**Exercise 4: Employee Management System**

**Understanding the Problem:**  
Managing employees requires operations like adding, searching, updating, and deleting employee records. Using arrays is a simple but limited method for this.

**Setup and Implementation:**  
The Employee class contains:

* employeeId (int)
* name (String)
* position (String)
* salary (double)

A fixed-size array is used to store employees. Scanner is used to take user input for all operations.

**Java Code:**

import java.util.\*;

class Employee {

int employeeId;

String name;

String position;

double salary;

Employee(int id, String n, String pos, double sal) {

employeeId = id;

name = n;

position = pos;

salary = sal;

}}

public class EmployeeManagement {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Employee[] employees = new Employee[100];

int count = 0;

while (true) {

System.out.println("1. Add 2. Search 3. View All 4. Delete 5. Exit");

int ch = sc.nextInt();

if (ch == 1) {

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

int id = sc.nextInt();

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

String name = sc.next();

System.out.print("Enter Position: ");

String pos = sc.next();

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

double sal = sc.nextDouble();

employees[count++] = new Employee(id, name, pos, sal);

} else if (ch == 2) {

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

int id = sc.nextInt();

for (int i = 0; i < count; i++) {

if (employees[i].employeeId == id) {

System.out.println(employees[i].employeeId + " " + employees[i].name + " " + employees[i].position + " " + employees[i].salary);

}

}

} else if (ch == 3) {

for (int i = 0; i < count; i++) {

System.out.println(employees[i].employeeId + " " + employees[i].name + " " + employees[i].position + " " + employees[i].salary);

}

} else if (ch == 4) {

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

int id = sc.nextInt();

for (int i = 0; i < count; i++) {

if (employees[i].employeeId == id) {

for (int j = i; j < count - 1; j++) {

employees[j] = employees[j + 1];

}

count--;

break;

}

}

} else break;

}

}

}

**Time Complexity Analysis:**

* Add: O(1)
* Search: O(n)
* Traverse: O(n)
* Delete: O(n) (due to shifting)

**Optimization Discussion:**  
Arrays are limited by size and lack dynamic flexibility. If the number of employees grows, using ArrayList or HashMap<Integer, Employee> would allow for dynamic resizing and faster searches (O(1) for HashMap).