Giri's Tech Hub, Pune.

Core Java Machine Test

Batch: July-25

Date : 11/10/2025
Time : 02 to 05 Pm.

Instructions:

1. Solve any 9 questions.
2. Input should be from user.
3. Indentation and comments mandatory.
4. Each program 1 marks & all comments 1 marks.
5. Without using any inbuilt functions.

Q1. Write a java program to print this pattern.

```
1
2     *     2
3     *     3     *     3
4     *     4     *     4     *     4
3     *     3     *     3
2     *     2
1
```

Q2. Write a java program to find the frequency of each digit in a given integer.

Q3. Write a java program to check number Is armstrong number or not armstrong using function.

Q4. Write a java program to display following series using function recursion.

1 4 9 16 25 36 49 64 81 100

Q5. Write a java program to find the maximum product of two integers in a given array of integers.

Example:
Input :
nums = { 2, 3, 5, 7, -7, 5, 8, -5 }
Output:
Pair is (7, 8), Maximum Product: 56

Q6. Write a Java program to find the kth highest value from an integer array using unsorted array.

Example :
Input: Array : [15, 8, 22, 4, 10, 18] k = 2
Output : 2nd highest value = 18

Q7. Write a java program to find the unique value from array.

Input array elements: 1, 2, 3, 5, 1, 5, 20, 2, 12, 10
All unique elements in the array are: 3, 20, 12, 10

Q8. Create a POJO class Employee with the following private fields:

- **int id**
- **String name**
- **double salary**

Then, perform the following operations without using any constructor:

1. **Use setter methods to assign values to id, name, and salary.**
2. **Write a method calculateBonus() that returns a bonus amount based on the employee's salary as per the given conditions:**
    - **If salary ≥ 100000 → bonus = 15% of salary**
    - **If salary ≥ 50000 and < 100000 → bonus = 10% of salary**
    - **If salary < 50000 → bonus = 5% of salary**
3. **Write another method displayEmployeeDetails() that prints:**
    - **Employee ID**
    - **Employee Name**
    - **Salary**
    - **Bonus (calculated using calculateBonus())**
    - **Total Compensation (salary + bonus)**
4. **In the main method, create an object of Employee,**
    - **set data using setter methods,**
    - **call calculateBonus() and displayEmployeeDetails() to show all employee information.**

Q9.Implement a program to calculate product discounts based on price using POJO class and a separate service class for logic.

Requirements:

1. POJO Class – Product

- **Fields (private):**
    - **int productId**
    - **String productName**
    - **double price**
- **Methods:**
    - **Public getter and setter methods for each field.**
- **Note: Do not use a constructor; values must be set using setters.**

2. Logic Class – ProductService

- **Method 1: calculateDiscount(Product p)**
    - **Calculates discount based on the product price:**
        - **Price ≥ 1000 → Discount = 20% of price**
        - **500 ≤ Price < 1000 → Discount = 10% of price**
        - **Price < 500 → Discount = 5% of price**
    - **Returns the discount amount.**

- **Method 2: displayProductDetails(Product p)**
  - **Displays:**
    - **Product ID**
    - **Product Name**
    - **Price**
    - **Calculated Discount**
    - **Final Price after discount (Price – Discount)**

3. Main Method Instructions:

1. **Create a Product object.**
2. **Assign values to the object using setter methods.**
3. **Use ProductService to:**
   - **Calculate discount using calculateDiscount().**
   - **Display product details using displayProductDetails().**

Q10. Implement a program to calculate allowances, deductions, and net salary for employees using a POJO class and a service class.

Requirements:

1. POJO Class – Employee

- **Fields (private):**
  - **int empId**
  - **String name**
  - **double basicSalary**
  - **double hra (House Rent Allowance)**
  - **double tax**
- **Methods:**
  - **Getter and setter methods only (do not use constructors).**

2. Logic Class – EmployeeService

- **Method 1: calculateAllowances(Employee e)**
  - **Calculate HRA:**
    - **Basic Salary ≥ 100000 → HRA = 25% of basicSalary**
    - **50000 ≤ Basic Salary < 100000 → HRA = 20% of basicSalary**
    - **Basic Salary < 50000 → HRA = 15% of basicSalary**
- **Method 2: calculateTax(Employee e)**
  - **Tax deduction based on total salary (basicSalary + HRA):**
    - **≥ 150000 → Tax = 20%**
    - **75000 ≤ total < 150000 → Tax = 10%**
    - **< 75000 → Tax = 5%**
- **Method 3: displayEmployeeDetails(Employee e)**
  - **Display: empId, name, basicSalary, HRA, Tax, and Net Salary (basicSalary + HRA – Tax)**

3. Main Method Instructions:

- **Create an Employee object and set values using setters.**
- **Use EmployeeService to calculate allowances, tax, and display employee details.**

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