
🚗 1. Vehicle Rental System (OOP + Inheritance + Interface)

Concepts: Classes, Inheritance, Interfaces, Polymorphism **Task:**

- Base class: `Vehicle`
- Subclasses: `Car`, `Bike`, `Truck`
- Interface: `IRentable` with method `Rent()`
- Display rent cost per day and allow renting multiple vehicles.

🧑‍💼 2. Employee Directory with LINQ

Concepts: Collections, LINQ, Filtering, Sorting **Task:**

- Create a list of `Employee` objects with `ID`, `Name`, `Department`, `Salary`
- Filter employees by department
- Sort by salary descending
- Find average salary per department.

🧠 3. Quiz Application (OOP + Collections)

Concepts: Class design, Dictionary, Random, User input **Task:**

- Create a class `Question` with text, options, correct answer
- Store questions in a list
- Display random questions and score the user's answers.

🧵 4. Multithreading File Processor

Concepts: Threads, Lock, File Handling **Task:**

- Create threads to read 3 large files in parallel
- Use `Thread.Join()` to ensure all are read
- Count total lines across all files.

🔤 5. String Analyzer Tool (LINQ + String + Dictionary)

Concepts: String manipulation, LINQ, Dictionary **Task:**

- Accept a paragraph from user input
- Count:
  - Word frequency
  - Vowel/consonant count
  - Longest and shortest word

📚 6. Library Management System (OOP + Collections)

Concepts: Class structure, object list management **Task:**

- Classes: `Book`, `Student`
- Add, delete, and borrow books
- Track who borrowed what and due dates.

🧮 7. Calculator App (Method Overloading)

Concepts: Overloading, Static methods **Task:**

- Overload `Add()`, `Subtract()`, `Multiply()`
    - Different datatypes (int, float, double)
- Take user input and dynamically call correct method.

🧾 8. Invoice Generator (Interface + Abstraction)

Concepts: Abstract Class, Interface, Inheritance **Task:**

- Abstract class `Invoice` with `CalculateTotal()`
- Subclasses for `RetailInvoice`, `WholesaleInvoice`
- Interface `IPrintable` with `Print()`
- Print an invoice with product list, tax, and discount.

🌦️ 9. Async Weather Fetcher (async/await + Task.Delay)

Concepts: async/await, Task.WhenAll **Task:**

- Simulate calling weather APIs for 3 cities
- Use `Task.Delay` to mimic delay
- Fetch all concurrently and print total time taken.

🎓 10. Student Gradebook with LINQ + Collections

Concepts: Dictionary, List, LINQ **Task:**

- Accept marks of students in multiple subjects
- Calculate average, highest, and lowest per student
- Find top 3 scorers using LINQ.
