SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

Department of Computer Engineering

Somaiya TRUST

Batch: D2 **Roll No.:**16010122323

Experiment / assignment / tutorial No. 03

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE: Multi-dimensional Arrays (Jagged Array)

AIM: Write a program which stores information about n players in a two-dimensional array. The array should contain number of rows equal to number of players. Each row will have number of columns equal to number of matches played by that player which may vary from player to player. The program should display player number (index +1), runs scored in all matches and its batting average as output. (It is expected to assign columns to each row dynamically after getting value from user.

Expected OUTCOME of Experiment:

CO2: Explore arrays, vectors, classes and objects in C++ and Java.

Books/ Journals/ Websites referred:

1. Ralph Bravaco , Shai Simoson , "Java Programing From the Group Up" Tata McGraw-Hill.
2. Grady Booch, Object Oriented Analysis and Design .

Pre Lab/ Prior Concepts:

Arrays

Multi-Dimensional Array:

10	12	43	11	22
20	45	56	1	33
30	67	32	14	44
40	12	87	14	55
50	86	66	13	66
60	53	44	12	11

A multi-dimensional array is one that can hold all the values above. You set them up like this:

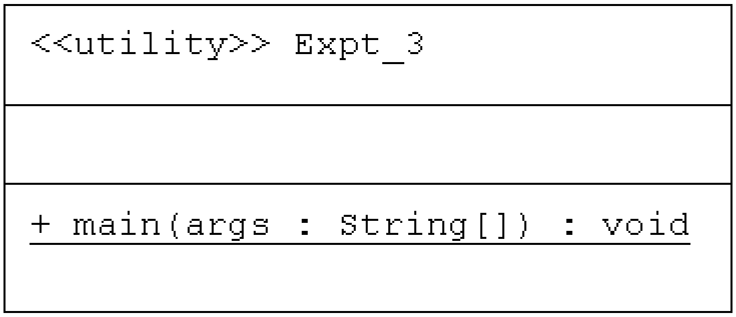
int[ ][ ] numbers = new int[6]**[5];**

The first set of square brackets is for the rows and the second set of square brackets is for the columns. In the above line of code, we're telling Java to set up an array with 6 rows and 5 columns.

aryNumbers[0][0] = 10;
aryNumbers[0][1] = 12;
aryNumbers[0][2] = 43;
aryNumbers[0][3] = 11;
aryNumbers[0][4] = 22;

So the first row is row 0. The columns then go from 0 to 4, which is 5 items.

Class Diagram:



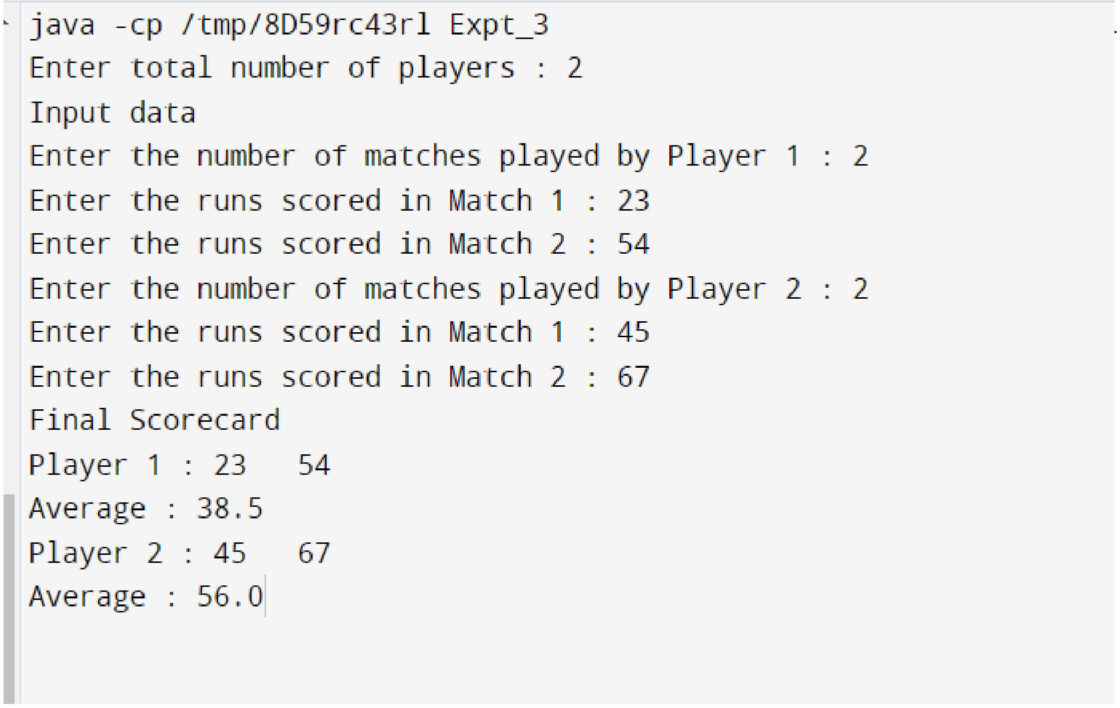
Algorithm:

1. Start
2. Get number of players from the user.
3. Declare an array with number of rows equal to the number of players.
4. Get the number of matches played by each player.
5. Declare the columns of the array equal to the number of matches played by each player.
6. Get the runs scored by each player in the respective row.
7. Print the runs by each player.
8. Find sum of runs scored by player in each match.
9. Get average by dividing sum by number of matches played by each player.
10. Printing the average for each player.
11. Exit

Implementation details:

```
import java.util.Scanner;


public class Expt_3 {
    public static void main(String[] args) {


        Scanner sc = new Scanner(System.in);
        System.out.print("Enter total number of players : ");
        int n = sc.nextInt();
        int[][]arr = new int[n][]; //Taking Rows equal to number of players


        System.out.println("Input data");
        for (int i=0;i<n;i++){
            System.out.print("Enter the number of matches played by Player "+(i+1)+" : ");
            int m = sc.nextInt();
            arr[i] = new int [m]; //Taking Columns equal to number of matches played by player


            for (int j=0;j<m;j++){
                System.out.print("Enter the runs scored in Match "+(j+1)+" : ");
                int run= sc.nextInt();
                arr[i][j]=run; //Taking runs by player
            }
        }
        System.out.println("Final Scorecard");
        for (int i=0;i<n;i++){
            System.out.print("Player "+(i+1)+" : ");
            for (int j=0;j<arr[i].length;j++)
                System.out.printf("%d\t",arr[i][j]); //Showing the runs in    matches played
            System.out.println();


            float sum=0,av=0;
            for (int j = 0; j < arr[i].length ; j++) {
                sum=sum+arr[i][j];
                av = sum / arr[i].length; //Calculating Average
            }
            System.out.println("Average : "+av); //Showing Average for matches played
        }
    }
```

Output:



```
java -cp /tmp/8D59rc43rl Expt_3
Enter total number of players : 2
Input data
Enter the number of matches played by Player 1 : 2
Enter the runs scored in Match 1 : 23
Enter the runs scored in Match 2 : 54
Enter the number of matches played by Player 2 : 2
Enter the runs scored in Match 1 : 45
Enter the runs scored in Match 2 : 67
Final Scorecard
Player 1 : 23   54
Average : 38.5
Player 2 : 45   67
Average : 56.0
```

Conclusion:

The aim of the experiment is verified.

The code is correct and implemented without error(s).

Post Lab Descriptive Questions

Q.1 Which of the following statements are valid array declaration?

(A) int number();
(B) float average[];
(C) double[] marks;
(D) counter int[];

(i) (D)
(ii) (A) & (C)
(iii) (A)
(iv) (B)&(C)

Ans: (iv) (B)&(C)

Q.2 Consider the following code

int number[] = new int[5];

After execution of this statement, which of the following are true?

(A) number[0] is undefined
(B) number[5] is undefined
(C) number[4] is null
(D) number[2] is 0
(E) number.length() is 5

(i) (C) & (E)
(ii) (A) & (E)
(iii) (E)
(iv) (B), (D) & (E)

Ans: (iv) (B), (D) & (E)

Q.3 What will be the content of array variable table after executing the following code?

```
for(int i = 0; i < 3; i + +)
for(int j = 0; j < 3; j + +)
if(j= = i) table[i][j] = 1;
else table[i][j] = 0;
```

A).

0 0 0
0 0 0
0 0 0

B).

0 0 1
0 1 0
1 0 0

C).

1 0 0
1 1 0
1 1 1

D)

1 0 0
0 1 0
0 0 1

Ans: D)