Bitcoin for developers – I

|  |  |
| --- | --- |
| Required lessons | Possible assignments and exams |
| 1. General discussion 2. Bits and bytes 3. Building environment (droplet? Raspberry pi?) 4. One functions + hashing 5. Byzantine general problem 6. Messages Version fields (not header yet) 7. Messages Version with header 8. Connect to another peer + sockets 9. Sending messages (wireshark, bitpy) 10. Receiving message (version, verack and ping) 11. Transactions - blockchain | * 1. Bits, bytes and encoding   2. Hashing   3. Messaging vanilla protocol in class |

Bitcoin for developers – II

|  |  |
| --- | --- |
| Required lessons | Possible assignments and exams |
| 1. Transactions - blockchain misconceptions 2. Mathematical trap door and keys 3. Creating key pair. From Private key to bitcoin address 4. Transactions creating transaction (no scripts yet) 5. Stack architecture and scripting language 6. Bitcoin sigscript | * 1. Creating keypair   2. Creating raw transaction   3. Protocol rules (written exam?)   4. Bitcoin scripting language |

Bitcoin for developers – III

|  |  |
| --- | --- |
| Required lessons | Possible assignments and exams |
| 1. The blockchain - looking at individual block. 2. The blockchain - consensus rules (and DAG? TBD) 3. The blockchain - Coinbase transactions 4. The blockchain - merkle trees, light clients. 5. The blockchain - proof of work, calculating the difficulty/target, block headers and the nonce. 6. Raspberry pi private blockchain 7. Raspberry pi bitcoin node 8. Optional open discussions - Segwit, lightning network, merged mining and side chains. | 1. Creating merkle trees/roots 2. Creating mock miners/ hashing 3. Deploying private blockchains 4. Calculating difficulty and target 5. Make a speech about future improvements? |