Experiment 4.1

Student Name: Akshat Srivastava **UID:** 22BCS11740
Branch: BE CSE **Section/Group:** 22BCS_IOT_618_A
Semester: 6th **DoP:** 18/02/2025
Subject Name: PBLJ Lab **Subject Code:** 22CSH-359

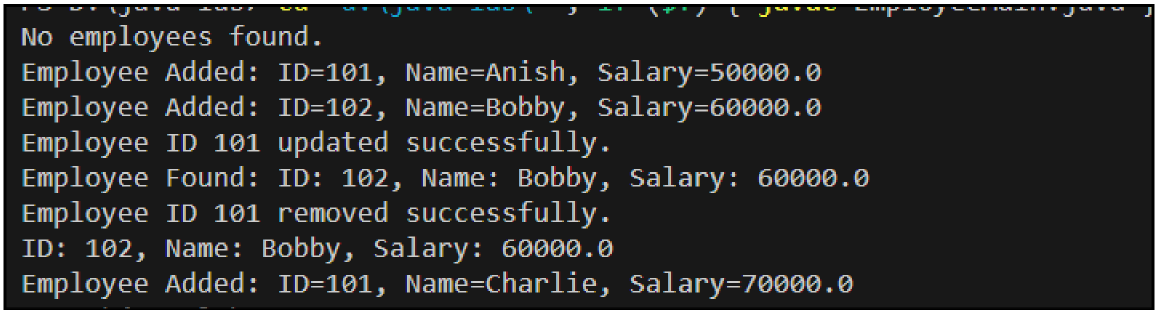
1. **Aim:** To develop a simple Java-based Employee Management System using ArrayList that allows users to add, update, remove, search, and display employee records.
2. **Objective:**
    - Implement Basic CRUD (Create,Read,Update,Delete) Operations
    - Ensure Data Integrity
    - Improve Programming Skills

3. 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;
    }
}
class EmployeeManagementSystem {
    List<Employee> employees = new ArrayList<>();

    public void addEmployee(int id, String name, double salary) {
        for (Employee emp : employees) {
            if (emp.id == id) {
                System.out.println("Error: Employee with ID " + id + " already exists.");
                return;
            }
        }
        employees.add(new Employee(id, name, salary));
        System.out.println("Employee Added: ID=" + id + ", Name=" + name + ", Salary=" + salary);
    }
    public void updateEmployee(int id, double newSalary) {
        for (Employee emp : employees) {
            if (emp.id == id) {
                emp.salary = newSalary;
                System.out.println("Employee ID " + id + " updated successfully.");
                return;
            }
        }
        System.out.println("Error: Employee ID " + id + " not found.");
    }
    public void removeEmployee(int id) {
        Iterator<Employee> iterator = employees.iterator();
        while (iterator.hasNext()) {
            Employee emp = iterator.next();
            if (emp.id == id) {
                iterator.remove();
                System.out.println("Employee ID " + id + " removed successfully.");
                return;
            }
        }
        System.out.println("Error: Employee ID " + id + " not found.");
    }
    public void searchEmployeeById(int id) {
        for (Employee emp : employees) {
            if (emp.id == id) {
                System.out.println("Employee Found: " + emp);
                return;
            }
        }
        System.out.println("Error: Employee ID " + id + " not found.");

    }

    public void displayEmployees() {

        if (employees.isEmpty()) {

            System.out.println("No employees found.");

        } else {

            for (Employee emp : employees) {

                System.out.println(emp);

            }}}}

public class Main {

    public static void main(String[] args) {

        EmployeeManagementSystem ems = new EmployeeManagementSystem();

        ems.displayEmployees();

        ems.addEmployee(101, "Anish", 50000);

        ems.addEmployee(102, "Bobby", 60000);

        ems.updateEmployee(101, 55000);

        ems.searchEmployeeById(102);

        ems.removeEmployee(101);

        ems.displayEmployees();

        ems.addEmployee(101, "Charlie", 70000);

    }}
```

4. Output



```
No employees found.
Employee Added: ID=101, Name=Anish, Salary=50000.0
Employee Added: ID=102, Name=Bobby, Salary=60000.0
Employee ID 101 updated successfully.
Employee Found: ID: 102, Name: Bobby, Salary: 60000.0
Employee ID 101 removed successfully.
ID: 102, Name: Bobby, Salary: 60000.0
Employee Added: ID=101, Name=Charlie, Salary=70000.0
```

5. Learning Outcome:

- Understanding and implementing **CRUD operations** in Java.
- Using **ArrayList** for managing dynamic employee records.
- Applying **object-oriented programming (OOP)** concepts like classes and objects.
- Implementing **iteration and search operations** using loops and iterators.