CHRIST (Deemed to be University)

Department of Computer Science

Master of Artificial Intelligence and Machine Learning

Course: MAI271 – JAVA Programming
Exercise No: LAB Exercise – 3
Date: 30 – 10 – 2024
Duration: 2 Hrs

Question 1:

You have been tasked with designing an employee payroll system for CHRIST (Deemed to be University) in Java. The system is intended to efficiently manage various employee roles with diverse payment structures, utilizing object-oriented programming principles for flexibility and scalability.

Base Class: Employee

- Develop a class named Employee with essential attributes:
  - employeeId (integer): unique employee identification number
  - employeeName (String): full name of the employee
  - designation (String): job title or role within the university
- Implement appropriate methods for setting and retrieving these attributes, ensuring adherence to professional coding standards.

Derived Classes:

- Create specialized classes, HourlyEmployee and SalariedEmployee, both extending the Employee class.
- For HourlyEmployee, introduce additional attributes:
  - hourlyRate (double): compensation per hour
  - hoursWorked (int): total hours worked in a week
- For SalariedEmployee, include an extra attribute:
  - monthlySalary (double): fixed monthly compensation

Functionality:

- Implement methods in each derived class to calculate the weekly salary:
  - HourlyEmployee calculates weekly earnings as hourlyRate * hoursWorked.
  - SalariedEmployee determines the weekly salary as monthlySalary / 4 (assuming a 4-week month).
- Design methods for displaying detailed employee information, encompassing the calculated weekly salary.

- Employ a cohesive method named calculateBonus() in the base class Employee, overridden in the derived classes, facilitating bonus computation tailored to each employee type. Utilize the super keyword judiciously for invoking base class functionality.

Inheritance Hierarchy:

- Extend the hierarchy with a new class, ExecutiveEmployee, derived from SalariedEmployee.
- Introduce an extra attribute:
  - bonusPercentage (double): the percentage of the annual salary allocated as a bonus.
- Override the calculateBonus() method in the ExecutiveEmployee class, utilizing the super keyword to invoke the overridden method from the base class.

Data Validation:

- Implement robust data validation mechanisms to ensure that hourly rates, hours worked, monthly salary, and bonus percentages conform to predefined ranges.

Additional Features:

- Extend the classes with methods to compute and display annual earnings for each employee.
- Integrate functionality for tracking and displaying the total payroll, maintaining a professional and comprehensive employee compensation overview.

Question 2:

Continuing from the previous question, your task is to implement a comprehensive test program showcasing the capabilities of the employee payroll system. Instantiate objects of HourlyEmployee, SalariedEmployee, and ExecutiveEmployee, set attributes, and display detailed information. Utilize the super keyword where appropriate, ensuring a seamless invocation of base class methods.

Test Program:

- Develop a thorough test program that instantiates objects of each employee type.
- Set attributes such as employeeId, employeeName, designation, hourlyRate, hoursWorked, monthlySalary, and bonusPercentage.
- Display detailed information for each employee, including calculated weekly salary, bonus, annual earnings, and total payroll.

Implementation Considerations:

- Ensure that the test program reflects the functionalities implemented in the system, including inheritance, method overriding, and data validation.
- Incorporate the super keyword where appropriate to invoke methods from the base class.

- Provide comments or documentation to enhance the understanding of your code.

Evaluation Scheme: (Total 10 Marks)

Execution of Code 1 & 2(4+4)	(8 marks)
Concept Clarity (Viva)	(2 marks)

General Instruction:

1. Ensure that your code includes relevant comments to enhance readability and understanding. Subsequently, upload your code to GitHub for version control and collaborative access.
2. Include descriptive comments within the code, explaining its functionality and logic.
3. In the Google Classroom submission, include the GitHub URL where your code is hosted.
4. Attach a PDF document named "your_register_number_exercise_No.pdf" to the submission.The PDF document should include screenshots of the code and the output screen.
5. Upload the answer document&GitHub URL in Google Classroom on or before the deadline mentioned.