



CPP-A20: File Handling Text File



Problem 1: Library Management System



Problem Statement:

You are building a **Library Management System** where books are stored in a file called `"library.txt"`.

Each book has:

- **Book ID** (Unique identifier)
- **Title**
- **Author Name**
- **Availability Status** (Available / Issued)



Tasks:

1. **Add new books** to `"library.txt"`.
2. **Search for a book by Book ID** and display its details.
3. **Update a book's details** (title, author, or status).
4. **Delete a book entry** from the file.



Example Input & Output:

Input:

Add Book: [101, "C++ Programming", "Bjarne Stroustrup", "Available"]

Update Book ID: 101 (Change Status to Issued)

Delete Book ID: 101

Output:

Book Found: C++ Programming by Bjarne Stroustrup (Available)

Book Status Updated Successfully!
Book Deleted Successfully!



Problem 2: Employee Payroll System



Problem Statement:

Develop an **Employee Payroll System** that stores employee data in `"employees.txt"`.

Each employee has:

- **Employee ID**
- **Name**
- **Department**
- **Salary**



Tasks:

1. **Add employee details dynamically.**
2. **Retrieve an employee's salary** based on Employee ID.
3. **Increase an employee's salary** by a given percentage.
4. **Update employee details** (name, department, salary).
5. **Delete an employee record** from the file.



Example Input & Output:

Input:

Add Employee: [201, "Alice", "HR", 50000]

Update Employee ID: 201 (Change Salary to 55000)

Delete Employee ID: 201

Output:

Employee Found: Alice (HR) - Salary: 50000

Employee Salary Updated Successfully!

Employee Deleted Successfully!



Problem 3: Online Store Order Management



Problem Statement:

Build an **Order Management System** for an online store using `"orders.txt"`.

Each order has:

- **Order ID**
- **Customer Name**
- **Product Name**
- **Price**



Tasks:

1. **Add new order details dynamically.**
2. **Find total sales amount** by reading all orders.
3. **Find all orders placed by a specific customer.**
4. **Update an order** (change product name or price).
5. **Delete an order** from the system.



Example Input & Output:

Input:

Add Order: [5001, "John Doe", "Laptop", 75000]

Update Order ID: 5001 (Change Product to "Tablet")

Delete Order ID: 5001

Output:

Order Found: Laptop by John Doe - Price: 75000

Order Updated Successfully!

Order Deleted Successfully!



Problem 4: Student Exam Results System



Problem Statement:

Create a **Student Exam Results System** where exam scores are stored in `"results.txt"`.

Each student has:

- **Roll Number**
- **Name**
- **Marks in 3 subjects**

◆ **Tasks:**

1. **Add student exam results dynamically.**
2. **Search for a student's marks** based on Roll Number.
3. **Calculate and display the total and percentage marks.**
4. **Update student details** (name or marks).
5. **Delete a student's record** from the file.

▼ **Example Input & Output:**

Input:

Add Student: [101, "David", 78, 85, 90]

Update Student ID: 101 (Change Marks)

Delete Student ID: 101

Output:

Student Found: David - Total Marks: 253, Percentage: 84.33%

Student Marks Updated Successfully!

Student Record Deleted Successfully!



Problem 5: Flight Booking System

📌 **Problem Statement:**

Develop a **Flight Booking System** where flight information is stored in `"flights.txt"`.

Each flight has:

- **Flight Number**
- **Departure City**
- **Destination City**
- **Ticket Price**

◆ Tasks:

1. **Add new flights dynamically.**
2. **Find flights between two cities** and display the available options.
3. **Calculate total earnings from all bookings.**
4. **Update flight details** (destination or price).
5. **Delete a flight entry** from the file.

▼ Example Input & Output:

Input:

Add Flight: [AI101, "Delhi", "Mumbai", 4500]

Update Flight: AI101 (Change Price to 5000)

Delete Flight: AI101

Output:

Flight Found: AI101 - Price: 4500

Flight Price Updated Successfully!

Flight Deleted Successfully!

Happy Coding!