



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 4.1

Student Name: Somnath
Branch: BE-CSE
Semester: 6th
Subject Name: JAVA
Subject Code: 22CSP-359

UID: 22BCS12737
Section/Group: IOT-605-A
Date of Performance: 19/2/25

1.Aim: Easy Level: Employee Management System Problem Statement

Write a Java program to implement an ArrayList that stores employee details (ID, Name, and Salary). Allow users to:

Add employees

Update employee details

Remove employees

Search for employees

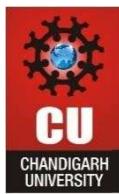
Key Concepts Used ArrayList: To store employee objects.

Encapsulation: Employee details are stored in a class with private fields and public getters/setters.

User Interaction: Using Scanner for input/output operations.

3. Code

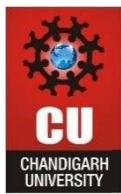
```
import java.util.ArrayList;  
import java.util.Scanner;  
  
class Employee {  
    private int id;  
    private String name;  
    private double salary;  
  
    public Employee(int id, String name, double salary) {  
        this.id = id;  
        this.name = name;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

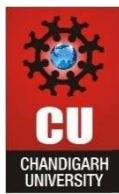
```
        this.salary = salary;  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public double getSalary() {  
        return salary;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public void setSalary(double salary) {  
        this.salary = salary;  
    }  
  
    @Override  
    public String toString() {  
        return "Employee ID: " + id + ", Name: " + name + ", Salary: " + salary;  
    }  
  
public class EmployeeManagement {  
    private static ArrayList<Employee> employees = new ArrayList<>();  
    private static Scanner scanner = new Scanner(System.in);  
  
    public static void main(String[] args) {
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
boolean exit = false;  
while (!exit) {  
    System.out.println("\nEmployee Management System");  
    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. Exit");  
    System.out.print("Choose an option: ");  
    int choice = scanner.nextInt();  
    scanner.nextLine(); // Consume newline  
  
    switch (choice) {  
        case 1:  
            addEmployee();  
            break;  
        case 2:  
            updateEmployee();  
            break;  
        case 3:  
            removeEmployee();  
            break;  
        case 4:  
            searchEmployee();  
            break;  
        case 5:  
            exit = true;  
            break;  
        default:  
            System.out.println("Invalid choice. Please try again.");  
    }  
}  
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

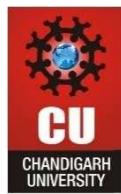
```
private static void addEmployee() {  
    System.out.print("Enter Employee ID: ");  
    int id = scanner.nextInt();  
    scanner.nextLine(); // Consume newline  
    System.out.print("Enter Employee Name: ");  
    String name = scanner.nextLine();  
    System.out.print("Enter Employee Salary: ");  
    double salary = scanner.nextDouble();  
  
    employees.add(new Employee(id, name, salary));  
    System.out.println("Employee added successfully.");  
}  
  
private static void updateEmployee() {  
    System.out.print("Enter Employee ID to update: ");  
    int id = scanner.nextInt();  
    scanner.nextLine(); // Consume newline  
  
    for (Employee emp : employees) {  
        if (emp.getId() == id) {  
            System.out.print("Enter new name: ");  
            String name = scanner.nextLine();  
            System.out.print("Enter new salary: ");  
            double salary = scanner.nextDouble();  
  
            emp.setName(name);  
            emp.setSalary(salary);  
            System.out.println("Employee updated successfully.");  
            return;  
        }  
    }  
    System.out.println("Employee not found.");  
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
private static void removeEmployee() {  
    System.out.print("Enter Employee ID to remove: ");  
    int id = scanner.nextInt();  
  
    for (Employee emp : employees) {  
        if (emp.getId() == id) {  
            employees.remove(emp);  
            System.out.println("Employee removed successfully.");  
            return;  
        }  
    }  
    System.out.println("Employee not found.");  
}  
  
private static void searchEmployee() {  
    System.out.print("Enter Employee ID to search: ");  
    int id = scanner.nextInt();  
  
    for (Employee emp : employees) {  
        if (emp.getId() == id) {  
            System.out.println(emp);  
            return;  
        }  
    }  
    System.out.println("Employee not found.");  
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

4.Output:

```
Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Exit
Choose an option: 1
Enter Employee ID: 12737
Enter Employee Name: Somnath
Enter Employee Salary: 72000
Employee added successfully.

Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Exit
Choose an option: 1
Enter Employee ID: 12729
Enter Employee Name: Priti
Enter Employee Salary: 80000
Employee added successfully.

Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Exit
Choose an option: 4
Enter Employee ID to search: 12737
Employee ID: 12737, Name: Somnath, Salary: 72000.0

Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Exit
Choose an option: 5
```