

# Lab-1 Java as Structured programming

---

## Objectives

Compiling and executing Java exercises using loop statements and arrays

## Theory

Making an array in a Java program involves three distinct steps:

1. Declare the array name.
2. Create the array.
3. Initialize the array values.

We refer to an array element by putting its index in square brackets after the array name: the code `a[i]` refers to element `i` of array `a[]`. For example, the following code makes an array of `n` numbers of type `double`, all initialized to 0:

```
double[] a;           // declare the array
a = new double[n];    // create the array
for (int i = 0; i < n; i++) // elements are indexed from 0 to n-1
    a[i] = 0.0;       // initialize all elements to 0.0
```

Run the following example for more details. Once complete, then continue to next part of the lab

```
public class ArrayEjemplo {
    public static void main(String[] args) {
        int n = 7;

        // initialize to random values between 0 and 1
        float[] a = new float[n];
        for (int i = 0; i < n; i++) {
            a[i] = 2*i+3;
        }

        // print array values, one per line
        System.out.println("a[] = ");

        for (int i = 0; i < n; i++) {
            System.out.println(a[i]);
        }
        System.out.println();
        System.out.println();

        // find the maximum
        double max = a[0];
        for (int i = 0; i < n; i++) {
            if (a[i] > max) max = a[i];
        }
    }
}
```

```

        System.out.println("max = " + max);

        // average
        double sum = 0.0;
        for (int i = 0; i < n; i++) {
            sum += a[i];
        }
        System.out.println("average = " + sum / n);

        // copy to another array
        double[] b = new double[n];
        for (int i = 0; i < n; i++) {
            b[i] = a[i];
        }

        // dot product of a[] and b[]
        double dotProduct = 0.0;
        for (int i = 0; i < n; i++) {
            dotProduct += a[i] * b[i];
        }
        System.out.println("dot product of a[] and b[] = " + dotProduct);
    }
}

```

## Exercises

This Section is intended to start in lab session, but you need to complete out the class and deliver the next lab-session.

1. Write an application that inputs three integers from the user and displays the sum, average, product, smallest and largest of the numbers. [Note: The calculation of the average in this exercise should result in an integer representation of the average. So if the sum of the values is 7, the average should be 2, not 2.3333....]

Sample output

```

Enter first integer: 10
Enter second integer: 20
Enter third integer: 30

For the numbers 10, 20 and 30
Largest is 30
Smallest is 10
Sum is 60
Product is 6000
Average is 20

```

2. Write an application that finds two neighboring numbers in an array with the smallest distance to each other. The program should return the index of the first number.

For example given the sequence of numbers in array  $A = \{4\ 8\ 6\ 1\ 2\ 9\ 4\}$  the minimum distance is 1 (between 1 and 2). The program should return the index 3 (of number 1). Hint: Use `Math.abs()` to calculate the distance of two numbers.

3. Create a Java program with the following characteristics:
- a) Prompts for and reads in the sales for each of 5 vendor in a company. It then prints out the *id* and amount of sales for each vendor and the total sales.
  - b) Compute and print the average sale.
  - c) Find and print the maximum sale. Print both the id of the vendor with the min sale and the amount of the sale, e.g., "*Vendor 3 had the lowest sale with \$4500.*"
  - d) Do the same with the minimum value.
  - e) After the list, sum, average, max, and min have been printed, ask the user to enter a value. Then print the *id* of each vendor who exceeded that amount, and the amount of their sales. Also print the total number of vendor whose sales exceeded the value entered.
  - f) Modify your program so that the *ids* run from 1–5 instead of 0–4.
  - g) Instead of always reading in 5 sales amounts, at the beginning ask the user for the number of sales people and then create an array that is just the right size.
4. Write a program that grades arithmetic quizzes as follows:
- a) Ask the user how many questions are in the quiz.
  - b) Ask the user to enter the key (that is, the correct answers). There should be one answer for each question in the quiz, and each answer should be an integer. For example the key for a 10-question quiz, can be (single line): 34 7 13 100 81 3 9 10 321 12. Hint, store the key in an array.
  - c) Ask the user to enter the answers for the quiz to be graded. As for the key, these can be entered on a single line. Note that these answers do not need to be stored; each answer can simply be compared to the key as it is entered.
  - d) When the user has entered all of the answers to be graded, print the number correct and the percent correct.