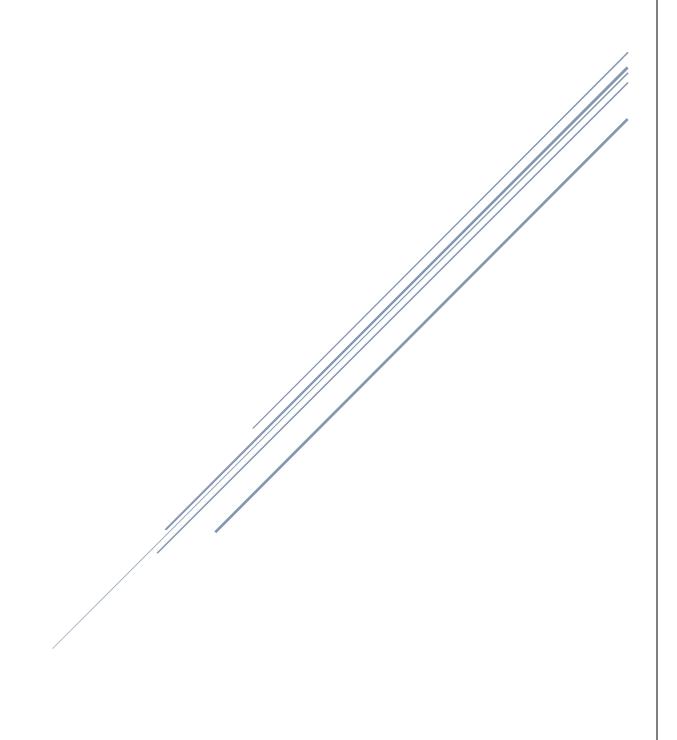
OBJECT ORIENTED PROGRAMMING AND DESIGN

Marking Scheme for Problem Sheet - 01



1. Student Attendance Management System

Criteria	Marks	Details
Student Class Implementation	30	- Private fields with correct data types (10)
		- Constructor to initialize all fields (10)
		- Getters and setters with appropriate access modifiers (10)
Classroom Class Implementation	35	- Array of Student objects initialized correctly (10)
		- Count variable to track number of students (5)
		- Method to add student with boundary checks (5)
		- Method to update attendance by student ID (10)
		- Method to display all student details (5)
Main Method / Demonstration	20	- Adding given students correctly (5)
		- Updating attendance correctly (5)
		- Handling update for non-existent student ID gracefully (5)
		- Displaying all student details (5)
Code Quality and Style	15	- Naming conventions (5)
		- Proper encapsulation and access modifiers (5)
		- Readable formatting and meaningful comments (5)

2. Parking Lot Vehicle Tracker

Criteria	Marks	Details
Vehicle Class Implementation	30	- Private fields with correct types (10)
		- Constructor initializing all fields (10)
		- Getters and setters with proper access modifiers (10)
ParkingLot Class Implementation	40	- Array to hold Vehicle objects with fixed size 5 (10)
		- Count variable tracking parked vehicles (5)
		- Method to add (park) a vehicle with capacity check (10)
		- Method to remove a vehicle by license plate with shifting (10)
		- Method to display all parked vehicles (5)
Main Method / Demonstration	20	- Park 3 given vehicles correctly (5)
		- Remove vehicle by license plate correctly (5)
		- Display parked vehicles correctly (5)
		- Handles non-existent removal gracefully (5)
Code Quality and Style	10	- Naming conventions, access modifiers, and formatting

3. Simple Bank Account Manager with Error Handling

Criteria	Marks Details		
BankAccount Class	30	- Private fields with correct types (10)	
		- Constructor and getters/setters (10)	
		- withdraw method with exception throwing (10)	
Bank Class	35	- Array of BankAccount objects with fixed size (10)	
		- Count to track number of accounts (5)	
		- Method to add accounts (5)	
		 Withdraw method with try-catch for IllegalArgumentException (10) 	
		- Method to display all accounts (5)	
Main Method / Demo	o 20	- Adding given accounts (5)	
		- Withdrawal calls including exception scenario (10)	
		- Displaying all accounts (5)	
Code Quality and Style	15	- Naming conventions, access control, formatting, comments	

4. Movie Ticket Booking System

Criteria	Marks Details	
Class Design	30	- Ticket class with private fields and constructor (15)
		- BookingSystem class with array for tickets (10)
		- Proper use of access modifiers and naming conventions (5)
Booking Functionality	25	- Book ticket method with seat availability check (15)
		- Prevent double booking of seat (10)
Cancellation Functionality	20	- Cancel ticket method by ticket number (15)
		- Handle case where ticket number not found (5)
Display Functionality	15	- Display all booked tickets correctly formatted
Main Method / Demonstration	10	- Demonstrate booking, canceling, booking again, display (10)

5. Student Grade Calculator with Exception Handling

Criteria	Marks Details		
Student Class	40	- Private fields and constructor (10)	
		- Input validation throwing IllegalArgumentException (15)	
		- Getter methods for all fields (10)	
		- calculateAverage() method returning correct average (5)	
Main Method	40	- Try-catch block to handle invalid exam score exception (20)	
		- Correct output on valid input (10)	
		- Printing exception message if invalid (10)	
Code Quality and Style	2 0	- Naming conventions, access modifiers, formatting, comments	