Extended Scenario-Based Java Problems (With Main Class)

---

1. University Student Portal

- Create a class `Student` with attributes: `rollNo`, `name`, `marks[]`, and a `final String universityName = "ABC University"`.
- Add a constructor to initialize student details and an initialization block to print: *"Student Record Created"*.
- Add methods:
    - `calculateAverage()` → returns average marks.
    - `displayInfo()` → prints student details and average.
- In `Main` class:
    - Ask the user (via `Scanner`) to enter details of 3 students and store them in an array.
    - Display each student's info.
    - Use a static method in `Student` class `findTopper(Student[] s)` to print the topper's name.

---

2. Online Banking Application

- Create a class `BankAccount` with attributes: `accountNumber`, `balance`, and `final String bankName = "XYZ Bank"`.
- Static variable: `totalAccounts` (increment when a new account is created).

- Add constructors (default and parameterized).
- Methods:
    - `deposit(double amount)`
    - `withdraw(double amount)` (if balance is enough)
    - `displayAccountInfo()`
- In `Main` class:
    - Create 3 accounts using constructor.
    - Perform deposits and withdrawals.
    - Print total accounts created (using static variable).

---

3. Online Movie Ticket Booking

- Create a class `Movie` with attributes `title`, `theaterName`, and a **jagged array** representing seats (rows of different sizes).
- Constructor initializes movie details and seat availability.
- Methods:
    - `bookSeat(int row, int seat)` → mark as booked.
    - `showSeats()` → print seating arrangement.
- In `Main` class:
    - Create a `Movie` object for `"Avengers"`.
    - Ask user to enter row and seat number to book (using `Scanner`).
    - Update and display the seating chart.

---

4. Employee Payroll Management

- Create a class `Employee` with attributes: `id`, `name`, `basicSalary`.
- A static variable `companyName = "TechCorp"`.
- Constructor initializes employee details.
- Methods:
    - `calculateBonus()` (10% of salary, rounded using `Math.round()`).
    - `displaySalarySlip()` → prints salary + bonus.
- In `Main` class:
    - Input details for 5 employees.
    - Print salary slip for each.
    - Use a static method `calculateTotalPayroll(Employee[] e)` to print total payroll.

---

5. Library Management System

- Create a `Book` class with attributes: `id`, `title`, `author`, `isIssued`.
- Initialization block: prints *"New Book Added to Library"*.
- Methods: `issueBook()`, `returnBook()`, `displayBook()`.
- In `Main` class:
    - Store 5 books in an array.
    - Ask the user which book to issue/return.

- Display all book statuses.

---

6. Hotel Room Reservation

- Represent hotel rooms using a **2D array** (floors × rooms).
- Create a class `Hotel` with:
  - Constructor to initialize rooms (`0 = free`, `1 = booked`).
  - Methods: `bookRoom(int floor, int room)`, `checkAvailability()`, `showRooms()`.
- In `Main` class:
  - Create a `Hotel` object with 3 floors and 5 rooms each.
  - Book rooms via user input (`Scanner`).
  - Display updated room chart.

---

7. Weather Report System

- Create a class `WeatherReport` with attributes: `day`, `temperatures[]` (morning & evening).
- Constructor initializes details.
- Methods:
  - `calculateAverage()`
  - `findMaxTemperature()` (using `Math.max()`)
- In `Main` class:

- Create reports for 7 days.
- Store in array.
- Print weekly average and hottest day.

---

8. Online Quiz Application

- Create a class `Quiz` with:
  - A final variable `TOTAL_QUESTIONS = 5`.
  - Static variable `score`.
  - Method `checkAnswer(String userAnswer, String correctAnswer)` to update score.
- In `Main` class:
  - Store 5 questions and answers in arrays.
  - Ask the user each question.
  - Call `checkAnswer()` to update score.
  - At the end, print final score.

---

9. Parking Lot Management

- Create a class `ParkingLot` with attributes: `lotNumber`, `isOccupied`.
- Static variable: `totalCars`.
- Method `parkCar(int lotNumber)`, `removeCar(int lotNumber)`.
- In `Main` class:

- Create an array of 10 parking lots.
- Randomly assign cars using `Math.random()`.
- Allow user to park/remove cars.
- Display available slots and total cars parked.

---

10. Student Hostel System

- Create a class `Hostel` with:
  - A jagged array for floors and rooms.
  - Static block: *"Hostel Management System Ready"*.
  - Methods: `allocateRoom(int floor, int room)`, `vacateRoom(int floor, int room)`, `displayRooms()`.
- In `Main` class:
  - Initialize a hostel with 3 floors (each having different no. of rooms).
  - Allocate/vacate rooms via user input.
  - Display updated hostel chart.