

## Dojo-Style SE\_01 Assessment: 17 August 2020

The goal of these challenges is for you to demonstrate your ability to develop software at a basic level, not merely to turn in solutions to problems. The module description reads:

*Software Development Basics is about basic programming competencies and the ability to develop a piece of software, starting from the very beginning. In doing this, students learn how to develop software using a programming language of their choice. In addition to the basics of programming itself, this module begins to go into the process of software development: How do I start a software project? How do I use a command line interface as a developer? How do I leverage version control? How do I document my code, and create a piece of software which other people can access and execute? Furthermore, students begin to practice the standard techniques of software development including iterative development, documentation, and debugging.*

Thus, in order to demonstrate your ability to develop software at a basic level, you need to write code, debug your code, and iterate on your code. The assessor will be keeping an eye out for this throughout the session. Furthermore, the code should be in a state that it can be read and be run by someone else at the end of the session; this means that it also needs to have instructions (like a readme file that explains how to run the code) and be understandable. The exercises themselves are a means to demonstrate this process. Note that there will be several exercises. It is not mandatory to finish all of them, but it is expected that you can at least complete the first exercise.

## Remote Assessment Guidelines

This is a remote written assessment. The maximum total time is 240 minutes (4hrs).

Please feel free to take short breaks (5-10min) (hit the “coffee mug” symbol in the zoom participants list) at your leisure.

If you have a question please use the “raise hand” feature at any time, I will then invite you to a breakout room to not interrupt the other attendees. If you have a short question feel free to ask in (private) zoom chat.

Please keep an Eye out for the chat, I will give updates about the time and clarifications there.

If you are done with an Exercise please send me the solution as a zip file (via Slack) or as a github link (my username is “codeFabio”). Then raise your hand in zoom (give me a moment to look through your solution ... now is a good time for a break), I will then give you the next exercise.

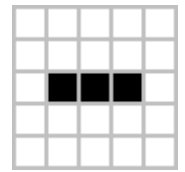
- You are allowed to use the internet to look at your language/framework documentation.
- You are allowed to ask me for help at any time.
- **You are NOT allowed to copy solutions or partial solutions.**
- **You are NOT allowed to ask other persons for help during the assessment**

(I will ask everyone of you individually and personally to promise to adhere to this rules).

In case your internet connection breaks down, first try to reconnect. If it keeps being broken for more than 5 minutes please call me under +49 172 9728955 we will then discuss how to proceed.

# Game of Life

Conway's Game of Life is what happens when Mathematicians create a solitaire game that accidentally turns out to have interesting scientific properties. While it was devised as a solitaire style game it is much more interesting to observe it as a simulation.



The world in Conway's game is a grid of cells. Each cell is either empty or has an inhabitant. Now the simulation starts: At each cell we count how many inhabitants live in neighbouring cells. Depending on how many neighbours a cell has the inhabitant survives, dies or a new inhabitant is born. This step (also called iteration) is then repeated. Depending on the initial configuration of the world different things can happen: sometimes a stable state occurs, after which the world does not change any longer, sometimes the world oscillates between different states.

## Exercise 1: Around the block

We will now focus on the neighbourhoods. A neighbour is a cell that is adjacent to a location, either horizontally, vertically or diagonally.

### Instructions:

Write a piece of software that creates a “population density heatmap”, that shows how many neighbours each cell has. You are given an initial representation of the world, where a 0 represents an empty cell and 1 represents an inhabitant.

Input:		Output (heatmap):
0 1 0		2 1 2
0 1 0	----->	3 2 3
0 1 0		2 1 2

### Hints:

- You need to be careful at the borders of the world.
- You are not your own neighbour