LAB MANUAL

Experiment: Multidimensional Arrays (2D and 3D)

AIDS II SEM

OOPS with JAVA 22AID111

Lab Instructor: Pooja Gowda

- **Problems 1–3 → 2D arrays (Solved, scenario-based, with output)**
- **Problems 4–5 → 3D arrays (Solved, simple, with clear explanation of [][][])**
- **Problem 6 → 3D array (Scenario-based, challenging, UNSOLVED)**

PROBLEM 1 (2D Array – Solved)

Scenario: Daily Sales Analysis

Problem Statement

A shop records daily sales for **3 products over 4 days**.
Calculate the **total sales for each product**.

Program

```
class SalesAnalysis {


  int[][] sales = {
     {10, 12, 15, 11},
     {8, 9, 10, 12},
     {14, 13, 16, 15}
  };


  void compute() {
     for (int i = 0; i < sales.length; i++) {
        int sum = 0;
        for (int j = 0; j < sales[i].length; j++) {
```

```
            sum += sales[i][j];
        }
        System.out.println("Product " + i + " total sales = " + sum);
    }
  }

  public static void main(String[] args) {
    new SalesAnalysis().compute();
  }
}
```

Expected Output

Product 0 total sales = 48

Product 1 total sales = 39

Product 2 total sales = 58

PROBLEM 2 (2D Array – Solved)

Scenario: Student Attendance Tracking

Problem Statement

Attendance of **3 students over 5 days** is recorded (1 = present, 0 = absent).
Calculate the **total attendance of each student**.

Program

```
class AttendanceAnalysis {

  int[][] attendance = {
    {1, 1, 0, 1, 1},
    {1, 0, 1, 1, 0},
    {1, 1, 1, 1, 1}
  };
```

```
  void compute() {
    for (int i = 0; i < attendance.length; i++) {
      int count = 0;
      for (int j = 0; j < attendance[i].length; j++) {
        count += attendance[i][j];
      }
      System.out.println("Student " + i + " attendance = " + count);
    }
  }

  public static void main(String[] args) {
    new AttendanceAnalysis().compute();
  }
}
```

Expected Output

Student 0 attendance = 4

Student 1 attendance = 3

Student 2 attendance = 5

PROBLEM 3 (2D Array – Solved, Analytical)

Scenario: Classroom Height Analysis

Problem Statement

Heights of students are recorded in a **3×3 classroom grid**.
Find the **maximum height and class average**.

Program

```
class HeightAnalysis {

  int[][] heights = {
    {150, 152, 148},
    {160, 158, 155},
    {165, 170, 168}
  };

  void compute() {
    int max = heights[0][0];
    int sum = 0;
    int count = 0;

    for (int i = 0; i < heights.length; i++) {
      for (int j = 0; j < heights[i].length; j++) {
        sum += heights[i][j];
        count++;
        if (heights[i][j] > max) {
          max = heights[i][j];
        }
      }
    }

    System.out.println("Maximum height = " + max);
    System.out.println("Average height = " + (double) sum / count);
  }

  public static void main(String[] args) {
    new HeightAnalysis().compute();
```

```
    }
}
```

Expected Output

Maximum height = 170

Average height = 159.55555555555554

3D ARRAY – Nested Matrices

1. Recap: 2D Array as a Matrix

A 2D array in Java directly corresponds to a matrix in mathematics.

int[][] A;

This represents a matrix where:

- A[i][j]
- i → row index
- j → column index

Example

```
int[][] A = {
   {10, 20, 30},
   {40, 50, 60},
   {70, 80, 90}
};
```

---

2. Why a 3D Array Is NOT Just a Bigger Matrix

A 3D array is not a single matrix.

A 3D array is a collection (stack) of matrices, grouped together.

---

3. Meaning of [][][] in Matrix Terms

int[][][] B;

This means:

- B is an array of 2D matrices
- Each B[i] is one complete matrix

Accessing an element:

B[i][j][k]

means:

Index	Matrix Interpretation
i	Which matrix (layer)
j	Row in that matrix
k	Column in that row

---

4. 3D Array as a Stack of Matrices

Example

```
int[][][] B = {

  {  // Matrix 0
    {1, 2, 3},
    {4, 5, 6}
  },

  {  // Matrix 1
    {7, 8, 9},
    {10, 11, 12}
  }
};
```

This represents two separate matrices.

This represents **two separate matrices**.

Matrix 0

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Matrix 1

$$\begin{bmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$$

---

5. Access Mapping in 3D array

Java Expression	**Matrix Meaning**
B[0]	Entire Matrix 0
B[0][1]	Row 1 of Matrix 0
B[0][1][2]	Column 2 of Row 1 in Matrix 0
B[1][0][1]	Column 1 of Row 0 in Matrix 1

6. Loop Structure Using Matrix Thinking

Traversal of a 3D array always follows this logic:

for each matrix

  for each row in that matrix

    for each column in that row

PROBLEM 4 (3D Array – Solved, Simple)

Scenario: Temperature Monitoring

Explanation of [][][]

temperature[room][day][reading]

- First [] → Room
- Second [] → Day

- Third [] → Reading

Program

```
class TemperatureMonitor {

  int[][][] temperature = {
    {{30, 32}, {33, 34}},
    {{28, 29}, {30, 31}}
  };

  void compute() {
    for (int room = 0; room < temperature.length; room++) {
      int sum = 0;
      int count = 0;

      for (int day = 0; day < temperature[room].length; day++) {
        for (int r = 0; r < temperature[room][day].length; r++) {
          sum += temperature[room][day][r];
          count++;
        }
      }
      System.out.println("Room " + room + " average = " + (double) sum / count);
    }
  }

  public static void main(String[] args) {
    new TemperatureMonitor().compute();
  }
}
```

Expected Output

Room 0 average = 32.25

Room 1 average = 29.5

PROBLEM 5 (3D Array)

Scenario: Student Marks Analysis

Explanation of [][][]

marks[class][student][subject]

Program

```
class MarksAnalysis {

  int[][][] marks = {
    {{80, 85}, {70, 75}},
    {{90, 95}, {88, 92}}
  };

  void compute() {
    for (int c = 0; c < marks.length; c++) {
      for (int s = 0; s < marks[c].length; s++) {
        int total = 0;
        for (int sub = 0; sub < marks[c][s].length; sub++) {
          total += marks[c][s][sub];
        }
        System.out.println("Class " + c + " Student " + s + " total = " + total);
      }
    }
  }
```

```
    public static void main(String[] args) {
        new MarksAnalysis().compute();
    }
}
```

Expected Output

Class 0 Student 0 total = 165

Class 0 Student 1 total = 145

Class 1 Student 0 total = 185

Class 1 Student 1 total = 180

PROBLEM 6 (3D Array – Scenario-Based, UNSOLVED)

Scenario: Hospital Bed Occupancy Analysis

Problem Description

A hospital records bed occupancy data across:

- Wards
- Days
- Time slots

occupancy[ward][day][timeSlot]

Tasks – To be solved during lab

Write **pseudocode first**, then code.

1. Identify overloaded wards (occupancy ≥ 45).
2. Identify critical days (average > 40).
3. Determine if hospital is under stress.
4. Find maximum occupancy and its position.

Expected Output Format

Ward 0 overloaded = true

Ward 1 overloaded = true

Ward 2 overloaded = false

Hospital under stress = true

Maximum occupancy = 50 at Ward 0, Day 1, Time 2

DELIVERABLE (MANDATORY)

Submit **ONE Word document on Teams** containing:

1. Write Steps for all problems – Pseudocode that explains those codes in steps
2. Java code (screenshots) – Run the all codes
3. Output screenshots
4. Clear headings for each problem