**COSC 1436: Programming Fundamentals I  
Lab 7: 2D Arrays**

As usual, please submit one source code file (.java file) for each exercise. (Ex. Lab7Ex1.java, Lab7Ex2,java, ...etc).

Remember to follow the commenting guidelines.

**Program 1: Section differences (41 points):**

An instructor is teaching 3 different sections of the same class: one starts at 8am, one starts at 11am, and the last one starts at 1pm. Other instructors have said that the worst time to have a class is early afternoon because everyone is falling asleep after lunch. However, our instructor is not so sure. The course has 4 tests. The average test grades for the 8am class (section 1) are: **83.1**, **85.6**, **87.3**, and **86.6**. For the 11am class (section 2), the average grades are: **85.1**, **84.5**, **89.2**, and **87.2**. The 1pm class (section 3) average grades are: **82.2**, **88.4**, **88.1**, and **85.3**.

Create a program that store the test grades in a 2D array and calculates the average test score for each section and prints out which section had the highest and which had the lowest average

Sample output:

Section 1 average = 85.4

Section 2 average = 86.5

Section 3 average = 86.0

Section 1 had the lowest average

Section 2 had the highest average

public class testprograms{

public static void main(String[] args){

double[][] tests={{83.1, 85.6, 87.3,86.6},

{85.1, 84.5, 89.2,87.2},

{82.2, 88.4, 88.1, 85.3}};

int row=0;

int col=0;

double sum=0;

double average=0;

double highest=0;

double lowest=100;

int highest\_section=1;

int lowest\_section=1;

for(row=0;row<3;row++){

sum=0;

for(col=0;col<4;col++){

sum=sum+tests[row][col];

}

average=sum/tests[row].length;

System.out.println("Section "+(row+1)+" average= "+average);

if (average>highest){

highest=average;

highest\_section=row+1;

}

if (average<lowest){

lowest=average;

lowest\_section=row+1;

}

}

System.out.println("Section "+lowest\_section+" had the lowest average");

System.out.println("Section "+highest\_section+" had the highest average");

}

}

**Program 2: Magic Square (41 pts)**

Magic squares are a mathematical curiosity that have appeared across Asia, the Middle East and Europe many times over the last 2100 years. They are a square grid of whole numbers where the sum of each row, the sum of each column, and the sum of each major diagonal are all the same. Below is the 4x4 magic square inscribed in the Parshavnath temple in India from the 12th century.

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 12 | 1 | 14 |
| 2 | 13 | 8 | 11 |
| 16 | 3 | 10 | 5 |
| 9 | 6 | 15 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 12 | 1 | 14 |
| 2 | 13 | 8 | 11 |
| 16 | 3 | 10 | 5 |
| 9 | 6 | 15 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 12 | 1 | 14 |
| 2 | 13 | 8 | 11 |
| 16 | 3 | 10 | 5 |
| 9 | 6 | 15 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 12 | 1 | 14 |
| 2 | 13 | 8 | 11 |
| 16 | 3 | 10 | 5 |
| 9 | 6 | 15 | 4 |

34

34

34

34

34

34

34 34 34 34

In a program, simulate the Parshavnath magic square using a 2D array. Populate the array with the above values and then verify that the sum of all of the columns, the rows, and the main diagonals are all the same.

Sample output:

The sum of row 1 is: 34

The sum of row 2 is: 34

The sum of row 3 is: 34

The sum of row 4 is: 34

The sum of column 1 is: 34

The sum of column 2 is: 34

The sum of column 3 is: 34

The sum of column 4 is: 34

The sum of diagonal 1 is: 34

The sum of diagonal 2 is: 34

All the sums are 34. It is a magic square!

Hint: Declare a new array to hold the sum of each row, and another array to hold the sum of each column. There are only 2 diagonals, so you can use an array or not. You are not making things much easier by using an array there.

public class Lab07Prog2{

public static void main(String[] args){

int[][] square={{7,12,1,14},

{2,13,8,11},

{16,3,10,5},

{9,6,15,4}};

int[] row\_sum={0,0,0,0};

int[] col\_sum={0,0,0,0};

int diag1\_sum=0;

int diag2\_sum=0;

int magic\_sum=0;

boolean magic=true;

int row=0;

int col=0;

int i=0;

for(row=0;row<4;row++){

for(col=0;col<4;col++){

row\_sum[row]=row\_sum[row]+square[row][col];

col\_sum[col]=col\_sum[col]+square[row][col];

}

}

for(i=0;i<4;i++){

System.out.println("The sum of row " +(i+1)+ " is: "+row\_sum[i]);

}

for(i=0;i<4;i++){

System.out.println("The sum of column " +(i+1)+ " is: "+col\_sum[i]);

}

for(i=0;i<4;i++){

diag1\_sum=diag1\_sum+square[i][i];

diag2\_sum=diag2\_sum+square[i][4-1-i];

}

System.out.println("The sum of diagonal 1 is: "+diag1\_sum);

System.out.println("The sum of diagonal 2 is: "+diag2\_sum);

magic\_sum=diag1\_sum;

if (diag2\_sum != magic\_sum){

magic=false;

}

for(i=0;i<4;i++){

if ((row\_sum[i]!=magic\_sum)||(col\_sum[i]!=magic\_sum)){

magic=false;

}

}

if (magic==true){

System.out.print("All the sums are "+magic\_sum+". It is a magic square!");

}

else{

System.out.print("Sorry, not a magic square :(");

}

}

}

**Exercise 3: Multiple Choice & True/False (18 points)**

1. Fill in the array with the values that would be stored after the code executes:

char[][] ch = new char[2][4];

int j=0;

int k=0;

for (j=0; j<2;j++){

for(k=0;k<4;k++){

ch[j][k]='0';

System.out.print(ch[j][k]);

}

}

char c= 'a';

ch[0][1] = 'e';

ch[1][3] = 'H';

ch[0][2] = '5';

ch[1][2] = ch[1][3];

c = ch[0][2];

ch[0][2] = 'M';

ch[1][2] = c;

ch[0][0] = '?';

ch[1][1] = '0’;

ch[1][1] = ch[0][0];

In your .java file, enter your answer as a list like:

ch[0][0] =

|  |  |  |  |
| --- | --- | --- | --- |
| ? | e | M | 0 |
| 0 | ? | 5 | H |

ch[0][1] =

ch[0][2] =

ch[0][3] =

…

ch[1][3] =

1. Which of the following choices is the correct syntax for declaring/initializing a 2D array of 6 double numbers?
   * 1. double[][] x=new double[1][2];
     2. double[2][3] x=new double[][];
     3. double[ , ] x = new double[2,3];
     4. double x[2][3] = new double[2][3];
     5. double[][] x=new double[2][3];
2. Fill in the array with the values that would be stored after the code executes:

int[] list = {21, -1, 0, 18, 6, -4, 5, 1};

int i=0;

for (i = 0; i < list.length; i++){

list[i] = list[i] + (list[i] / list[0]);

}

As before, in your .java file, enter your answer as a list like:

list[0] =

list[1] =

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | -1 | 0 | 18 | 6 | -4 | 5 | 1 |

…

list[7] =

1. Which of the following choices is the correct syntax for quickly declaring/initializing an array of String to store a particular list of strings?
2. String[] a = {"Bob", "Bill", "Susan", "Steven", "Jean"};
3. String a {"Bob", "Bill", "Susan", "Steven", "Jean"};
4. String[] a = new String[5] {"Bob", "Bill", "Susan", "Steven", "Jean"};
5. String[5] a = {"Bob", "Bill", "Susan", "Steven", "Jean"};
6. String[] a = new {"Bob", "Bill", "Susan", "Steven", "Jean"} [5];

5. Will the below lines code compile? Yes or No?. If No, explain why not.

int[] data = {67, 24, -16, 33, -45};

int[] myArray = new int[data.length];

yes(true). Data.length is defined once the data array is created