Experiment - 5

Student Name: Manjot Singh **UID:** 23BCS12540
Branch: BE-CSE **Section/Group:** KRG-2B
Semester: 5th **Date of Performance:**24/9/25
Subject Name: Project Based Learning in Java
Subject Code: 23CSH-304

Aim: Create a menu-based Java application with the following options.

1.Add an Employee

2. Display All

3. Exit If option 1 is selected, the application should gather details of the employee like employee name, employee id, designation and salary and store it in a file.

If option 2 is selected, the application should display all the employee details.

If option 3 is selected the application should exit.

Objective: To combine object-oriented programming, file handling, and menu-driven console interaction.

Procedure:

1. Present a menu:
a) Add Employee
b) Display All
c) Exit

2. On choosing Add, take input for:
a) Employee Name
b) Employee ID
c) Designation
d) Salary

3. Write this data to a file.
4. On choosing Display, read and display all employee data from the file.
5. Exit on selection of option 3.

Sample Output -

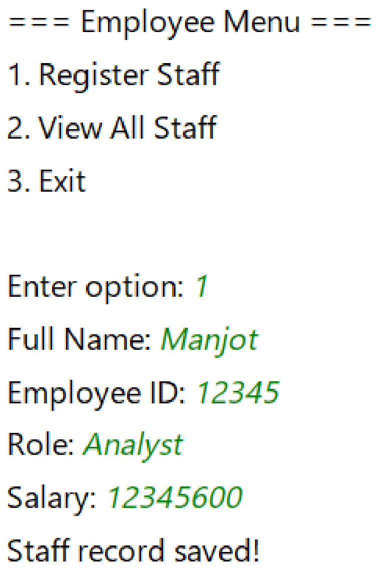
```
Menu:
1. Add Employee
2. Display All
3. Exit
Enter choice: 1
Name: Manjot
ID: 1001
Designation: Analyst
Salary: 21000
Employee added successfully!

Enter choice: 2
Employee List:
Manjot | 1001 | Analyst | 21000
```

Code –

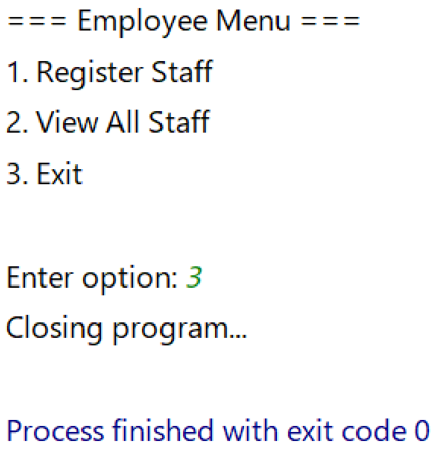
```
import java.io.*;
import java.util.*;

class Staff {
    private String fullName;
    private String empId;
    private String role;
    private double pay;

    public Staff(String fullName, String empId, String role, double pay) {
        this.fullName = fullName;
        this.empId = empId;
        this.role = role;
        this.pay = pay;
    }

    public String toFileFormat() {
        return fullName + "|" + empId + "|" + role + "|" + pay;
    }

    public static Staff fromFileFormat(String line) {
        String[] parts = line.split("\\|");
        return new Staff(parts[0], parts[1], parts[2], Double.parseDouble(parts[3]));
    }

    @Override
    public String toString() {
        return fullName + " | " + empId + " | " + role + " | " + pay;
    }
}

public class Main {

    private static final String FILE_PATH = "staff_records.txt";
    private static Scanner input = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            System.out.println("\n=== Employee Menu ===");
            System.out.println("1. Register Staff");
            System.out.println("2. View All Staff");
            System.out.println("3. Exit");
            System.out.print("\nEnter option: ");

            int option = input.nextInt();
            input.nextLine();

            switch (option) {
                case 1 -> registerStaff();
                case 2 -> viewAllStaff();
                case 3 -> {
                    System.out.println("Closing program...");
                    System.exit(0);
                }
                default -> System.out.println("Invalid option, please retry.");
            }
        }
    }
```

```java
    private static void registerStaff() {
        System.out.print("Full Name: ");
        String fullName = input.nextLine();
        System.out.print("Employee ID: ");
        String empId = input.nextLine();
        System.out.print("Role: ");
        String role = input.nextLine();
        System.out.print("Salary: ");
        double pay = input.nextDouble();
        input.nextLine();

        Staff s = new Staff(fullName, empId, role, pay);

        try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE_PATH, true))) {
            writer.write(s.toFileFormat());
            writer.newLine();
            System.out.println("Staff record saved!");
        } catch (IOException e) {
            System.out.println("Error saving to file.");
        }
    }

    private static void viewAllStaff() {
        System.out.println("\n--- Staff Records ---");
        try (BufferedReader reader = new BufferedReader(new FileReader(FILE_PATH))) {
            String line;
            boolean found = false;
            while ((line = reader.readLine()) != null) {
                Staff s = Staff.fromFileFormat(line);
                System.out.println(s);
                found = true;
            }
            if (!found) System.out.println("No records available.");
        } catch (FileNotFoundException e) {
            System.out.println("No staff records file found.");
        } catch (IOException e) {
            System.out.println("Error reading file.");
        }
    }
}
```

Output -



```
=== Employee Menu ===
1. Register Staff
2. View All Staff
3. Exit

Enter option: 1
Full Name: Manjot
Employee ID: 12345
Role: Analyst
Salary: 12345600
Staff record saved!
```



```
=== Employee Menu ===
1. Register Staff
2. View All Staff
3. Exit

Enter option: 3
Closing program...

Process finished with exit code 0
```