID *

adarshkashyap1256@gmail.com

✓ **Correct** 10/10 Points

3. What is the primary purpose of Java Virtual Machine (JVM)? *

- ○ To compile Java source code
- ◉ To execute Java bytecode

Feedback:
JVM is responsible for executing Java bytecode, which is an intermediate code generated by the Java compiler. This bytecode is platform-independent and can run on any device with a compatible JVM.

- ○ To interpret Java source code
- ○ To optimize Java programs

✓ **Correct** 10/10 Points

4. Which component is essential for writing and running Java programs? *

- ○ Java Compiler
- ○ Java Virtual Machine (JVM)
- ◉ Java Development Kit (JDK)

  Feedback:
  JDK is required for writing and running Java programs as it includes the Java compiler, JVM, and other tools needed for development.

- ○ Integrated Development Environment (IDE)

✓ **Correct** 10/10 Points

5. What is the purpose of type casting in Java? *

- ◉ Changing the data type of a variable

  Feedback:
  Type casting is used to change the data type of a variable, ensuring compatibility in expressions and assignments.

- ○ Initializing a variable
- ○ Declaring a variable
- ○ Accessing array elements

✓ **Correct** 10/10 Points

6. In Java, what is the purpose of the "switch-case" statement? *

- ○ Iteration over a collection
- ◉ Conditional branching based on multiple values of an expression

Feedback:
The "switch-case" statement is used for conditional branching based on the values of an expression.

- ( ) Executing a block of code repeatedly
- ( ) Handling exceptions

✓ **Correct** 10/10 Points

7. What is the term for using the same method name with different parameter lists within a class? *

- ( ) Method overriding
- (•) Method overloading

  Feedback:
  Method overloading is the practice of using the same method name with different parameter lists within a class.
- ( ) Method inheritance
- ( ) Method polymorphism

✓ **Correct** 10/10 Points

8. How do you access elements in a Java array? *

- (•) Using index notation (e.g., array[0])

  Feedback:
  Elements in a Java array are accessed using index notation, such as array[0] to access the first element.
- ( ) Using dot notation (e.g., array.element)
- ( ) Using arrow notation (e.g., array->element)

Using parentheses (e.g., array(0))

✓ **Correct** 10/10 Points

9. What is the primary purpose of constructors in Java? *

Initializing static variables

Creating objects and initializing instance variables

Feedback:
Constructors in Java are used for creating objects and initializing instance variables.

Inheriting properties from a superclass

Overriding superclass methods

✓ **Correct** 10/10 Points

10. Which block is optional in a Java "try-catch-finally" statement? *

try

catch

finally

Feedback: The "finally" block in a Java "try-catch-finally" statement is optional.

All blocks are mandatory

✗ **Incorrect** 0/10 Points

11. What is the primary purpose of the Stream API in Java file handling? *

Reading and writing binary files

- Reading and writing character-based files
- Handling exceptions in file operations
- Managing file permissions ion 4

✓ **Correct** 10/10 Points

12. Which Java Collection is based on the principle of key-value pairs? *

- ArrayList
- LinkedList
- HashSet
- (●) HashMap

Feedback: HashMap is a Java Collection based on the principle of key-value pairs.

✓ **Correct** 10/10 Points

13. Which operation is commonly performed using the "Map" function in Java Stream API? *

- Filtering elements
- (●) Transforming elements

Feedback:
The "Map" function in Java Stream API is commonly used for transforming elements.

- Reducing elements
- Sorting elements

✓ **Correct** 10/10 Points

14. What is the correct syntax for declaring a variable of integer type in Java? *

- (●) int number;

  Feedback:
  The correct syntax for declaring a variable of integer type in Java is 'int number;'.

- ( ) integer number;
- ( ) variable int;
- ( ) number = int;

✓ **Correct** 10/10 Points

15. Write the command to compile a Java program named "HelloWorld.java" using the command line. *

- ( ) java HelloWorld.java
- (●) javac HelloWorld.java

  Feedback: The correct command to compile a Java program is 'javac HelloWorld.java'.

- ( ) compile HelloWorld.java
- ( ) run HelloWorld.java

✓ **Correct** 10/10 Points

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

```
int x = 5;
double y = 2.5;
System.out.println(x + y); *
```

- (●) 7.5

Feedback:
The output of the code snippet is '7.5' because the integer 'x' is implicitly converted to a double before addition.

- 8
- 7
- 2.5

✓ **Correct** 10/10 Points

17. What is the value of 'result' after the following code execution?

```
int a = 10;
int b = 3;
int result = a % b; *
```

- [x] 1

Feedback:
The value of 'result' is '1' because the modulus operator (%) gives the remainder of the division.

- 0
- 3
- 10

✗ **Incorrect** 0/10 Points

18. Define a method named 'multiply' that takes two integers as parameters and returns their product. *

- [x] public static int multiply(int x, int y) { return x*y; }
- int multiply(int x, int y) { return x*y; }

- public int multiply(int x, int y) { return x*y; }
- int multiply(int x, double y) { return x*y; }

✗ **Incorrect** 0/10 Points

19. Choose the right Java code to find the sum of all elements in the given array.

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

- int sum = 0; for (int num : numbers) { sum += num; } ;
- int sum = 0; for (int num in numbers) { sum += num; } ;
- (selected) int sum = 0; for (int i = 0; i < numbers.length; i++) { sum += i; } ;
- int sum = 0; for (int num : numbers) { i += num; } ;

✓ **Correct** 10/10 Points

20. Choose right lambda expression to represent the following functional interface:

    interface Calculator { int operate(int a, int b); } *

- (selected) Calculator addition = (a, b) -> a + b;

  Feedback:
  The correct lambda expression representing the 'Calculator' interface is 'Calculator addition = (a, b) -> a + b;'.

- (a, b) -> int { return a + b; }
- Calculator addition = (int a, int b) -> a + b;
- (int a, int b) -> { return a + b; }

✕ **Incorrect** 0/10 Points

21. Given the following:

```
1. class Ex1{
2. public static void main(String[] args) {
3. int a[] = {1,2,3,4};
4. System.out.print(a instanceof Object);
5. }
6. }
```

What is the result?

NOTE: The keyword "instanceof" is use to check whether an object is of a particular type. *

- ○ Will produce output as true
- ○ Compilation fails due to error at line 3.
- ◉ Compilation fails due to error at line 4.
- ○ Length of this array is 3.
- ○ Will produce output as false

✕ **Incorrect** 0/10 Points

22. Given the following:

```
1. class Ex1{
2. public static void main(String[] args) {
3. int a[] = { 1,2,053,4};
4. int b[][] = { {1,2,4}, {2,2,1},{0,43,2}};
5. System.out.print(a[3]==b[0][2] );
6. System.out.print("" + (a[2]==b[2][1]));
7. }
8. }
```

What is the result? *

- true true
- Compilation fails.
- true false
- false true
- false false

Incorrect 0/10 Points

23. Given the following:

```
1. class Ex1{
2. int a=10;
3. public static void main(String[] args) {
4. new Ex1().print();
5. }
6. public void print(){
7. int a = 8;
8. System.out.print(a + " ");
9. }
10. }
```

What is the result? *

- 18
- Output is unpredictable.
- Compilation fails.
- 10

8

✓ **Correct** 10/10 Points

24. Given

```
import java.util.ArrayList;
import java.util.List;
public class JavaSETest {
public static void main(String[] args) {
List elements = new ArrayList<>();
elements.add(10);
int firstElmnt = elements.get(1);
System.out.println(firstElmnt);
}
}
```

What is the result? *

- null
- 10
- 0
- Throws IndexOutOfBoundException on runtime.

Feedback: Throws IndexOutOfBoundException on runtime.

✗ **Incorrect** 0/10 Points

25. Given:

```
class Product {
        double price;
}

public class Test {
   public void updatePrice(Product product, double price) {
      price = price * 2;
```

```
        product.price = product.price + price;
    }

    public static void main(String[] args) {
        Product prt = new Product();
        prt.price = 200;
        double newPrice = 100;
        Test t = new Test();

        t.updatePrice(prt, newPrice);
        System.out.println(prt.price + ":" + newPrice);
    }
}
```

What is the result? *

- 200.0 : 100.0
- 400.0 : 200.0
- 400.0 : 100.0
- Compilation fails.
- as newPrice variable is not getting updated. The instance variable price is getting updated