**Exercise 4: Employee Management System**

Scenario: Developing an employee management system for a company. Efficiently managing employee records is crucial.

1. Understand Array Representation: Arrays are contiguous memory blocks that store elements of the same type. Advantages include:

* Constant time access: O(1) for accessing elements using an index.
* Memory locality: Speeds up data access.

However, arrays have limitations:

* Fixed size: Cannot dynamically expand or shrink.
* Insertions and deletions are expensive (O(n)) if not at the end.

1. Setup: Create a class Employee with the following attributes:

* String employeeId
* String name
* String position
* double salary

1. Implementation:

**Java Code:**

class Employee {

String employeeId;

String name;

String position;

double salary;

public Employee(String employeeId, String name, String position, double salary) {

this.employeeId = employeeId;

this.name = name;

this.position = position;

this.salary = salary;

}

public String toString() {

return employeeId + " | " + name + " | " + position + " | ₹" + salary;

}

}

class EmployeeManagement {

Employee[] employees = new Employee[100];

int count = 0;

void addEmployee(Employee e) {

employees[count++] = e;

}

void searchEmployee(String empId) {

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

if (employees[i].employeeId.equals(empId)) {

System.out.println(employees[i]);

return;

}

}

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

}

void deleteEmployee(String empId) {

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

if (employees[i].employeeId.equals(empId)) {

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

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

}

employees[--count] = null;

System.out.println("Employee deleted.");

return;

}

}

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

}

void displayEmployees() {

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

System.out.println(employees[i]);

}

}

public static void main(String[] args) {

EmployeeManagement ems = new EmployeeManagement();

ems.addEmployee(new Employee("E001", "John Doe", "Manager", 75000));

ems.addEmployee(new Employee("E002", "Jane Smith", "Engineer", 55000));

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

ems.displayEmployees();

System.out.println("\nSearching for E002:");

ems.searchEmployee("E002");

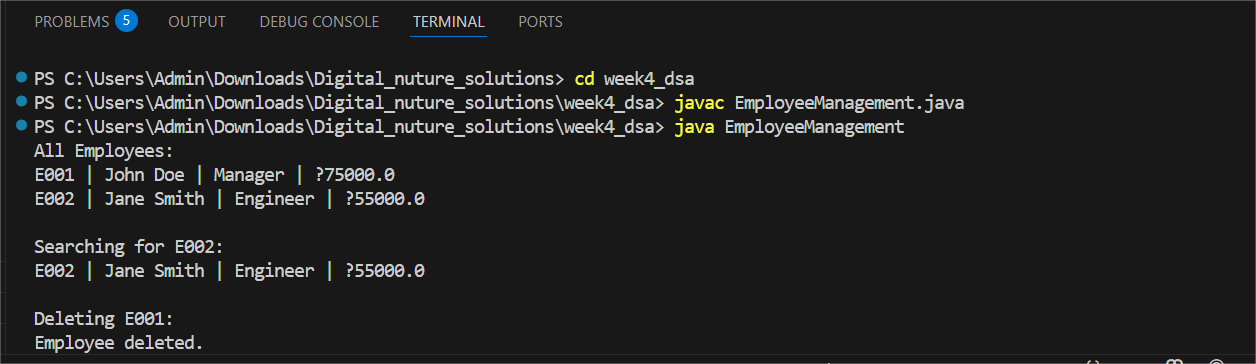
System.out.println("\nDeleting E001:");

ems.deleteEmployee("E001");

ems.displayEmployees();

}

}  
  
OUTPUT:



1. Analysis:

* **Add:** O(1) (adding at end).
* **Search:** O(n) (linear search).
* **Traverse:** O(n).
* **Delete:** O(n) (shift required).

Limitations of arrays:

* Cannot shrink or expand dynamically.
* Insertion and deletion require shifting elements.

For dynamic and large datasets, consider using ArrayList or LinkedList.