**Grand Test-3**

This is Grand Test-3 which consists of 5 coding challenges. You are required to attempt any 3 questions out of the 5. Each question carries 30 marks. If you attempt the 4th question, you will receive an additional bonus of 10 points.

Duration: 3 hours

**Challenge-1**

Create a program that maintains an array of Student objects. Each Student object should have a studentId, studentName, rollNo, mobileNumber, marks, and address. The program should allow the user to input the data for each student dynamically with validations. After inputting the data for five students, the program should display the data for all five students.

1. studentId should be a unique id and id should contains 5-digit number only.
2. Name should contain only letters and should be between 2 and 30 characters long.
3. rollNo should be a positive integer.
4. Mobile should contains only 10 digits and should starts with 9 or 8 or 7 or 6.
5. Marks should be a positive integer, and should a two digits number.
6. Address should be between 5 and 100 characters long.

To complete this task, you'll need to write Java code that includes the following:

Define the Student class with the required fields (studentId, studentName, RollNo, mobileNumber, marks, and address) and create setter and getter methods.

1. Create an array of Student objects with a length of 5.
2. Prompt the user to input the data for each student dynamically.
3. Store the inputted data in the corresponding Student object in the array.
4. Display the data for all five students after all data has been inputted.

Here's an example input and output for the program:

-----------------------------------------------------------

Sample Input:

-------

Enter how many students do you want to add : 2

Enter data for student 1:

Student ID: 12345

Name: John Smith

Roll number: 101

Mobile: 934637673

Marks: 87.5

Address: 123 Main St, Anytown USA

Enter data for student 2:

Student ID: 56784

Name: Jane Doe

Roll number: 102

Mobile: 928972973

Marks: 92.0

Address: 456 Elm St, Anytown USA

Output:

--------

Student ID: 12345

Name: John Smith

Roll number: 101

Mobile: 934637673

Marks: 87.5

Address: 123 Main St, Anytown USA

Student ID: 56784

Name: Jane Doe

Roll number: 102

Mobile:  928972973

Marks: 92.0

Address: 456 Elm St, Anytown USA

**Challenge-2**

Suppose you are working on a project that requires you to count the number of elements in an array that have a specific property, such as being an even number, odd number, prime number, and palindrome. You decide to create a program to automate this process.

Write a Java program to count the number of elements in an array that have a specific property. The program should prompt the user to enter the size of the array and then input the elements of the array dynamically. After inputting the array elements, the program should count the number of elements that have each specific property and display the results.

For example, if the user enters an array of size 5 with the elements {3, 2, 5, 7, 121}, the program should output:

Number of even elements: 1

Number of odd elements: 4

Number of prime elements: 4

Number of palindrome elements: 1

**Challenge-3**

You are working for a publishing company that sells books and magazines. The company is interested in tracking the orders and sales of their publications. As a software developer, your task is to create a program that can calculate the total number of copies of a given book that have been ordered and the total sale of the publication.

Task: Write a program in Java that takes input for a book's title, price, number of copies sold, and number of copies ordered. The program should then calculate the total number of copies of the book that have been ordered and the total sale of the publication. The program should also display the results to the user.

A. Create a class called Publication with the following instance variables:

1. title (String)

2. price (double)

3. copies (int)

B. Create a class called Book that extends Publication and adds the following instance variable:

1. author (String)

C. Create a class called Magazine that extends Publication and adds the following instance variables:

1. orderQty (int)

2. currentIssue (String)

D. Add a method called saleCopy() to the Publication class that calculates and returns the total sale of the publication.

Add a method called orderCopies() to the Book class that takes an integer parameter representing the number of copies ordered and adds it to the total number of copies of the book.

Add a method called receiveNewIssue() to the Magazine class that takes a String parameter representing the new issue and updates the current issue of the magazine.

In the main method of the program, prompt the user to enter the book's title, price, number of copies sold, and number of copies ordered.

Create an instance of the Book class with the user's input and call the orderCopies() method with the number of copies ordered.

Calculate the total number of copies of the book that have been ordered and the total sale of the publication using the saleCopy() method.

Display the total number of copies of the book that have been ordered and the total sale of the publication to the user.

Example Output:

Enter title: Introduction to Java Programming

Enter price: 59.99

Enter copies sold: 1000

Enter copies ordered: 500

Total copies of book ordered: 1500

Total sale of publication: 59990.0

**Challenge-4:**

You have been tasked with developing a payroll system for a company. The company has various types of employees such as programmers, team leads, assistant project managers and project managers. All employees have some common attributes like Emp\_name, Emp\_id, Address, Mail\_id and Mobile\_no. In addition, all employees have a Basic Pay (BP) which is used to calculate their salary.

The company has a fixed salary structure for all employees which includes 97% of BP as DA, 10% of BP as HRA, 12% of BP as PF and 0.1% of BP for staff club fund. Your task is to design and implement an inheritance hierarchy to represent the employees and their salaries.

Your program should allow the user to input the employee details such as name, ID, address, email and mobile number. Additionally, the program should allow the user to input the Basic Pay for each employee. The program should then generate the pay slip for each employee, showing their gross and net salary.

Write a Java program that implements the above scenario using inheritance.

**Challenge-5:**

You are tasked with developing a banking solution that allows customers to create accounts, deposit and withdraw money, check their balance, and view their account information. As a software developer, your task is to create a program that can perform these operations using object-oriented programming concepts.

Task: Write a Java program that implements a Bank class and an Account class. The Bank class should contain methods for creating new accounts, depositing and withdrawing money, and checking daily withdrawal limits. The Account class should store information about individual customer accounts, such as the account number, account holder name, and current balance.

Instructions:

Create an Account class with the following instance variables:

1. accountNumber (String)

2. accountHolderName (String)

3. balance (double)

Create a Bank class that contains an Object Array of Account objects, which will be used to store the account information.

Add a method called createAccount() to the Bank class that takes the account holder name and initial balance as parameters and creates a new Account object with a unique account number.

Add a method called deposit() to the Account class that takes an amount parameter and adds it to the account balance.

Add a method called withdraw() to the Account class that takes an amount parameter and subtracts it from the account balance. The method should also check if the withdrawal amount exceeds the daily withdrawal limit, which should be set to a default value of $500.

Add a method called checkBalance() to the Account class that returns the current balance of the account.

Add a method called displayInfo() to the Account class that displays the account number, account holder name, and current balance of the account.

In the main method of the program, prompt the user to select an operation:

1 to create a new account

2 to deposit money

3 to withdraw money

4 to check the balance

5 to display account information

6 to exit the program

Based on the user's selection, call the appropriate method in the Bank or Account class.

Repeat steps 6 until the user chooses to exit the program.

Example Output:

Welcome to the Bank Management System!

Please select an operation:

Create an account

Deposit money

Withdraw money

Check the balance

Display account information

Exit the program

Enter your choice: 1

Enter account holder name: John Doe

Enter initial balance: 1000

Account created successfully!

Account number: 1001

Account holder name: John Doe

Current balance: 1000.0

Please select an operation:

Create an account

Deposit money

Withdraw money

Check the balance

Display account information

Exit the program

Enter your choice: 2

Enter account number: 1001

Enter amount to deposit: 500

Amount deposited successfully!

Current balance: 1500.0

Please select an operation:

Create an account

Deposit money

Withdraw money

Check the balance

Display account information

Exit the program

Enter your choice: 3

Enter account number: 1001

Enter amount to withdraw: 800

Amount withdrawn successfully!

Current balance: 700.0

Please select an operation:

Create an account

Deposit money

Withdraw money

Check the balance

Display account information

Exit the program

Enter your choice: 4

Enter account number: 1001

Current balance: 700