**Experiment - 4**

**Name: Armaan UID: 22BCS12007**

**Aim:** To solve JAVA problems

1. Problem : EmployeeManagement.java (Easy Level)

Code:

import java.util.ArrayList;

import java.util.Scanner;

// Employee class with encapsulation

class Employee {

private int id;

private String name;

private double salary;

// Constructor

public Employee(int id, String name, double salary) {

this.id = id;

this.name = name;

this.salary = salary;

}

// Getters and Setters

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;

}

// Display employee details

public void display() {

System.out.println("ID: " + id + ", Name: " + name + ", Salary: " + salary);

}

}

public class EmployeeManagement {

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

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

int choice;

do {

System.out.println("\n=== Employee 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. Display All Employees");

System.out.println("6. Exit");

System.out.print("Enter your choice: ");

choice = getIntInput();

switch (choice) {

case 1 -> addEmployee();

case 2 -> updateEmployee();

case 3 -> removeEmployee();

case 4 -> searchEmployee();

case 5 -> displayAllEmployees();

case 6 -> System.out.println("Exiting... Goodbye!");

default -> System.out.println("Invalid choice. Please try again.");

}

} while (choice != 6);

}

private static void addEmployee() {

System.out.print("Enter Employee ID: ");

int id = getIntInput();

System.out.print("Enter Employee Name: ");

String name = scanner.nextLine();

System.out.print("Enter Employee Salary: ");

double salary = getDoubleInput();

Employee emp = new Employee(id, name, salary);

employeeList.add(emp);

System.out.println("✅ Employee added successfully!");

}

private static void updateEmployee() {

System.out.print("Enter Employee ID to update: ");

int id = getIntInput();

Employee emp = findEmployeeById(id);

if (emp != null) {

System.out.print("Enter new name: ");

emp.setName(scanner.nextLine());

System.out.print("Enter new salary: ");

emp.setSalary(getDoubleInput());

System.out.println("✅ Employee updated successfully!");

} else {

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

}

}

private static void removeEmployee() {

System.out.print("Enter Employee ID to remove: ");

int id = getIntInput();

Employee emp = findEmployeeById(id);

if (emp != null) {

employeeList.remove(emp);

System.out.println("✅ Employee removed successfully!");

} else {

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

}

}

private static void searchEmployee() {

System.out.print("Enter Employee ID to search: ");

int id = getIntInput();

Employee emp = findEmployeeById(id);

if (emp != null) {

System.out.println("🎯 Employee Found:");

emp.display();

} else {

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

}

}

private static void displayAllEmployees() {

if (employeeList.isEmpty()) {

System.out.println("📭 No employees to display.");

} else {

System.out.println("📋 All Employees:");

for (Employee emp : employeeList) {

emp.display();

}

}

}

private static Employee findEmployeeById(int id) {

for (Employee emp : employeeList) {

if (emp.getId() == id) {

return emp;

}

}

return null;

}

private static int getIntInput() {

while (!scanner.hasNextInt()) {

System.out.print("❗ Please enter a valid number: ");

scanner.next(); // discard invalid input

}

int val = scanner.nextInt();

scanner.nextLine(); // consume newline

return val;

}

private static double getDoubleInput() {

while (!scanner.hasNextDouble()) {

System.out.print("❗ Please enter a valid decimal number: ");

scanner.next(); // discard invalid input

}

double val = scanner.nextDouble();

scanner.nextLine(); // consume newline

return val;

}

}

Output :

