Java Class Design Problem Statements

1. **Library Book Management System**

Class Name: `Book`
Attributes:

- `int bookId`: Unique ID for the book
- `String title`: Title of the book
- `String author`: Author of the book
- `String genre`: Genre of the book (e.g., Fiction, Non-Fiction, Science, etc.)
- `int publicationYear`: Year of publication
- `boolean isAvailable`: Availability status of the book
- `String shelfLocation`: Location of the book within the library
- **Methods**:
  - `void setBookDetails(...)`: Set details of the book
  - `void getBookDetails()`: Display book details

Problem Statement: Create a `Book` class to help the library manage its collection. The library should be able to add, remove, and display book details.

---

2. **Bank Account Management System**

Class Name: `BankAccount`
Attributes:

- `int accountNumber`: Unique bank account number
- `String accountHolderName`: Name of the account holder
- `double balance`: Current balance in the account
- `String accountType`: Type of account (e.g., Savings, Checking)
- `String branch`: Branch where the account is held
- `boolean isActive`: Status of the account (active or inactive)
- `String currencyType`: Currency type for the account (e.g., USD, EUR)
- **Methods**:
  - `void deposit(double amount)`: Deposit money
  - `void withdraw(double amount)`: Withdraw money
  - `void displayAccountDetails()`: Display account details

Problem Statement: Create a `BankAccount` class to manage customer accounts, allowing deposits, withdrawals, and displaying account information.

---

3. **Hotel Room Booking System**

Class Name: `Room`
Attributes:

- `int roomId`: Room ID
- `String roomType`: Type of room (e.g., Single, Double, Suite)
- `double pricePerNight`: Price per night for the room
- `boolean isBooked`: Booking status of the room
- `int capacity`: Number of people the room can accommodate
- `String amenities`: List of amenities (e.g., Wi-Fi, AC, TV)
- `String floor`: Floor number
- **Methods**:
  - `void bookRoom()`: Book the room
  - `void cancelBooking()`: Cancel the booking
  - `void displayRoomInfo()`: Display room information

Problem Statement: Design a `Room` class to manage hotel rooms. The hotel should be able to book, cancel, and display room information for guests.

---

4. **Employee Payroll System**

Class Name: `Employee`
Attributes:

- `int employeeId`: Employee's unique ID
- `String name`: Name of the employee
- `String department`: Department of the employee
- `double salary`: Monthly salary
- `String jobTitle`: Job title of the employee
- `int experience`: Years of experience
- `boolean isPermanent`: Employment type (permanent or temporary)
- **Methods**:
  - `void calculateAnnualSalary()`: Calculate annual salary
  - `void promote(String newTitle)`: Update the job title
  - `void displayEmployeeInfo()`: Display employee information

Problem Statement: Create an `Employee` class to manage employees' payroll and job details. The system should track promotions and calculate annual salaries.

---

5. **University Course Management System**

Class Name: `Course`
Attributes:

- `String courseCode`: Unique code for the course
- `String courseName`: Name of the course
- `String instructorName`: Name of the instructor
- `int durationWeeks`: Duration of the course in weeks
- `int credits`: Credits awarded for the course
- `String[] prerequisites`: Prerequisite courses, if any
- `boolean isElective`: Whether the course is an elective or not

- **Methods**:
  - `void enrollStudent()`: Enroll a student in the course
  - `void dropStudent()`: Drop a student from the course
  - `void displayCourseInfo()`: Display course information

Problem Statement: Create a `Course` class to handle course enrollment, track prerequisites, and provide course information to students.

---

6. **E-commerce Product Management**

Class Name: `Product`
Attributes:

- `int productId`: Product ID
- `String name`: Name of the product
- `String category`: Category of the product (e.g., Electronics, Clothing)
- `double price`: Price of the product
- `int stock`: Number of units in stock
- `double rating`: Average customer rating
- `String supplier`: Supplier of the product
- **Methods**:
  - `void updateStock(int newStock)`: Update stock quantity
  - `void applyDiscount(double percentage)`: Apply discount on the price
  - `void displayProductDetails()`: Display product details

Problem Statement: Design a `Product` class for an e-commerce platform to manage product details, stock levels, and pricing information.

---

7. **Online Student Record Management**

Class Name: `Student`
Attributes:

- `int studentId`: Unique student ID
- `String name`: Name of the student
- `String major`: Major/field of study
- `double gpa`: Grade Point Average
- `int year`: Current year of study
- `String[] courses`: List of enrolled courses
- `boolean isInternational`: Whether the student is an international student
- **Methods**:
  - `void addCourse(String course)`: Enroll in a course
  - `void dropCourse(String course)`: Drop a course
  - `void displayStudentInfo()`: Display student information

Problem Statement: Create a `Student` class to handle student records, including their enrolled courses, GPA, and year of study.

---

8. **Real Estate Property Management**

Class Name: `Property`
Attributes:

- `int propertyId`: Property ID
- `String address`: Address of the property
- `double price`: Sale or rental price
- `String propertyType`: Type of property (e.g., Apartment, Villa)
- `int areaSqFt`: Area in square feet
- `boolean isAvailable`: Availability status
- `String agent`: Agent responsible for the property
- **Methods**:
  - `void markSold()`: Mark property as sold
  - `void updatePrice(double newPrice)`: Update the property price
  - `void displayPropertyDetails()`: Display property details

Problem Statement: Design a `Property` class to manage property listings, including availability, pricing, and agent details.

---

9. **Car Rental System**

Class Name: `Car`
Attributes:

- `int carId`: Unique car ID
- `String make`: Make of the car (e.g., Toyota)
- `String model`: Model of the car
- `int year`: Year of manufacture
- `String fuelType`: Type of fuel used (e.g., Petrol, Diesel, Electric)
- `double rentalPricePerDay`: Daily rental price
- `boolean isRented`: Rental status of the car
- **Methods**:
  - `void rentCar()`: Mark the car as rented
  - `void returnCar()`: Mark the car as returned
  - `void displayCarInfo()`: Display car information

Problem Statement: Create a `Car` class to manage car rentals, including rental pricing, availability, and car details.

---

10. **Restaurant Menu Management**

Class Name: `MenuItem`
Attributes:

- `int itemId`: Unique item ID
- `String name`: Name of the menu item

- `String category`: Category (e.g., Appetizer, Main Course, Dessert)
- `double price`: Price of the item
- `boolean isAvailable`: Availability status
- `int calories`: Caloric content
- `String description`: Description of the item
- **Methods**:
  - `void markUnavailable()`: Mark item as unavailable
  - `void updatePrice(double newPrice)`: Update the price
  - `void displayMenuItem()`: Display menu item details

Problem Statement: Design a `MenuItem` class to manage restaurant menu items, track availability, and update pricing information.
