MODULE 5 – I/O STREAMS

PRACTICAL

1. Student Report Management System

Scenario:
A university wants to maintain student records in a text file. Each student record contains the **Student ID, Name, Course, and Marks**. The university needs a Java program that can:

- Add a new student record to the file.
- Display all student records from the file.
- Search for a student by ID and display the details.
- Update the marks of a student.

Task:

- Implement the program using **FileReader and FileWriter** for text-based operations.
- Ensure the program handles **exceptions** properly when reading/writing files.
- Use **BufferedReader** for efficient input handling.

Challenge Extension:

- Modify the program to use **ObjectOutputStream** and **ObjectInputStream** to store student objects as binary data.

2. Banking System with Transaction Logging

Scenario:
A bank needs a system to manage customer transactions and store them securely. Each transaction contains **Account Number, Transaction Type (Deposit/Withdraw), and Amount**. The bank requires a Java program that:

- Records each transaction in a text file for future reference.
- Reads and displays all past transactions from the file.
- Allows the user to search for transactions by account number.

Task:

- Implement the program using **PrintWriter** and **BufferedWriter** to store transactions in a readable format.
- Use **BufferedReader** to read transaction logs efficiently.

Challenge Extension:

- Use **DataOutputStream** and **DataInputStream** to store transaction data in a binary file instead of a text file.

3. Employee Payroll System

Scenario:
A company needs a payroll management system that can:

- Store employee details (**Employee ID, Name, Salary**) in a file.
- Retrieve employee details and calculate total salary expenditure.
- Update an employee's salary based on performance.

Task:

- Implement the system using **FileWriter and FileReader** for storing employee records.
- Use **Scanner** for user input.
- Implement exception handling to prevent file corruption.

Challenge Extension:

- Store employee details in a **binary file** using **ObjectOutputStream and ObjectInputStream** to enable efficient storage and retrieval.