



## Experiment 4

**Student Name:** Jorawar Singh

**Branch:**BE/CSE

**Semester:** 6

**Subject Name:**JAVA

**UID:**22BCS13118

**Section/Group:** 801/B

**Date of Performance:**17/01/25

**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.

### 2. 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 scanner = new Scanner(System.in);
        List<Employee> employees = new ArrayList<>();

        while (true) {
            System.out.println("1. Add Employee");
```

```
System.out.println("2. Update Employee");
System.out.println("3. Remove Employee");
System.out.println("4. Search Employee");
System.out.println("5. Display All Employees");
System.out.println("6. Exit");
System.out.print("Enter your choice: ");
```

```
int choice = scanner.nextInt();
switch (choice) {
    case 1:
        System.out.print("Enter ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Salary: ");
        double salary = scanner.nextDouble();
        employees.add(new Employee(id, name, salary));
        System.out.println("Employee added successfully!");
        break;

    case 2:
        System.out.print("Enter Employee ID to update: ");
        int updateId = scanner.nextInt();
        boolean found = false;
        for (Employee emp : employees) {
            if (emp.id == updateId) {
                scanner.nextLine();
                System.out.print("Enter new Name: ");
                emp.name = scanner.nextLine();
                System.out.print("Enter new Salary: ");
                emp.salary = scanner.nextDouble();
                System.out.println("Employee updated successfully!");
                found = true;
                break;
            }
        }
        if (!found) {
            System.out.println("Employee not found!");
        }
        break;
```

**case 3:**

```
System.out.print("Enter Employee ID to remove: ");
int removeId = scanner.nextInt();
Iterator<Employee> iterator = employees.iterator();
boolean removed = false;
while (iterator.hasNext()) {
    Employee emp = iterator.next();
    if (emp.id == removeId) {
        iterator.remove();
        System.out.println("Employee removed successfully!");
        removed = true;
        break;
    }
}
if (!removed) {
    System.out.println("Employee not found!");
}
break;
```

**case 4:**

```
System.out.print("Enter Employee ID to search: ");
int searchId = scanner.nextInt();
boolean foundSearch = false;
for (Employee emp : employees) {
    if (emp.id == searchId) {
        System.out.println("Employee Found: " + emp);
        foundSearch = true;
        break;
    }
}
if (!foundSearch) {
    System.out.println("Employee not found!");
}
break;
```

**case 5:**

```
if (employees.isEmpty()) {
    System.out.println("No employees found.");
} else {
    System.out.println("Employee List:");
    for (Employee emp : employees) {
        System.out.println(emp);
    }
}
```

```
        }  
    }  
    break;  
  
    case 6:  
        System.out.println("Exiting program...");  
        scanner.close();  
        return;  
  
    default:  
        System.out.println("Invalid choice! Please try again.");  
    }  
}  
}
```

### 3. Output

```
1. Add Employee  
2. Update Employee  
3. Remove Employee  
4. Search Employee  
5. Display All Employees  
6. Exit  
Enter your choice: 1  
Enter ID: 12  
Enter Name: dipesh  
Enter Salary: 500000  
Employee added successfully!  
1. Add Employee  
2. Update Employee  
3. Remove Employee  
4. Search Employee  
5. Display All Employees  
6. Exit  
Enter your choice: █
```

### 4. Learning Outcome

1. Proficiency in Java Basics
2. Understanding Data Structures
3. Improved Problem-Solving Skills
4. Application of Encapsulation and Modularity: