1. Basic Class and Object Creation

Question: Create a Student class with attributes id, name, and grade. Implement a constructor to initialize these attributes and provide methods to get the values of each attribute. Create a few Student objects in the main method and print their details.

Objective: Understand how to create classes, define attributes, write constructors, and access methods.

2. Class Interaction

Question: Create a Book class with attributes title, author, and price. Then, create a Library class that contains an array of Book objects. Implement methods in the Library class to add a book, get the total price of all books, and list all book titles in the library.

Objective: Learn how classes can interact with each other and how to manage arrays of objects within another class.

3. Method Implementation

Question: Create a Rectangle class with attributes length and width. Implement methods to calculate the area and the perimeter of the rectangle. Write a main method to create a few rectangles and display their area and perimeter.

Objective: Practice writing methods that perform calculations based on the attributes of a class.

4. Search Functionality

Question: Create a Product class with attributes productId, productName, and price. Then, create a Store class that contains an array of Product objects. Implement a method in the Store class to find and return a Product by its productName.

Objective: Learn how to search for an object in an array based on a specific attribute value.

5. Aggregation of Data

Question: Create a Team class that contains an array of Player objects, where each Player has attributes like playerName and score. Implement a method in the Team class to calculate the total score of all players and return the highest score.

Objective: Practice aggregating data from multiple objects and performing operations like summing or finding maximum values.

6. Working with Multiple Classes

Question: Create a Car class with attributes like make, model, and price. Then, create a Garage class that can store multiple Car objects. Implement methods in the Garage class to add a car, remove a car, and find the most expensive car in the garage.

Objective: Get comfortable working with multiple classes, managing objects, and implementing common operations.

7. Sorting Objects

Question: Create a Movie class with attributes title, director, and rating. Implement a method to sort an array of Movie objects by their rating in descending order.

Objective: Learn how to sort objects based on one of their attributes.

8. Filtering Objects
Question: Create a Course class with attributes courseName, instructor, and duration. Then, create a School class that contains an array of Course objects. Implement a method in the School class to return all courses taught by a specific instructor.
Objective: Practice filtering objects based on specific criteria.
9. Simple Inventory Management
Question: Create an Item class with attributes itemId, itemName, and quantity. Then, create an Inventory class to manage a collection of Item objects. Implement methods to add an item, remove an item, and check if an item is in stock (i.e., its quantity is greater than zero).
Objective: Practice managing collections of objects and implementing basic inventory management logic.
10. Calculation Across Objects
Question: Create an Order class with attributes orderId, product, quantity, and pricePerUnit. Implement a method to calculate the total price for the order. Then, create a Customer class that has an array of Order objects and a method to calculate the total amount spent by the customer.
Objective: Practice performing calculations across multiple objects and understanding how classes can be designed to model real-world scenarios.