OOP EXERCISE

Each exercise includes:

- **Learning Objectives**
- **Class Hierarchy**
- **Attributes & Validations**
- **Task Requirements**
- **Expected Output**

Display all input data and validate all input as mentioned in the table

---

QUESTION 1 — Hospital Management System

Learning Objectives

- Demonstrate inheritance across 10 related classes.
- Apply encapsulation using private fields and accessor methods.
- Use constructors for initialization with validations.
- Accept user input using the `Scanner` class.
- Implement `generateBill()` in the final class.

Class Hierarchy

Entity → Hospital → Department → Doctor → Nurse → Patient → Admission → Treatment → Bill → HospitalRecord

Attributes & Validation

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0; dates not null/empty
Hospital	hospitalName, address, phoneNumber, email	phone = 10 digits; valid email format
Department	departmentName, departmentCode	code alphanumeric; ≥ 3 chars
Doctor	doctorName, specialization, doctorEmail, phone	specialization not empty; valid email & phone
Nurse	nurseName, shift, yearsOfExperience	shift = “Day/Night”; years ≥ 0
Patient	patientName, age, gender, contactNumber	age > 0; gender = “Male/Female/Other”

Class	Attributes	Validation
Admission	admissionDate, roomNumber, roomCharges	roomCharges > 0; admissionDate not null
Treatment	diagnosis, treatmentGiven, treatmentCost	treatmentCost > 0; fields not empty
Bill	doctorFee, medicineCost, totalBill	all > 0
HospitalRecord (final)	—	Method: `generateBill()` = roomCharges + treatmentCost + doctorFee + medicineCost

Task Requirements

- Implement all 10 classes with inheritance.
- Include private fields and public getters/setters.
- Add validation in constructors; throw `HospitalDataException` for invalid input.
- Create main method for data entry using `Scanner`.
- Display final computed bill.

Expected Output

Displays all patient and hospital details, total charges, and final computed bill.

QUESTION 2 — School Management System

Class Hierarchy

Entity → School → Department → Teacher → Student → Course → Exam → Result → Fee → StudentRecord

Attributes & Validation

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0; dates not null/empty
School	schoolName, address, phoneNumber, email	valid phone & email
Department	departmentName, departmentCode	code ≥ 3 chars; alphanumeric
Teacher	teacherName, subject, teacherEmail, phone	subject not empty; valid email & phone

Class	Attributes	Validation
Student	studentName, rollNumber, grade, contactNumber	rollNumber > 0; grade not empty
Course	courseName, courseCode, creditHours	creditHours > 0
Exam	examName, maxMarks, examDate	maxMarks > 0; date not empty
Result	obtainedMarks, remarks	obtainedMarks ≥ 0; remarks not empty
Fee	tuitionFee, examFee, totalFee	all > 0
StudentRecord (final)	averageMarks	`calculateAverageMarks()` = (obtainedMarks / maxMarks) × 100

QUESTION 3 — Employee Payroll System

Class Hierarchy

Entity → Company → Department → Manager → Employee → Attendance → Allowance → Deduction → Salary → PayrollRecord

Attributes & Validation

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0; not null
Company	companyName, address, phoneNumber, email	phone = 10 digits; valid email
Department	departmentName, departmentCode	alphanumeric; ≥ 3 chars
Manager	managerName, managerEmail, phone	not empty; valid email & phone
Employee	employeeName, employeeId, designation, contactNumber	id > 0; contact = 10 digits
Attendance	totalDays, presentDays, leaveDays	all ≥ 0; presentDays ≤ totalDays
Allowance	housingAllowance, transportAllowance	all ≥ 0
Deduction	taxDeduction, loanDeduction	all ≥ 0
Salary	basicSalary, grossSalary, netSalary	> 0
PayrollRecord (final)	—	`calculateNetSalary()` = basicSalary + allowances − deductions

---

Hotel Reservation System

Class Hierarchy

Entity → Hotel → Room → Customer → Booking → Service → Payment → Bill → Feedback → ReservationRecord

Attributes & Validation

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0; not null
Hotel	hotelName, address, phoneNumber, email	valid formats
Room	roomNumber, roomType, pricePerNight	price > 0
Customer	customerName, customerEmail, contactNumber	valid email/phone
Booking	bookingDate, checkInDate, checkOutDate	not empty
Service	serviceName, serviceCost	serviceCost > 0
Payment	paymentMethod, paymentDate	not null
Bill	roomCharge, serviceCharge, totalBill	all > 0
Feedback	rating, comments	rating between 1–5
ReservationRecord (final)	—	`generateBill()` = roomCharge + serviceCharge

QUESTION 5 — Vehicle Rental System

Class Hierarchy

Entity → Company → Branch → Vehicle → Customer → Rental → Charge → Payment → Invoice → RentalRecord

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0
Company	companyName, address, phoneNumber	phone = 10 digits
Branch	branchName, locationCode	code ≥ 3 chars
Vehicle	vehicleType, registrationNumber, dailyRate	dailyRate > 0
Customer	customerName, licenseNumber, contactNumber	license not empty

Class	Attributes	Validation
Rental	rentalDate, returnDate, rentalDays	rentalDays > 0
Charge	rentalCharge, penaltyCharge	≥ 0
Payment	paymentMode, transactionId	not empty
Invoice	totalCharge	> 0
RentalRecord (final)	—	`calculateTotalCharge()` = rentalCharge + penaltyCharge

QUESTION 6 — Banking System

Class Hierarchy

Entity → Bank → Account → Customer → Transaction → Deposit → Withdrawal → Loan → Payment → AccountRecord

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0
Bank	bankName, branchCode, address	branchCode ≥ 3 chars
Account	accountNumber, accountType, balance	balance ≥ 0
Customer	customerName, email, phoneNumber	valid formats
Transaction	transactionId, transactionType, amount	amount > 0
Deposit	depositAmount, depositDate	depositAmount > 0
Withdrawal	withdrawalAmount, withdrawalDate	withdrawalAmount > 0
Loan	loanAmount, interestRate, duration	all > 0
Payment	paymentAmount, paymentDate	amount > 0
AccountRecord (final)	—	`calculateInterest()` = (loanAmount × interestRate × duration)/100

QUESTION 7 — Real Estate Management System

Class Hierarchy

Entity → Agency → Agent → Property → Seller → Buyer → Agreement → Payment → Commission → RealEstateRecord

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0
Agency	agencyName, location, phoneNumber	valid phone
Agent	agentName, email, licenseNumber	valid email
Property	propertyCode, propertyType, price	price > 0
Seller	sellerName, contactNumber	not empty
Buyer	buyerName, email	valid email
Agreement	agreementDate, terms	not empty
Payment	paymentAmount, paymentDate	paymentAmount > 0
Commission	commissionRate, commissionAmount	rate ≥ 0
RealEstateRecord (final)	—	`calculateCommission()` = (price × commissionRate)/100

QUESTION 8 — Library Management System

Entity → Library → Section → Book → Member → Borrow → Fine → Payment → Record → LibraryRecord

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0
Library	libraryName, location, phoneNumber	valid phone
Section	sectionName, sectionCode	≥ 3 chars
Book	title, author, ISBN	ISBN ≥ 10 chars
Member	memberName, memberId, contactNumber	memberId > 0
Borrow	borrowDate, returnDate	not empty
Fine	fineAmount, daysLate	fineAmount ≥ 0
Payment	paymentDate, paymentMode	not empty
Record	totalFine	> 0
LibraryRecord (final)	—	`calculateFine()` = fineAmount × daysLate

QUESTION 9 — Airline Ticketing System

Entity → Airline → Flight → Passenger → Seat → Ticket → Baggage → Payment → Invoice → TicketRecord

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0
Airline	airlineName, address, contactEmail	valid email
Flight	flightNumber, destination, departureTime	not empty
Passenger	passengerName, passportNumber, nationality	passport not empty
Seat	seatNumber, seatType	seatType = “Economy/Business”
Ticket	ticketNumber, price	price > 0
Baggage	baggageWeight, baggageFee	≥ 0
Payment	paymentDate, paymentMode	not empty
Invoice	totalFare	> 0
TicketRecord (final)	—	`generateInvoice()` = price + baggageFee

QUESTION 10 — Online Shopping System

Entity → Store → Category → Product → Customer → Order → Payment → Shipping → Invoice → OrderRecord

Class	Attributes	Validation
Entity	id, createdDate, updatedDate	id > 0
Store	storeName, address, email	valid email
Category	categoryName, categoryCode	≥ 3 chars
Product	productName, productCode, price	price > 0
Customer	customerName, contactNumber, address	not empty
Order	orderDate, orderId	not empty
Payment	paymentMethod, paymentStatus	not empty
Shipping	shippingAddress, shippingCost	shippingCost ≥ 0
Invoice	totalAmount	> 0

Class	Attributes	Validation
OrderRecord (final)	—	`calculateTotalAmount()` = price + shippingCost