Giri’s Tech Hub Pvt.Ltd, Pune

Programming (Machine) Test

Batch: Nov-24 to March-25 — **Date: 11/08/2025**
Time: 02:00 to 05:00 Pm

Instructions: **Total:- 10 Marks**

1. Solve any 9 questions.
2. Input should be from user.
3. Indentation and comments mandatory.
4. Each program 1 Marks and all comments 1 Marks.

Q1. Write a java program to print 1 to nth happy number.

Q2. Write a java program to print this pattern.

```
        1

      1 0 1

    2 1 0 1 2

  3 2 1 0 1 2 3

4 3 2 1 0 1 2 3 4
```

Q3. Write a java program to find the largest missing number from an integer array.

Input Array : arr = {3, 7, 1, 9, 4}
Range: From 1 to 9 All numbers in range: 1 2 3 4 5 6 7 8 9
Present in array: 1 3 4 7 9 Missing numbers: 2, 5, 6, 8
Output : 8 (This is the largest missing number)

Q4. Write a java program to create pojo class name as Employee with the following properties as id, name, email, salary to perform.

Case 1: Add 5 Records Of Employee.
Case 2: Display All Employee Details.
Case 3: Update Employee Record By Name.
Case 4: Delete Employee Record By salary.
Case 5: Search Employee Record By Id.

Q5. Write a java Program to calculate overtime pay of 5 employees. The overtime pay rate is Rs.50/- (per Hour). Daily shift hour time is only 8 hours.

Note- for a week only 40 hours of working are allowed.
1. Create class Employee with data member ID, Name, total working, salary, overtime Set Information by using a parameterized constructor and create a display Information() method to display all information with salary.
2. Create another class name as OverTime with method setEmployee(Employee emp[ ]) and

void calculateOvertime() to calculate overtime.

Q6. Create class name as ArrayOperation with method name as setArray() and create its Two child classes name as Unique , MergeArray. We need to inherit the ArrayOperation class in Unique , MergeArray and create method. and write the logic.

1. Unique Class : -
   Input array elements: 1, 2, 3, 5, 1, 5, 20, 2, 12, 10
   Output :
   All unique elements in the array are: 3, 20, 12, 10
2. MergeArray class :-
   Input -First Array :- 1 2 3 4 5
   Second Array :- 6 7 8 9 10
   Output - 1 10 2 9 3 8 4 7 5 6

Q7. Problem Statement:
Create an abstract class Student with attributes roll number, name, and an array of marks (5 subjects).
Create an interface ResultOperations with methods calculateTotal(), calculatePercentage(), and assignGrade().

- Implement UGStudent and PGStudent classes with grading rules:
  - UG: Pass if percentage ≥ 40%
  - PG: Pass if percentage ≥ 50%

Additional Requirements:

1. Store details for N students in an array.
2. Display:
   - List of passed and failed students separately.
   - Top 3 students by percentage.
   - Average marks in each subject.

Explanation:
Covers abstraction for common structure, interface for calculations, array processing for N students, sorting for top students, and subject-wise aggregation.

Q8. Create a Java program to process a range of numbers using multithreading.
Requirements:

1. Accept a number N from the user.
2. Create two threads:
   - EvenThread: Prints all even numbers from 1 to N and calculates their sum.
   - OddThread: Prints all odd numbers from 1 to N and calculates their product.
3. Use Thread.join() to ensure both threads complete before the main thread prints results.
4. Display the sum of even numbers and product of odd numbers at the end.

Logic Operations Involved:

- Thread creation and execution order
- Mathematical sum and product calculations
- Thread coordination

Q9. Write a Java program using ArrayList, HashMap, and TreeMap to:

1. Store student names and their marks in 3 subjects.
2. Calculate total marks for each student and store in a HashMap.
3. Sort students in ascending order of their total marks using a TreeMap.
4. Display only those students whose average marks are greater than 60.
5. Remove students who have scored less than 40 in any subject from the list and re-display the result.

Q10. Write a Java program that:

- Accepts a list of strings from the user.
- Uses Queue to store the input order.
- Uses Stack to check whether each word is a palindrome.
- Stores palindromes in a TreeSet (sorted order).
- Prints:
    1. All palindromes in alphabetical order.
    2. Count of palindromes.

Explanation:
This tests:

- Queue for maintaining insertion order.
- Stack for palindrome checking logic.
- TreeSet for storing sorted unique results.

-----ALL THE BEST-----