Java Class and Object Exercise

1. **Car Class**

Create a Car class with three attributes: brand, model, and year.

- Add a method displayInfo() that prints all details.

Create two Car objects in the main method and call displayInfo() for each.

2. **Rectangle**

- Create a Rectangle class with attributes: length, width.
  - Method calculateArea() to return area.
  - Method calculatePerimeter() to return perimeter.
- In main, create a Rectangle object and display its area and perimeter.

3. **Bank Account**

- Create a BankAccount class with:
  - Attributes: accountNumber, balance.
  - Methods: deposit(), withdraw(), and displayBalance().
- In main, create two BankAccount objects and transfer money between them.

4. **Book Information**

- Create a Book class with fields: title, author, price.
- In the main method, create three Book objects and assign values directly.
- Display all book details using a method showBook().

5. **Employee Salary**

- Create an Employee class with fields: name, designation, and salary.
- Assign values directly in main.
- Write a method increaseSalary() to add a given amount to salary.
- Display updated salary.

6. **Product Discount Calculator**

- Create a Product class with fields: name, price, discountPercentage.

- Assign values directly in main.
- Write a method calculateDiscountedPrice() to print the price after discount.

7. **Circle Area & Circumference**

- Create a Circle class with a field: radius.
- Assign value directly.
- Write methods to calculate and display the **area** and **circumference**.

8. **Shopping Cart Item**

- Create a CartItem class with fields: itemName, unitPrice, quantity.
- Assign values directly in main.
- Write a method to calculate and display the total cost.

9. **Library Member**

- Create a LibraryMember class with fields: memberName, membershipType, booksBorrowed.
- Assign values directly in main.
- Write methods borrowBook() and returnBook() to update booksBorrowed.
- Display updated status.

10. **Bank Transaction Tracker**

- Create a BankAccount class with fields: accountHolder, balance, and transactionHistory (as an array of Strings).
- Add methods:
    - deposit(double amount) → updates balance and stores "Deposited X" in history.
    - withdraw(double amount) → updates balance if enough funds, else print error.
    - showHistory() → prints all transaction history.
-

11. **Grade Management System**

- Create a Student class with fields: name, rollNumber, and an array marks[] for 5 subjects.

- Add methods:
    - calculateAverage() → returns average marks.
    - calculateGrade() → returns grade (A/B/C/D/F) based on average.
- In main, assign marks directly and display name, roll number, average, and grade.

12. **Online Shopping Cart**

- Create a Product class with fields: name, price, quantity.
- Add methods:
    - totalPrice() → returns price * quantity.
- In main, create an array of 3 products, calculate each product's total price, and sum all for the cart total.

13. **Hotel Room Booking**

- Create a Room class with fields: roomNumber, isBooked, pricePerNight, nightsBooked.
- Add methods:
    - bookRoom(int nights) → set booked status and nights.
    - calculateBill() → returns pricePerNight * nightsBooked.
- In main, create multiple rooms, book them, and display bills.

14. **Library Book Issue System**

- Create a Book class with fields: title, author, isIssued (boolean).
- Add methods:
    - issueBook() → sets isIssued = true if not already issued.
    - returnBook() → sets isIssued = false.
- In main, create a book, issue it, try issuing again, return it, then issue again.

15. **Employee Attendance**

- Create an Employee class with fields: name, id, daysPresent, totalWorkingDays.
- Add methods:
    - markAttendance(int days) → increases daysPresent.
    - attendancePercentage() → returns (daysPresent / totalWorkingDays) * 100.

- In main, create two employees, update attendance for a month, and display percentage.