**Experiment 4**

**Student Name:** Kashish **UID:** 22BCS11866

**Branch:** BE-CSE **Section/Group:** 801-A

**Semester:** 06 **Date of Performance:** 20/02/25

**Subject Name:** PBLJ **Subject Code:** 22CSH-359

1. **Aim:** Write a Program to perform the basic operations like insert, delete, display and search in list. List contains String object items where these operations are to be performed.

# Implementation/Code:

import java.util.\*;

class Employee { int id;

String name; double salary;

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

this.name = name; this.salary = salary;

}

public String toString() {

return "ID: " + id + ", Name: " + name + ", Salary: " + salary;

}

}

public class EmployeeManagement { public static void main(String[] args) {

Scanner sc = new Scanner(System.in); List<Employee> employees = new ArrayList<>();

while (true) {

System.out.println("\n1. Add Employee | 2. Update | 3. Remove | 4.

Search | 5. Display | 6. Exit"); System.out.print("Enter your choice: "); int choice = sc.nextInt();

if (choice == 6) { System.out.println("Exiting..."); sc.close();

return;

}

System.out.print("Enter Employee ID: "); int id = sc.nextInt();

sc.nextLine(); // Consume newline

switch (choice) { case 1 -> {

System.out.print("Enter Name: "); String name = sc.nextLine(); System.out.print("Enter Salary: "); double salary = sc.nextDouble();

employees.add(new Employee(id, name, salary)); System.out.println("Employee added.");

}

case 2 -> {

Employee emp = findEmployee(employees, id); if (emp != null) {

System.out.print("Enter new Name: "); emp.name = sc.nextLine(); System.out.print("Enter new Salary: "); emp.salary = sc.nextDouble(); System.out.println("Employee updated.");

} else {

System.out.println("Employee not found.");

}

}

case 3 -> {

if (employees.removeIf(e -> e.id == id)) { System.out.println("Employee removed.");

} else {

System.out.println("Employee not found.");

}

}

case 4 -> {

Employee emp = findEmployee(employees, id); System.out.println(emp != null ? "Employee Found: " + emp :

"Employee not found.");

}

case 5 -> {

if (employees.isEmpty()) System.out.println("No employees

found.");

else employees.forEach(System.out::println);

}

default -> System.out.println("Invalid choice! Try again.");

}

}

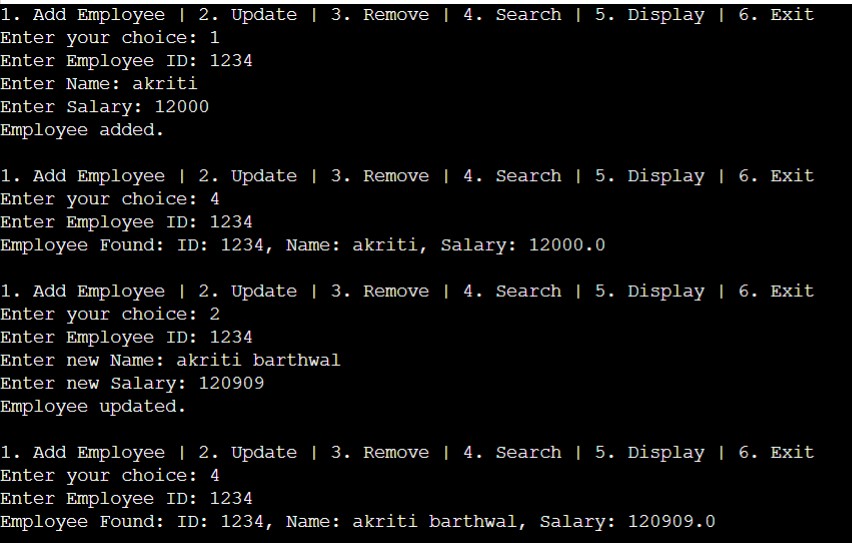
}

private static Employee findEmployee(List<Employee> employees, int id) { return employees.stream().filter(e -> e.id == id).findFirst().orElse(null);

}

}

# Output

****

1. **Learning Outcome**
   * Proficiency in Java Basics
   * Understanding Data Structures
   * Improved Problem-Solving Skills