Some sample questions.

Q1.

Design a system to manage a collection of employees in a company.

1. Create an `Employee` class with the following fields:
    - Name (String)
    - Employee ID (String)
    - Salary (double)
2. Create a generic class `Company<T>` where `T` represents any type of employee. The class should:
    - Use a collection (`HashMap<String, T>`) to store employees where the key is the employee ID.
    - Provide methods to add, remove, and display employees.
    - Handle exceptions for cases like adding an employee with an existing ID or removing a non-existent employee.
3. Implement the following methods in `Company`:
    - `void addEmployee(T employee)` – Add an employee.
    - `void removeEmployee(String employeeId)` – Remove an employee by ID.
    - `void displayEmployees()` – Display all employees in the company.

In the `main` method:

- Create a `Company<Employee>` instance.
- Add at least 3 employees, display them, then remove one and display the updated list.

Q2.

Create a multithreaded file processing system that reads and writes student data.

1. Create a `Student` class with fields:
    - Name (String)
    - Roll Number (String)
    - Marks (int)
2. Create a `FileProcessor` class that implements `Runnable`. This class should:
    - Read student data from a file, calculate the average marks of all students, and write the result to another file.
    - Use synchronized methods to ensure only one thread processes file reading/writing at a time.

3. Use multithreading to simulate multiple file processors reading different student data files and calculating averages concurrently.

In the `main` method:

- Simulate 3 file processing threads reading from different files.
- Ensure proper file synchronization to prevent data corruption.