

The Test is divided into three sections. The first section has MCQ's on Java, each question carries 2 marks, And the second section has Queries on MySQL, each question carries 3 marks. For the third section, three questions need to be completed, each question carries 25 marks.

Time Duration: 3hours

Section-1: Core Java MCQ's

Question 1: Given

```
class Person{
    String name="No name";
    public Person(String nm)
    {
        name=nm;
    }
}
class Student extends Person{
    String rollNo="5678";
    public Student(String rno)
    {
        rollNo=rno;
    }
}
public class Main
{
    public static void main(String[] args) {
        Student student=new Student("2967");
        System.out.println(student.rollNo);
    }
}
```

What is the Result

- A. 5678
- B. 2967
- C. Compilation fails because of an error in program.
- D. An Exception is thrown at run time.

Question 2: Given

```

class Particle {
    Particle() {
        System.out.print("particle ");
    }
}

class Stone extends Particle {
    Stone(String type) {
        System.out.print(type);
    }
}

public class Hill extends Stone {
    Hill() {
        super("slate ");
        new Stone("slate ");
    }

    public static void main(String[] args) {
        new Hill();
    }
}

```

What is the result?

- A. Compilation fails.
- B. particle slate
- D. particle slate slate
- E. particle slate particle slate

Question 3: Given

```
public class Example {  
    public static void parse(String str) {  
        float result = 0;  
        try {  
            result = Float.parseFloat(str);  
        } catch (NumberFormatException nfe) {  
            result = 0;  
        } finally {  
            System.out.println(result);  
        }  
    }  
  
    public static void main(String[] args) {  
        parse("invalid");  
    }  
}
```

What is the result?

- A. 0.0
- B. Compilation fails.
- C. A ParseException is thrown by the parse method at runtime.
- D. A NumberFormatException is thrown by the parse method at runtime.

Question 4: Given

```

abstract class Transport {
    public int getSpeed() {
        return 0;
    }
}

class Bus extends Transport {
    public int getSpeed() {
        return 40;
    }
}

class HighSpeedTrain extends Bus {
    public int getSpeed() {
        return 200;
    }
}

public class Main {
    public static void main(String[] args) {
        HighSpeedTrain train = new
HighSpeedTrain();
        Bus bus = new HighSpeedTrain();
        Transport transport = new
HighSpeedTrain();
        System.out.println(train.getSpeed() +
", " + bus.getSpeed() + ", " +
transport.getSpeed());
    }
}

```

What is the result?

- A. 0, 0, 0
- B. 200, 40, 0
- C. Compilation fails.
- D. 200, 200, 200
- E. An exception is thrown at runtime.

Question 5. Given:

```
StringBuilder sb1 = new StringBuilder("123");  
String s1 = "123";  
// Insert code here  
System.out.println(sb1 + " " + s1);
```

Which code fragment, inserted at line 24, outputs "123abc 123abc"?

- A. sb1.append("abc"); s1.append("abc");
- B. sb1.append("abc"); s1.concat("abc");
- C. sb1.concat("abc"); s1.append("abc");
- D. sb1.concat("abc"); s1.concat("abc");
- E. sb1.append("abc"); s1 = s1.concat("abc");
- F. sb1.concat("abc"); s1 = s1.concat("abc");

Section-2: MySQL Queries

Question 1:

Consider a database for tracking product sales. There are two tables:

Example Records in Products Table:

ProductID	ProductName	CategoryID	Price
1	Laptop	101	899.99
2	Smartphone	102	599.99
3	T-shirt	201	19.99
4	Novel	301	12.50
5	Coffee Maker	103	149.99

Example Records in Sales Table:

SaleID	ProductID	SaleDate	Quantity
1	1	2023-10-26 09:15:00	2
2	2	2023-10-26 10:30:00	5
3	3	2023-10-26 11:45:00	3
4	1	2023-10-26 13:00:00	1
5	4	2023-10-26 14:15:00	2

Query

Write an SQL query that calculates the total sales revenue for each product category and lists the category name and total revenue. Sort the result in descending order of total revenue.

Ans: _____

Question 2:

In an HR database, there are two tables: "Employees" and "Departments."

Employees Table

Employees Table:

EmployeeID	FirstName	LastName	DepartmentID
1	John	Smith	101
2	Sarah	Johnson	102
3	Michael	Williams	101
4	Emily	Brown	103
5	David	Jones	102

Departments Table

DepartmentID	DepartmentName
101	HR
102	Sales
103	IT
104	Marketing

Query:

Write an SQL query to find the department with the highest average employee salary and list the department name and average salary. Consider only departments with at least five employees.

Ans: _____

Question 3:

In an HR database with "Employees" and "Departments" tables (Use the above tables structure only), write an SQL query to retrieve a list of employees who have not been assigned to any department.

Ans: _____

Question 4:

Consider a product inventory database. There are two tables:

Products

ProductID	ProductName	SupplierID	Price
1	Laptop	101	899.99
2	Smartphone	102	599.99
3	T-shirt	103	19.99
4	Novel	104	12.50
5	Coffee Maker	101	149.99

Suppliers

SupplierID	SupplierName
101	ABC Electronics
102	XYZ Gadgets
103	Fashion Corp
104	Book World
105	Home Appliances Co.

Query:

Write an SQL query to find the names of products that are supplied by more than one supplier, and list the product names.

Ans: _____

Question 5:

Consider a database with "Students" and "Enrollments" tables.

Students Table

StudentID	FirstName	LastName
1	John	Smith
2	Sarah	Johnson
3	Michael	Williams
4	Emily	Brown
5	David	Jones

Enrollments Table

EnrollmentID	StudentID	CourseName	Grade
1	1	Math	A

2	1	Science	B
3	1	History	A
4	2	Math	A
5	2	Science	B
6	3	History	A
7	3	Science	A
8	4	Math	B
9	4	Science	C
10	5	Math	A
11	5	Science	A
12	5	History	A

Query:

Write an SQL query to find the students who are enrolled in all courses offered.

Ans: _____

Section-3: Coding Challenges

Challenge 1:

You are an intrepid explorer on a quest to uncover a hidden treasure that is believed to be buried deep in the heart of a mysterious island. To your advantage, you've found an ancient map filled with cryptic clues and riddles, each associated with a unique positive integer. As you journey further into the island's dense forest, you realize that some of the numbered clues have been scattered and lost in the wilderness.

You must solve the puzzle and determine the smallest missing positive integer to unlock the treasure's secret location. To aid you in this adventurous endeavor, you decide to write a Java program using the remaining clues found on your map.

Sample Input:

You carefully examine the remaining clues you've collected from your map, each represented by a positive integer.

Remaining Clues: [12, 3, 8, -1, 9, 1, 2, 6, 5]

Sample Output:

After running your program and following the clues, you uncover the hidden treasure's location, which corresponds to the smallest missing positive integer.

Congratulations! You've found the treasure. The smallest missing positive integer is: 4

Challenge 2:

You are tasked with designing the core classes for an online shopping system. The system involves various types of products, including electronics, clothing, and books. Your goal is to apply OOP principles to represent this system effectively and include a custom exception for out-of-stock products.

Product Hierarchy: Create a class hierarchy for products, with a base class `Product` and derived classes for different product types (e.g., `Electronics`, `Clothing`, and `Books`). Each product type should have attributes such as a unique product ID, name, description, price, quantity, and any type-specific attributes.

Polymorphic Methods: Implement polymorphic methods like `calculatePrice` and `displayDetails` that work differently for each product type.

Shopping Cart: Model a shopping cart class that can hold various products. The cart should allow customers to add products, and each product should have a specified quantity in stock. If a product is out of stock (quantity is zero), raise a custom `OutOfStockException`.

Custom Exception: Implement a custom exception, `OutOfStockException`, which is raised when an out-of-stock product is added to the cart.

Sample Input and Output:

The user interacts with an online shopping system. The system offers the following options:

1. Add a product to the shopping cart
2. Display the shopping cart
3. Calculate the total cart value
4. Quit the program

Select an option (1-4):

Positive Test Cases:

Adding Products to the Cart (Option 1):

Positive Test Case 1 (Adding an Electronic Product)

Input:

1

E12345

Smartphone

Description: The latest smartphone model with advanced features.

Price: \$599.99

Quantity in Stock: 5

Output (Success):

Product added to the shopping cart.

Positive Test Case 2 (Adding a Clothing Product)

Input:

1

C67890

Jeans

Description: Slim-fit jeans made of high-quality denim.

Price: \$49.99

Quantity in Stock: 10

Output (Success):

Product added to the shopping cart.

Displaying the Shopping Cart (Option 2):

Input:

2

Output (With Products):

Shopping Cart:

1. Product ID: E12345

Name: Smartphone

Description: The latest smartphone model with advanced features.

Price: \$599.99

In Stock: 5

2. Product ID: C67890

Name: Jeans

Description: Slim-fit jeans made of high-quality denim.

Price: \$49.99

In Stock: 10

Calculating Total Cart Value (Option 3):

Input:

3

Output (Total Cart Value):

Total Cart Value: \$659.00

Negative Test Case 2 (Calculating Total Cart Value with Out-of-Stock Product)

Input:

1

E12345

Smartphone

Description: The latest smartphone model with advanced features.

Price: \$599.99

Quantity in Stock: 5

1

B98765

Tablet

Description: High-performance tablet with an out-of-stock issue.

Price: \$349.99

Quantity in Stock: 0

Output (OutOfStockException during Calculation):

Out of stock! Product cannot be added to the shopping cart.

Challenge 3:

Scenario: Restaurant Menu Management System

As the owner of a restaurant, you need to manage your restaurant's menu and related operations. You have a system that stores menu items (item ID, name, price) in an ArrayList of MenuItem objects. Your task is to create a Java program that helps you manage your restaurant's menu efficiently.

Create a console-based menu-driven program that allows the user to perform the following operations:

1. Add new menu items
2. Calculate the total value of all items on the menu
3. Remove a menu item by its item ID
4. Update the price of a menu item by its item ID
5. Search for menu items by name
6. List all menu items
7. Quit the program

If no menu items are found for the given item name during a search, the program should display a message indicating that no items were found.

Additionally, implement a mechanism for tracking the popularity of each menu item. Keep a record of the number of times each item is ordered.

Each menu option should call the corresponding method and display the results or appropriate error messages.

Sample Input and Output:

Sample Input:

1. Add new menu items
2. Remove a menu item by its item ID
3. Update the price of a menu item by its item ID
4. Search for menu items by name
5. List all menu items
6. Quit the program Select an option (1-6):

Sample Output (Adding Menu Items):

Enter item ID: 1001 Enter item name: Spaghetti Carbonara Enter item price: \$15.99 Menu item added successfully!

Sample Output (Removing a Menu Item):

Enter item ID to remove: 1002 Menu item removed successfully!

Sample Output (Updating Menu Item Price):

Enter item ID to update: 1003 Enter new item price: \$12.50 Menu item price updated successfully!

Sample Output (Searching for Menu Items by Name):

Enter item name to search: Margherita Pizza No menu items found with the name "Margherita Pizza."

Sample Output (Listing All Menu Items):

Item ID: 1001,

Item Name: Spaghetti Carbonara,

Price: \$15.99 Item ID: 1004

Item Name: Caesar Salad

Price: \$7.95

Challenge 4

You are developing a text analysis utility that requires the identification of the longest palindromic substring within a given text. A palindromic substring is a sequence of characters that reads the same forwards and backwards. Your task is to create a Java program to find the longest palindromic substring within a given string.

Requirements:

Input String:

Users will input a string of characters for analysis. This string may include letters, numbers, and special characters.

Your program should identify and return the longest palindromic substring within the given input string.

Sample Input and Output:

Sample Input: "babad"

Sample Output: "bab"

Sample Input: "cbbd"

Sample Output: "bb"

Challenge 5:

Imagine you are working for a tech startup that specializes in data transmission and storage solutions. Your team has been tasked with developing a text compression algorithm to optimize data transfer and storage for various applications.

Requirements:

Input String: Users will input a text string, which may contain both uppercase and lowercase alphabetic characters along with other characters.

Compression: Your program needs to analyze the input string and identify consecutive sequences of the same character.

For each consecutive sequence, the program should represent it as a character-count pair.

Build a compressed string using these character-count pairs while maintaining the order of characters from the input string.

Output: Display the compressed string, where each character-count pair represents a character and the number of times it appears consecutively in the input string.

Sample Input and Output:

Input: "aaabbbcc"

Output: "a3b3c2"

Input: "rrrraaaaaawwwwtttt"

Output: Compressed String: "r4a5w3t4"

Input: "hello world"

Output: Compressed String: "h1e1l2o1 1w1o1r1l1d1"