Experiment - 5

Student Name: Mankaran Singh Tandon **UID:** 23BCS10204
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.

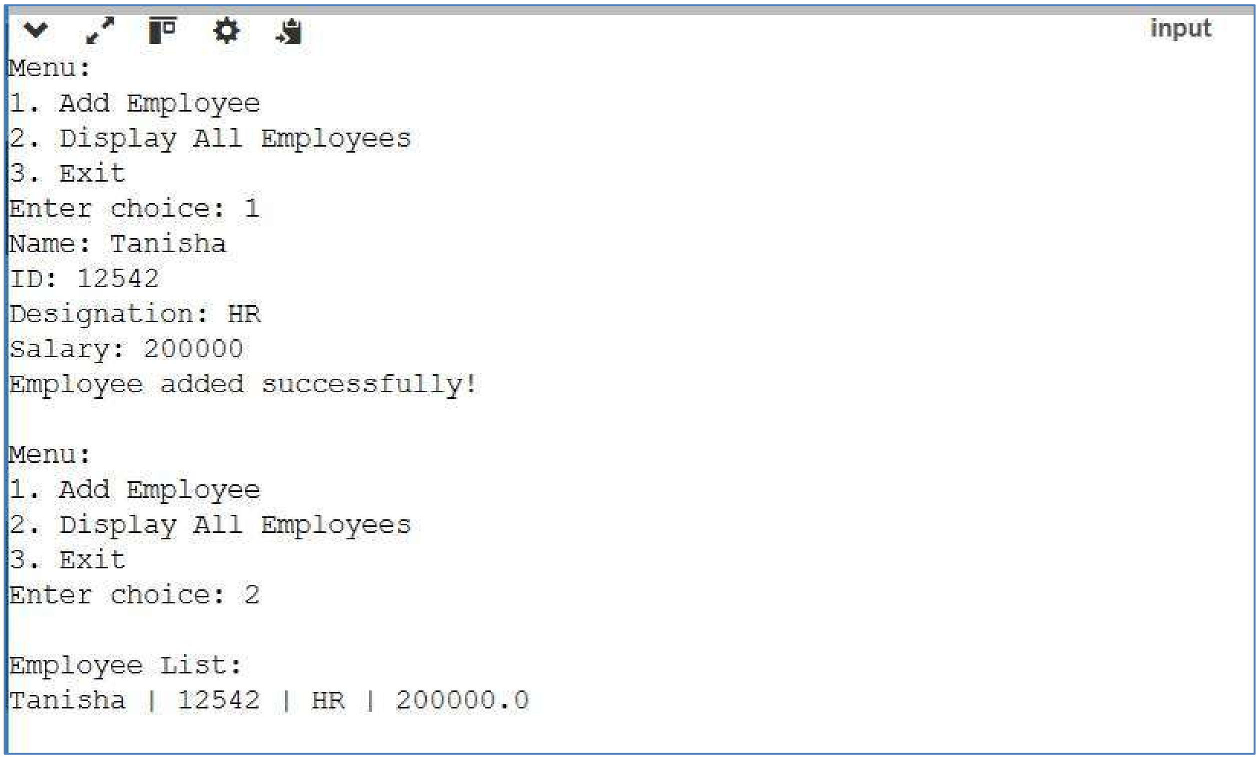
Sample Output -

```
Menu:
1. Add Employee
2. Display All
3. Exit
```

Code -

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

class  Employee  {
    private String name;
    private String id;
    private String designation;
    private double salary;

    public Employee(String name, String id, String designation, double salary) {
        this.name = name;
        this.id = id;
        this.designation = designation;
        this.salary = salary;
    }

    public String toFileString() {
        return name + "|" + id + "|" + designation + "|" + salary;
    }

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

    public String toString() {
        return name + " | " + id + " | " + designation + " | " + salary;
    }
}

public class Main {
    private static final String FILE_NAME = "employees.txt";
    private static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Add Employee");
            System.out.println("2. Display All Employees");
            System.out.println("3. Exit");
            System.out.print("Enter choice: ");
            int choice = sc.nextInt();
            sc.nextLine(); // consume newline

            switch (choice) {
                case 1:
                    addEmployee();
                    break;
                case 2:
                    displayAll();
                    break;
                case 3:
                    System.out.println("Exiting...");
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice! Try again.");
            }
        }
    }

    private static void addEmployee() {
        System.out.print("Name: ");
        String name = sc.nextLine();
        System.out.print("ID: ");
        String id = sc.nextLine();
        System.out.print("Designation: ");
        String designation = sc.nextLine();
        System.out.print("Salary: ");
        double salary = sc.nextDouble();
        sc.nextLine(); // consume newline

        Employee emp = new Employee(name, id, designation, salary);

        try (BufferedWriter bw = new BufferedWriter(new FileWriter(FILE_NAME, true)))
{
            bw.write(emp.toFileString());
            bw.newLine();
            System.out.println("Employee added successfully!");
        } catch (IOException e) {
            System.out.println("Error writing to file.");
        }
    }

    private static void displayAll() {
        System.out.println("\nEmployee List:");
        try (BufferedReader br = new BufferedReader(new FileReader(FILE_NAME))) {
            String line;
            while ((line = br.readLine()) != null) {
                Employee emp = Employee.fromFileString(line);
                System.out.println(emp);
            }
        } catch (FileNotFoundException e) {
            System.out.println("No employees found.");
        } catch (IOException e) {
            System.out.println("Error reading file.");
        }
    }
}
```

Output -



```
Menu:
1. Add Employee
2. Display All Employees
3. Exit
Enter choice: 1
Name: Tanisha
ID: 12542
Designation: HR
Salary: 200000
Employee added successfully!

Menu:
1. Add Employee
2. Display All Employees
3. Exit
Enter choice: 2

Employee List:
Tanisha | 12542 | HR | 200000.0
```