Technical University of Munich
School of Engineering and Design
Prof. Dr. Martin Werner

Computational Foundations I (Winter Term 2021/22)
Tutorial 1

Tasks marked with a star like **Optional Task*** are optional. Tasks marked like **Hard Task**+ are given, but it is not expected that you solve them now. It is great if you learn to solve them during the lecture. Go back to them after a few weeks and see your own progress.

Learning Outcome: You learn how to use MATLAB, you learn to know basic loops and how to write functions with and without arguments. You accomodate yourself with variables, global variables, and state updates through the turtle graphics mechanism.

Task 1: Basics in MATLAB

In this task, we want to practice certain elements of the MATLAB language which you will find helpful when trying to write your first programs. This includes
- Types and variables - Casting - Matrices
Recall all numeric data types of the MATLAB language as can be found in the official MATLAB manual (`https://de.mathworks.com/help/matlab/numeric-types.html`). Then try to answer the following questions and complete the following program snippets.

a. **Some Casting:** Look up casting here `https://www.mathworks.com/help/matlab/ref/cast.html`. Then cast the number given by π multiplied with 100 to int32, int8 and uint8. What is happening when you cast to int8 and uint8?

b. **Some Rounding:** Implement a small script in which one variable is set to 2.4 and another variable is set to 42. Compute the sum of both and round it up and down to the nearest integer. Store the result in variables called `sum_rounded_up` and `sum_rounded_down`. For that, you can either use typecasting between floating point and integer numbers which typically truncates numbers or you can have a look at dedicated functions such as `ceil`, `floor`, `round`. See as well `https://de.mathworks.com/help/matlab/ref/round.html`.

c. **Identity Matrix:** Create the following matrix. Do this by hand or using the the eye function (https://de.mathworks.com/help/matlab/ref/eye.html).

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Optional: When you come back to this task after learning loops, try to write a custom eye function yourself.