



Experiment2.4

Student Name: Nikhil

Branch: CSE

Semester: 6

Subject Name: JAVA LAB

UID: 21BCS5892

Section/Group: 620/A

Date of Performance: 28-03-24

Subject Code: 21CSH-319

1. 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.

2. Objective: To learn about concept of File Handling in java. To learn about LinkedList, Exception Handling in java.

3. Code and output:

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

public class EmployeeManagement {

    private static final String FILE_NAME = "employees.txt";

    private static ArrayList<Employee> employees = new ArrayList<>();

    public static void main(String[] args) {

        loadEmployeesFromFile();

        Scanner scanner = new Scanner(System.in);

        while (true) {

            System.out.println("Menu:");

            System.out.println("1. Add an Employee");

            System.out.println("2. Display All");
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        System.out.println("3. Exit");

        System.out.print("Select an option: ");

        int option = scanner.nextInt();

        switch (option) {

            case 1:

                addEmployee(scanner);

                break;

            case 2:

                displayAllEmployees();

                break;

            case 3:

                saveEmployeesToFile();

                System.out.println("Exiting...");

                return;

            default:

                System.out.println("Invalid option. Please try again.");

        }

    }

}

private static void loadEmployeesFromFile() {

    try (BufferedReader reader = new BufferedReader(new
    FileReader(FILE_NAME))) {

        String line;

        while ((line = reader.readLine()) != null) {

            String[] parts = line.split(",");

            if (parts.length == 4) {
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        employees.add(new Employee(parts[0], parts[1], parts[2],
Double.parseDouble(parts[3])));
    }
}
} catch (IOException | NumberFormatException e) {
}
}

private static void saveEmployeesToFile() {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE_NAME,
true))) {
        for (Employee employee : employees) {
            writer.write(employee.getName() + "," + employee.getId() + "," +
employee.getDesignation() + "," + employee.getSalary());
            writer.newLine();
        }
    } catch (IOException e) {
    }
}

private static void addEmployee(Scanner scanner) {
    System.out.print("Enter Employee Name: ");
    String name = scanner.next();
    System.out.print("Enter Employee ID: ");
    String id = scanner.next();
    System.out.print("Enter Designation: ");
    String designation = scanner.next();
    System.out.print("Enter Salary: ");
    double salary = scanner.nextDouble();
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
Employee employee = new Employee(name, id, designation, salary);
employees.add(employee);
System.out.println("Employee added successfully.");
}

private static void displayAllEmployees() {
    if (employees.isEmpty()) {
        System.out.println("No employees to display.");
    } else {
        for (Employee employee : employees) {
            System.out.println(employee);
        }
    }
}

private static 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 getName() {
        return name;
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
    }  
    public String getId() {  
        return id;  
    }  
    public String getDesignation() {  
        return designation;  
    }  
    public double getSalary() {  
        return salary;  
    }  
    @Override  
    public String toString() {  
        return "Employee{" +  
            "name=" + name + " +  
            ", id=" + id + " +  
            ", designation=" + designation + " +  
            ", salary=" + salary +  
            '}';  
    }  
}  
}
```

4. Output:

```
Menu:
1. Add an Employee
2. Display All
3. Exit
Select an option: 1
Enter Employee Name: a
Enter Employee ID: 1
Enter Designation: s
Enter Salary: 12
Employee added successfully.
```

```
Menu:
1. Add an Employee
2. Display All
3. Exit
Select an option: 2
Employee{name='Kartik', id='6833', designation='student', salary=0.0}
Employee{name='Kartik', id='6833', designation='student', salary=0.0}
Employee{name='vinay', id='833', designation='student', salary=0.0}
Employee{name='Kartik', id='6833', designation='student', salary=0.0}
Employee{name='Kartik', id='6833', designation='student', salary=0.0}
Employee{name='vinay', id='833', designation='student', salary=0.0}
Employee{name='karti', id='33', designation='student', salary=0.0}
Employee{name='a', id='1', designation='s', salary=12.0}
```

```
Menu:
1. Add an Employee
2. Display All
3. Exit
Select an option: 3
Exiting...

Process finished with exit code 0
```