 Contact: 8087883669

Java Array Notes

1.What is an array?

An array is a collection of elements of the same data type, stored in contiguous memory locations, which can be accessed using an index or a subscript.

2.Declaring an array

to declare an array in Java, you need to specify the data type of the array elements, followed by the name of the array and the size of the array in square brackets. For example, to declare an array of integers with a size of 5, you can write:

```
int[] myArray = new int[5];
```

3.Initializing an array

You can initialize an array at the time of declaration or later. To initialize an array at the time of declaration, you can enclose the values in curly braces and separate them with commas. For example, to initialize an array of integers with the values 1, 2, 3, 4, and 5, you can write:

```
int[] myArray = {1, 2, 3, 4, 5};
```

4.Accessing array elements

You can access array elements using the index or subscript. The index of the first element is 0, and the index of the last element is size - 1. For example, to access the first element of an array, you can write:

```
int firstElement = myArray[0];
```

5.Modifying array

elements You can modify the value of an array element using the index or subscript. For example, to change the value of the first element of an array to 10, you can write:

```
myArray[0] = 10;
```

6.Array length

You can get the length of an array using the length property. For example, to get the length of an array, you can write:

```
int arrayLength = myArray.length;
```

7.Iterating over an array

You can iterate over an array using a for loop. For example, to print all the elements of an array, you can write:

```
for (int i = 0; i < myArray.length; i++)
{
        System.out.println(myArray[i]);
}
```

8.Multidimensional arrays

Java supports multidimensional arrays, which are arrays of arrays. For example, to declare a two-dimensional array of integers with a size of 3x3, you can write:

cssCopy code

```
int[][] myArray = new int[3][3];
```

To access an element of a two-dimensional array, you need to provide two indices or subscripts. For example, to access the element at row 1 and column 2 of a two-dimensional array, you can write:

```
int element = myArray[1][2];
```

9.Declaration

To declare a two-dimensional array in Java, you need to specify the number of rows and columns in the array. For example:

cssCopy code

```
int[][] myArray = new int[3][4];
```

This creates a two-dimensional array with 3 rows and 4 columns.

10.Initialization

You can initialize a two-dimensional array at the time of declaration or later using nested loops. For example:

cssCopy code

```
int[][] myArray = {{1,2,3}, {4,5,6}, {7,8,9}};
```

This creates a two-dimensional array with 3 rows and 3 columns and initializes the elements with the specified values.

11.Accessing elements: You can access elements in a two-dimensional array using the row and column indices. For example:

cssCopy code

```
int element = myArray[1][2];
```

This accesses the element in the second row and third column of the array.

12.Looping through elements: You can use nested loops to iterate through all the elements in a two-dimensional array. For example:

```
for (int i = 0; i < myArray.length; i++)
{
for (int j = 0; j < myArray[i].length; j++)
{
        System.out.print(myArray[i][j] + " ");

          }

    System.out.println();

}
```

Assignment Questions

1.Find the sum of all elements in an array:

2.Find the largest element in an array:

3.Find the average of all elements in an array:

4.Copy one array to another array:

5.Write a program to print two dimensional array in below format

1 | 2 | 3

- - - - -

4 | 5 | 6

- - - - -

7 | 8 | 9

Example java program:

```
public class ArraysTest {
    //What is an array
    //Array is consecutive memory location

    public static void main(String[] args) {
        //1 dimensional
        int i[];
        int[] j = new int[] {1, 2, 3, 4, 5};



        //2 dimensional
        int i_2d[][] = new int[10][5];
        int[][] j_2d = {
                        {1, 2, 3, 4, 5},
                        {6, 7, 8, 9, 10}
                    };

        //3 dimensional
        int i_3d[][][] = new int[10][5][4]
        int[][][] j_3d = {
                            {
                               {1, 2, 3, 4, 5} ,
                               {6, 7, 8, 9, 10}
                            },
                            {
                               {1, 2, 3, 4, 5},
                               {6, 7, 8, 9, 10}
                            }
                    };
```



```
        //To access an array, add index number in [] after variable
        j_2d[1][1] = 10;

        //printing element of an array
        System.out.println(j[2]);
        System.out.println(j_2d[1][3]);
        System.out.println(j_3d[1][0][3]);


    }


}
```

Array operation code (example in assignment questin) :

```
public class ArrayOperation {
    public static void main(String[] args) {
        int[] oneDimensionalArray = {10, 100, 33, 44, 1};
        int len = oneDimensionalArray.length;
        //Find minimum number in an array
        if(oneDimensionalArray.length == 0){
            System.out.println("Array is empty");
        }else {

            //Find maximum element in an array
            int minNumber = oneDimensionalArray[0];

            for (int i = 0; i < len; i++) {
                if (minNumber > oneDimensionalArray[i]) {
                    minNumber = oneDimensionalArray[i];
                }
```



```
            }
            System.out.println("Min number is " + minNumber);
        }
    }
}
```