

1 Objectives

- Write, compile and run a simple Java application using the terminal
- Write static functions using primitive and String types
- Write static functions using arrays

Software installation

- Java Development Kit (JDK):
<https://www.oracle.com/java/technologies/javase-jdk15-downloads.html>
- IntelliJ IDEA Community edition: <https://www.jetbrains.com/idea/download/>

Online Java environment:

https://www.tutorialspoint.com/compile_java_online.php

I. Simple programs - main function

1. Write a program that prints your name and country of origin.

HINT: `System.out.println`

2. Write a program that prints each character of your name in a new line.

```
String name = "Margit Antal";
```

```
name.charAt(0) -> 'M'
```

```
name.length() -> 12
```

3. Write a program that splits a name into parts and prints each part on a new line.

For example: name: "Joe Donald Biden", output:

```
Joe  
Donald  
Biden
```

HINT: Use the `split` function.

4. Write a program that prints the characters of a word in a word pyramid:

Input: TREE

Output:

T

TR

TRE

TREE

Help:

```
String name = "apple";  
name.substring(0,2) -> "ap"
```

II. Static Functions

1. Consider the following static function.

```
public static double maxElement(double array[]){  
    double max = Double.NEGATIVE_INFINITY;  
    for(int i=0;i<array.length;++i){  
        if(array[i] > max){  
            max = array[i];  
        }  
    }  
    return max;  
}
```

Write some test code for the `maxElement` function. For example:

```
double x[] ={ 7, 1, -3, 45, 9};  
System.out.printf("MAX: %.2f\n", maxElement( x ));
```

2. Write a function that returns the value of a given bit in a number. The bits are numbered from right to left, starting from 0. Use bit operations (loops are forbidden!).

```
public static byte getBit(int number, int order)
```

Let $n = 13$
Binary representation: `00001101`
`getBit(n, 0) → 1`, `getBit(n, 1) → 0`, `getBit(n, 2) → 1`, ...
In case of non-existent order return `-1`.

3. Write a function that counts the number of '1' bits. Use bit operations.

```
public static int countBits(int number)
```

4. Write a function that computes the mean of an array. In case of an empty array, return `Double.NaN`. Test your function!

```
public static double mean(double array[])
```

5. Write a function that computes the standard deviation of an array of real numbers. In case of an empty array returns `Double.NaN`. Test your function!

Standard deviation:

<https://www.khanacademy.org/math/probability/data-distributions-a1/summarizing-spread-distributions/a/calculating-standard-deviation-step-by-step>

6. Compute the two largest elements of an array using a single loop.

```
public static double[] max2(double array[]){
    double max[] = {Double.NEGATIVE_INFINITY,
                    Double.NEGATIVE_INFINITY};
    if( array.length == 0 ){
        return max;
    }
    //...
}
```

HINT:

```
max2({100}) → {100, 100}
max2({}) → {Double.NEGATIVE_INFINITY, Double.NEGATIVE_INFINITY}
max2({1, 2, 3, 200, -7, 78, 9, 42}) → {78, 200}
```

7. Create a function that populates an array of 1000000 integers with random values. Sort the array (use `Arrays.sort`), then prints the number of duplicates.