1) Assignment on class implementation – Level 1

WeCare insurance company wants to calculate premium of vehicles. Vehicles are of two types – "Two Wheeler" and "Four Wheeler". Each vehicle is identified by vehicle id, type, cost and premium amount. Premium amount is 2% of the vehicle cost for two wheelers and 6% of the vehicle cost for four wheelers. Calculate the premium amount and display the vehicle details.

Identify the class name and attributes to represent vehicles.

- CalculatePremium()
- vehicleCost
- TwoWheeler
- vehicleType
- vehicleId
- Vehicle
- premiumAmount
- FourWheeler
- premiumPercentage
- calculateVehicleCost()
- constructor()
- displayVehicleDetails()

Write a program to implement the class chosen with its attributes and methods.

Note:

1. Consider all instance variables to be private and methods to be public
2. Include getter and setter methods for all instance variables
3. Display appropriate error message, if the vehicle type is invalid
4. Perform case sensitive string comparison

Represent few objects of the class, initialize instance variables using setter methods, invoke appropriate methods and test your program.

2) Problem Statement: Level - 1

KHP, a technology training center, wants to allocate courses for instructors.
An instructor is identified by name, technology skills, experience and average feedback.
An instructor is allocated a course, if he/she satisfies the below two conditions:

- eligibility criteria:
    - if experience is more than 3 years, average feedback should be 4.5 or more
    - if experience is 3 years or less, average feedback should be 4 or more
- he/she should posses the technology skill for the course

Identify the class name and attributes from the list of options below to represent instructors.

- checkEligibility()
- avgFeedback
- experience
- instructorName
- allocateCourse()
- allocateCourse(technolody)
- constructor()
- Instructor
- calculateAvgFeedback()
- technologySkill

Write a program to implement the class chosen with its attributes and methods.

Note:

1. Consider all instance variables to be private and methods to be public
2. An instructor may have multiple technology skills, so consider instance variable, technology_skill to be a list
3. **checkEligibility():** Return true if eligibility criteria is satisfied by the instructor. Else, return false
4. **allocateCourse(technology):** Return true if the course which requires the given technology can be allocated to the instructor. Else, return false
5. Perform case sensitive string comparison

Represent few objects of the class, initialize instance variables using setter methods, invoke appropriate methods and test your program.

3) Problem Statement: (Level -2)

A university wants to automate their admission process. Students are admitted based on marks scored in a qualifying exam.
A student is identified by student id, age and marks in qualifying exam.
Data are valid, if:

- Age is greater than 20
- Marks is between 0 and 100 (both inclusive)

A student qualifies for admission, if

- Age and marks are valid and
- Marks is 65 or more

Write a program to represent the students seeking admission in the university.

The details of student class are given below.

Class name: Student

Attributes (private)	studentId marks age	
Methods (public)	Constructor()	Create and initialize all instance variables to None
	validateMarks()	If data is valid, return true. Else, return false
	validateAge()	
	checkQualification()	Validate marks and age. • If valid, check if marks is 65 or more. ○ If so return true ○ Else return false • Else return false
	setter methods	Include setter methods for all instance variables to set its values
	getter methods	Include getter methods for all instance variables to get its values

4) Problem statement (Continuation to problem statement 3)

Continuing with the previous scenario, a student eligible for admission has to choose a course and pay the fees for it. If they have scored more than 85 marks in qualifying exam, they get 25% discount on fees.

Valid course ids and fees are given below:

course id	**fees**
1001	25575.0
1002	15500.0

Extend the program written in the previous assignment to include the above requirement.

Instance variables and methods to be included in Student class are given below.

Class name: Student

Attributes (private)	courseId fees	
Methods (public)	Constructor()	Create and initialize newly created instance variables also to None
	chooseCourse(courseId)	Accept the courseId chosen by the student. • If courseId is valid, ○ set attributes courseId and fees ○ if marks is more than 85, apply 25% discount on fees ○ return true • Else, return false
	getter methods	Include getter methods for newly added instance variables