Essay – Lab 1 – Florin-George Bărcan

This lesson introduced basic notions regarding Solidity and Blockchain, like *contracts*.

*Contracts* are the building blocks of Solidity since it is where the code is encapsulated. Inside of these structures you can declare different data types, functions and events. Solidity seems like a combination of JavaScript, Python3 and a pinch of C.

You are introduced to *structs* quite early since a lot of operations depend on these data types which tend to become very complex. In this tutorial you are introduced to the `Zombie` data structure which contains a string and an unsigned integer.

Functions are quite malleable. Parameters for functions are passed both ways: by value and by reference. In case of passing by value, a copy is created of that variable and passed as argument without worrying that the value might be modified, while in the other case, the function is called with a reference to the variable, i.e., the variable will be modified. You can insert different modifiers such as `view`, or `pure`, or the most common: `public` and `private`.

Solidity features typecasting which is very helpful in certain situations like in the tutorial where you need to change a 256-bit hexadecimal number into an unsigned integer.

Events are ways of communication for our contract to emit that something somewhere in the code changed to the front-end application.

All in all, the tutorial was prompt. With more practice, instead of zombies, different node-points could be created to mask our location from big brothers but this would require a lot of processing power and with that, a lot of money.