Q1: Implement a Book class with five attributes: title, author, numberOfPages, price, and stockQuantity. The class should have a default constructor, a parameterized constructor, setter-getter methods and show details method. Finally, add two custom methods: getTotalValueOfStock(): Calculates and returns the total value of the current stock by multiplying the price of the book by the stockQuantity, and isLongBook(), which returns true if the book has more than 300 pages and false otherwise. You should complete the Book class and after that write a Start class that will have the main method and create two Sample objects of the Book class in the main method and demonstrate the use of different methods.

Q2: Implement a Student class that has six attributes: name, studentID, major, cgpa, creditCompleted and email. The class should have a default constructor, as well as a parameterized constructor, setter-getter methods and show details method. Additionally, add a custom method: isEligibleForDeanList(), which returns true if the student has a CGPA of 3.7 or higher and has completed at least 30 credits, indicating eligibility for the Dean's List. You should complete the Student class and after that write a Start class that will have the main method and create two Sample objects of the Student class in the main method and demonstrate the use of different methods.

Q3: Implement a Movie class that has six attributes: title, director, duration, and rating. The class should have a default constructor, as well as a parameterized constructor, setter-getter methods and show details method. Add a custom method as well: getRatingCategory(): Returns a string indicating the rating category based on the rating attribute (e.g., "Excellent" for ratings 9 and above, "Good" for 7-8.9, "Average" for 5-6.9, and "Poor" for below 5). You should complete the Movie class and after that write a Start class that will have the main method and create two Sample objects of the Movie class in the main method and demonstrate the use of different methods.

Q4: Implement a Product class that has seven attributes: name, productID, price, stockQuantity, category, supplierName, and expirationDate. The class should have a default constructor, as well as a parameterized constructor, setter-getter methods and show details method. Add a custom method as well: applyDiscount(double discountPercentage): Applies a discount to the product's price based on the given percentage. This method reduces the price by the specified percentage and return the discounted price. You should complete the Product class and after that write a Start class that will have the main method and create two Sample objects of the Product class in the main method and demonstrate the use of different methods.