# PROGRAMMING IN JAVA

## Assignment 12

### TYPE OF QUESTION: MCQ

**Number of questions**: 10 Total marks: **10 × 1 = 10**

---

## QUESTION 1:

**Execution of the following SQL command**

```
SELECT * FROM myTable
```

**using JDBC program will return a ResultSet object. This object is:**

a. **Same as the myTable.**

b. **All records in verbatim from the table.**

c. **All records in verbatim from the table but those records with null values.**

d. **All records in verbatim from the table but those records are not with null values.**

**Correct Answer:**

b. **All records in verbatim from the table.**

**Detailed Solution:**

When executing an SQL SELECT query using JDBC, the result is returned as a `ResultSet` object. This `ResultSet` object contains all the records (rows) returned by the SELECT query from the specified table (`myTable` in this case), without any filtering based on null values. Therefore, option b is correct.

---

DriverManager: Manages a list of database drivers and establishes a connection to the database.
Connection: Represents a connection with a specific database. It is the foundation for executing SQL commands.
Statement: Used to execute SQL queries and update commands. JDBC provides different types of statements, including Statement, PreparedStatement, and CallableStatement.
ResultSet: Holds the data retrieved from a database after executing a query, allowing applications to read the data.
SQLException: Handles errors and exceptions that may arise from database operations.

## QUESTION 2:

**Which of the following method is used to set a frame, f with size 300 × 200 pixels?**

```
JFrame f = newJFrame();
```

**a. f.setSize(300, 200);**

**b. f.setSize(200, 300);**

**c. f.paint(300, 200);**

**d. f.setVisible(300, 200);**

**Correct Answer:**

**a. f.setSize(300, 200);**

**Detailed Solution:**

The method `setSize(int width, int height)` is used to set the size of a `JFrame` (or any component) in Java Swing. Therefore, option a is the correct answer to set a `JFrame` named `f` with size 300 × 200 pixels.

## QUESTION 3:

**Consider the following program:**

```
public class Question {
    public static void main(String[] args) {

        String str = "NPTEL - Programming in JAVA - JULY 2024";

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

**a. 38**

**b. 39**

**c. 40**

**d. 41**

**Correct Answer:**

**b. 39**

**Detailed Solution:**

The provided Java program calculates and prints the length of the string `str`, which contains the text "NPTEL - Programming in JAVA - JULY 2024".
The output of this program will be: 39
This is because the string "NPTEL - Programming in JAVA - JULY 2024" consists of 39 characters, including spaces and hyphens.

## QUESTION 4:

**What is the output of the following program?**

```
public class Test {
    public static void aMethod() throws Exception {
        try {
            throw new Exception();
        } finally {
            System.out.print("finally ");
        }
    }

    public static void main(String args[]) {
        try {
            aMethod();
        } catch (Exception e) {
            System.out.print("exception ");
        }
        System.out.print("finished ");
    }
}
```

a. **finally**

b. **exception finished**

c. **finally exception finished**

d. **Compilation fails**

**Correct Answer:**

c. **finally exception finished**

**Detailed Solution:**

The program is syntactically correct and here for two try blocks, there is one catch block.
Here's the step-by-step explanation:

i. The `main` method calls a`Method()`, which throws an `Exception`.
ii. Inside a`Method()`, an `Exception` is thrown in the `try` block.
iii. The `finally` block is always executed, regardless of whether an exception is thrown or not. In this case, it prints "finally ".
iv. Since a`Method()` throws an `Exception`, control moves to the `catch` block in `main`.
v. The `catch` block prints "exception ".
vi. After the `try-catch` block in `main`, "finished " is printed.
vii. Therefore, the complete output is "finally exception finished".

## QUESTION 5:

**What is the output of the following program?**

```
class Program {
    public static void main(String[] args) {
        int counter = 10;
        do {
            System.out.print(2 / counter);
            counter--;
        } while (counter > 2);
    }
}
```

**a. 00000012**

**b. 00000000**

**c. 10011001**

**d. 12211221**

**Correct Answer:**

**b. 00000000**

**Detailed Solution:**

The provided program uses a do-while loop to iterate from counter = 10 down to counter = 3. During each iteration, it prints the result of 2 / counter.

1. counter = 10: 2 / 10 results in 0 (integer division).
2. counter = 9: 2 / 9 results in 0.
3. counter = 8: 2 / 8 results in 0.
4. counter = 7: 2 / 7 results in 0.
5. counter = 6: 2 / 6 results in 0.
6. counter = 5: 2 / 5 results in 0.
7. counter = 4: 2 / 4 results in 0.

When counter = 3: 8. 2 / 3 results in 0.

After counter becomes 2, the loop condition counter > 2 fails, and the loop terminates.

Therefore, the output of the program is 00000000.

## QUESTION 6:

**What should be the value of X and Y for the output of the below program to be 36?**

```
public class Question {
    public static void main(String[] args) {
        int X = 4;
        int Y = 5;
        int sum = 0;
        for (int i = 0; i < X; i++) {
            for (int j = i; j < Y; j++) {
                sum = sum + j;
            }
        }
        System.out.print(sum);
    }
}
```

a. **X = 6 and Y = 5**

b. **X = 2 and Y = 7**

c. **X = 1 and Y = 10**

d. **X = 4 and Y = 5**

**Correct Answer:**

d. **X = 4 and Y = 5**

**Detailed Solution:**

The following is the output of the program for different values of X and Y:

a. *when* X = 6 and Y = 5 *output is* 40

b. *when* X = 2 and Y = 7 *output is* 42

c. *when* X = 1 and Y = 10 *output is* 45

d. *when* X = 4 and Y = 5 *output is* 36

## QUESTION 7:

**Which of the following options correctly initializes the elements of the numbers array with values 1, 2, 3, 4, and 5?**

```
public class NPTEL {
    public static void main(String[] args) {
        int[] numbers = new int[5];
        // #1 : Missing code block
        System.out.println("First element: " + numbers[0]);
    }
}
```

a. **numbers = {1, 2, 3, 4, 5};**

b. 
```
for (int i = 1; i < numbers.length; i++) {
    numbers[i] = i;
}
```

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

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

**Correct Answer:**

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

**Detailed Solution:**

`numbers = new int[]{1, 2, 3, 4, 5};` is the correct answer because it initializes the `numbers` array with values 1, 2, 3, 4, and 5 using array initializer syntax.

## QUESTION 8:

**Which of the following statements are correct and would NOT cause a compilation error?**

i. `float[] = new float(3);`
ii. `float f1[] = new float[];`
iii. `float[] f2 = new float[3];`
iv. `float f3[] = new float[3];`
v. `float f4[]= { 1.0f, 2.0f, 2.0f };`
vi. `float f5[] = new float[] { 1.0f, 2.0f, 3.0f};`

a. **iii, iv, v, vi**

b. **i, ii, iii, iv**

c. **ii, iii, v, vi**

d. **i, ii, iv, vi**

**Correct Answer:**

a. **iii, iv, v, vi**

**Detailed Solution:**

Option iii, iv, v and vi are syntactically correct for declaration of an array.

## QUESTION 9:

**What will be the output of this program?**

```
public class NPTEL {
    public static void main(String[] args) {
        String str1 = "Hello";
        String str2 = "Hello";
        String str3 = new String("Hello");
        System.out.print((str1 == str2) + " ");
        System.out.print(str1 == str3);
    }
}
```

a. **true false**

b. **false true**

c. **true true**

d. **false false**

**Correct Answer:**

a. **true false**

**Detailed Solution:**

`str1` and `str2` are string literals and will be interned to the same memory location, so `str1 == str2` will be true. However, `str3` is created using the new keyword, so it will be stored in a different memory location, leading `str1 == str3` to be false.

## QUESTION 10:

**What will be the output of this program?**

```
public class NPTEL {
    public static void main(String[] args) {
        try {
            int num = 10 / 0;
            System.out.println(num);
        } catch (ArithmeticException e) {
            System.out.println("Arithmetic exception occurred");
        } finally {
            System.out.println("Finally block executed");
        }
    }
}
```

a. **Compilation ERROR**

b. **"Finally block executed"**

c. **"Arithmetic exception occurred**
**Finally block executed"**

d. **Runtime ERROR**

**Correct Answer:**

c. **"Arithmetic exception occurred**
**Finally block executed"**

**Detailed Solution:**

The division by zero will throw an `ArithmeticException`, which will be caught in the `catch` block. Then, the `finally` block will be executed.