OOP

Lab Manual (Lab 3)

**A logo with a black background

Description automatically generated**

Session: Spring 2024

LAB INSTRUCTOR: AYESHA MAJID ALI

**Java Tutorial: Multidimensional Arrays in Java**

A multidimensional array is an array of arrays.

Multidimensional arrays are useful when you want to store data as a tabular form, like a table with rows and columns.

To create a two-dimensional array, add each array within its own set of **curly braces**:

A close up of a number

Description automatically generated

**myNumbers** is now an array with two arrays as its elements

Access Elements

To access the elements of the **myNumbers** array, specify two indexes:

One for the array, and one for the element inside that array.

This example accesses the third element (2) in the second array (1) of myNumbers:

A black text with blue and purple text

Description automatically generated with medium confidence

Change Element Values

You can also change the value of an element:

A screenshot of a computer code

Description automatically generated

Loop Through a Multi-Dimensional Array

We can also use a for loop inside another for loop to get the elements of a two-dimensional array (we still have to point to the two indexes):

A screenshot of a computer code

Description automatically generated

**How to pass Arrays to Methods in Java?**

You can pass arrays to a method just like normal variables. When we pass an array to a method as an argument, actually the address of the array in the memory is passed (reference). Therefore, any changes to this array in the method will affect the array.

Suppose we have two methods [**min()**](https://www.tutorialspoint.com/java/number_min.htm)and [**max()**](https://www.tutorialspoint.com/java/number_max.htm) which accepts an array and these methods calculates the minimum and maximum values of the given array respectively:

**Sample Program:**

**class** array

{

**public** **static** **void** main(String[] args)

{

**int**[] n={12,24,2,89,34,45};

System.***out***.println("Before sorting");

*display*(n);

*sort*(n);

System.***out***.println("\n After Sorting :");

*display*(n);

}

**static** **void** display(**int** n[])

{

**for**(**int** i=0; i<n.length;i++)

System.***out***.print(n[i] + " ");

}

**static** **void** sort(**int** n[])

{

**int** i, j, temp;

**for**(i=0; i<n.length-i;i++)

{

**for**(j=0; j<n.length-i-1;j++)

{

**if**(n[j]>n[j+1])

{

temp = n[j];

n[j] = n[j+1];

n[j+1] = temp;

}

}

}

}

}

**TASKS**

1. Create a simple Rock, Paper Scissors game in Java. (#Use Conditional Statements)
2. Write a program to find out whether a given integer is present in an array or not. Use a separate class to perform this task.
3. Write a Java program to find the maximum and minimum element in an array. Use a separate Class and methods to perform this task. Take array as an input from the user and pass it to the other method of the class. Make sure that the other class is present in a separate java file.