

1st Exam Extra Credit

Instructions:

Work on all the questions below related to Java Method creation. Test your program either using NetBeans or the editor provided in the link (Go to the editor). After you got your code correctly implemented, paste it in a Word document-like app along with its question. Print this document with your name and give it to me on the next lecture (either February 17th or 20th).

Avoid searching for their answers on the internet. By doing so you will not learn anything. You can discuss with friends, but be sure to understand why you are implementing the solution in that way. Be prepared to answer question(s) that I might ask you related to this topic.

A - Questions about Methods

1. Write a Java method (called `getSmallerNumber`) to find the smallest number among three numbers. [Go to the editor](#)

Test Data:

```
Input the first number: 25
Input the Second number: 37
Input the third number: 29
```

Expected Output:

```
The smallest value is 25.0
```

2. Write a Java method to compute the average of three numbers. [Go to the editor](#)

Test Data:

```
Input the first number: 25
Input the second number: 45
Input the third number: 65
```

Expected Output:

```
The average value is 45.0
```

3. Write a Java method to display the middle character of a string. [Go to the editor](#)

Note: a) If the length of the string is odd there will be two middle characters.

b) If the length of the string is even there will be one middle character.

Test Data:

```
Input a string: 350
```

Expected Output:

```
The middle character in the string: 5
```

4. Write a Java method to count all vowels in a string. [Go to the editor](#)

Test Data:

```
Input the string: w3resource
```

Expected Output:

```
Number of Vowels in the string: 4
```

5. Write a Java method to count all words in a string. [Go to the editor](#)

Test Data:

```
Input the string: The quick brown fox jumps over the  
lazy dog.
```

Expected Output:

```
Number of words in the string: 9
```

6. Write a Java method to compute the sum of the digits in an integer. [Go to the editor](#)

Test Data:

```
Input an integer: 25
```

Expected Output:

```
The sum is 7
```

7. Write a Java method to display the first X prime numbers, where X is a number that must be provided as input argument for your method. [Go to the editor](#)

Note: A prime number is a number that is only divisible by itself, e.g., 7 is prime number because $7\%7$ is 0 and 7% (either 6, 5, ..., 2 is not 0, meaning that there is a remaining on the integer division). Example of a non prime number: 6 because $6\%3$ and $6\%2$ returns 0. Hint use a WHILE loop, a counter (to count up to X) to keep going until you get X prime numbers, and a variable number to increment it as much as needed while getting the X prime numbers.

Test Data:

```
Input number of prime numbers to be generated: 20
```

Expected Output:

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,  
53, 59, 61, 67, 71
```

8. Write a Java method to check whether a year (integer) entered by the user is a leap year or not. [Go to the editor](#)

Leap Year Rules: How to Calculate Leap Years

1. The **year** must be evenly divisible by 4;
2. If the **year** can be evenly divided by 100, it is not a **leap year**; unless,
3. The **year** is also evenly divisible by 400. Then it is a **leap year**.

Test Data:

```
Input a year: 2020
```

Expected Output:

```
Input a year: 2020  
true
```

B - Questions about Arrays

1. Write a Java program to sum values of an array. [Go to the editor](#)
2. Write a Java program to calculate the average, the maximum and minimum values of array elements. [Go to the editor](#)
3. Write a Java program to test if an array contains a specific value. [Go to the editor](#)
4. Write a Java program to find the index of an array element. [Go to the editor](#)
5. Write a Java program to arrange the elements of a given array of integers where all negative integers appear before all the positive integers. [Go to the editor](#)