Assignment 6

Subject: PPWJ (CSE 3838)
Session: Sep 2025 to Jan 2026
Branch: CSIT
Section: All
Course Outcomes: CO3

Learning Levels: Remembering (L1), Understanding (L2), Application (L3), and Analysis (L4)

Q No.	Questions	Learning Levels
Q1	You are required to build a Student Management System in JavaScript using Object-Oriented Programming concepts.Create a class named Student with the following properties inside the constructor(name,rollNumber,marks) Implement getter and setter methods for:name and marks Ensure that the marks setter only accepts values between 0 and 100.Add the following methods inside the class: • displayInfo() → should return the student's name, roll number, and marks • result() → should return "Pass" if marks 40, otherwise "Fail" • Create at least five Student objects and store them in an array. • Use a for...of loop to display each student's details and pass/fail status in the console. Update at least two students' information (one name and one marks) using the setter methods, and again use a for...of loop to display the updated list.	L3,L4
Q2	Create a class named **Person** which includes a constructor for the properties: firstname,lastname.Create a variable and assign it to a new **Person** object using your friend's first name and last name.Add a second variable for another friend, also using their first and last names.Output both friends into the console with a greeting message: `"Hello, Firstname Lastname"`.	L3,L4

Q3	Using the Person class from Q2, add a method called `fullname`, which returns the concatenated value of firstname and lastname when invoked. Create values for person1 and person2 using two friends' first and last names. Using the `fullname` method within the class, return the full name of one or both people.	L3, L4
Q4	Create a class named `BankAccount` with the following properties: • `accountNumber` • `holderName` • `balance` Use a **getter** method to access the current balance of the account. Use a **setter** method to update the balance (through deposit or withdrawal). Ensure that: • Deposit increases the balance. • Withdrawal decreases the balance only if sufficient funds are available. Create at least two `BankAccount` objects with different holders and perform deposit and withdrawal operations. Display the final balance of each account using the getter method.	L3,L4

Q5	Design and implement a program in JavaScript to demonstrate the concept of **inheritance**. Follow the steps below:
	
	1. Create a base class `Vehicle` with:
	• Fields: `color`, `currentSpeed`, `maxSpeed`.
	• Methods:
	– `move()` – prints the message `"moving at <currentSpeed>"`.
	– `accelerate(amount)` – increases the `currentSpeed` by the given amount.
	
	2. Create a subclass `Motorcycle` that extends `Vehicle` and includes:
	• An additional field `fuel`.
	• A new method `doWheelie()` that prints `"Driving on one wheel!"`.
	
	3. Instantiate an object of `Motorcycle` with appropriate values and perform the following:
	• Print the `color` of the motorcycle (inherited property).
	• Accelerate the motorcycle by 50 units using the `accelerate()` method.
	• Call the `move()` method to display the updated speed.
	• Call the subclass-specific method `doWheelie()`.
	
	4. Display the program output to verify inheritance and subclass extension.
	
	Expected Output Example:
	
	`Black`
	`moving at 50`
	`Driving on one wheel!`
		L3,L4

	-END-	