Andreas Tsantes
Dipl. Ing. (FH) M.Sc.
Lecturer
Blockchain

CONTENTS

1	General instructions	4
2	Foreword	5
3	Introduction	6
3.1	Part 00: Technical onboarding	7
3.2	Part 01: Hash functions and cryptography	8
3.3	Part 02: Full blockchain with DB	9
3.4	Part 03: Proof-of-Work	10
3.5	Part 04: Interface	11
3.6	Part 05: Full node	12
3.7	Part 06: Wallet	13
3.8	Part 07: Peer-to-peer	14

GENERAL INSTRUCTIONS

Welcome to the course "Blockchain" at the University of Applied Sciences Würzburg-Schweinfurt.

Starting from the ground, we will develop a full blockchain system in Python, including a peer-to-peer network, a proof-of-work consensus algorithm, and a wallet for sending and receiving transactions.

Each week, you will receive a new part of the blockchain. Your task is to implement the assigned part and integrate it into your existing blockchain.

Every week, we will discuss the lecture and the exercises in the lecture hall.

Please read the instructions carefully. Ask questions if something is unclear.

Use the forum on Moodle for questions and discussions. Do not share your solutions there.

Use a version control system (Git) to track your changes.

Write clean, readable code. Quality matters.

FOREWORD

WHY A CHAIN OF HASHES CHANGES EVERYTHING

Imagine a world where everyone keeps their own version of the truth. Banks maintain their ledgers, governments track their records, companies manage their databases. To prove something, you need someone you trust to vouch for the data. This works well until someone wants to cheat, or one database gets hacked, or two parties disagree about what happened.

Blockchain solves this problem by making it impossible to secretly change past data. Each block contains a hash of the previous block, and each hash depends on all the data in that block. If someone changes anything in the past, the hashes break and everyone sees it immediately.

This course teaches you how this works by building it yourself. You will implement hashing, mining, wallets, and peer-to-peer networking. By the end, you will understand why blockchain technology matters and how it actually works.

INTRODUCTION

WHAT YOU WILL BUILD BEFORE YOU DIE

WHY WE'RE DOING THIS

Blockchain is a distributed ledger. Instead of one database controlled by one company, many computers keep copies of the same data. To add new data, they must agree on it. To change old data, someone must convince everyone that the change is valid.

This course teaches you how blockchain works by building it from scratch. You will implement hashing, mining, wallets, and peer-to-peer networking. By the end, you will understand why blockchain matters and how it actually works.

WHAT YOU'LL BUILD

Each week, you will add a new part to your blockchain. The parts build on each other:

- Part 00: Technical onboarding
- Part 01: Hash functions and cryptography
- Part 02: Full blockchain with DB
- Part 03: Proof-of-Work
- Part 04: Interface
- Part 05: Full node
- Part 06: Wallet
- Part 07: Peer-to-peer

HOW WE'LL WORK

Every week, we will discuss the lecture and the exercises in the lecture hall. Please read the instructions before the lecture. Ask questions if something is unclear.

Use the forum on Moodle for questions and discussions. Do not share your solutions there.

Use a version control system (Git) to track your changes.

Write clean, readable code. Quality matters.

PART 00: Technical onboarding

WHAT WE'LL DO TODAY

Today we will set up the tools we need for the course. We will create a GitHub account, install Python and Visual Studio Code, and write our first program.

GITHUB

GitHub is a website for storing and sharing code. It uses Git, a version control system, to track changes to files. You can see who changed what, when, and why. You can also work on different versions of the code at the same time and merge them later.

PYTHON

Python is a programming language. We will use it for this course because it is easy to read and write. Python 3.12 or higher is required.

VISUAL STUDIO CODE

Visual Studio Code is a text editor. It helps you write code with syntax highlighting, auto-completion, and error checking. We will use it for all assignments.

TASK

1. Create a GitHub account if you do not have one.
2. Install Python 3.12 or higher.
3. Install Visual Studio Code.
4. Write a program that prints "Hello, World!".
5. Run your program from the command line.

Part 01: Hash functions and cryptography

WHAT YOU'LL DO TODAY

Today we will learn how hashes work and why they are important for blockchain. We will write a function that calculates hashes and a function that checks if a hash is valid.

HASHES

A hash function takes any data and produces a fixed-length output. The same input always gives the same output. Even a small change in the input gives a completely different output. This makes it easy to detect if data has been changed.

SHA-256

We will use SHA-256 to hash our blocks. It produces a 256-bit output, which is 64 hexadecimal characters. Bitcoin uses SHA-256 twice, but we will use it once.

TASK

1. Write a function that calculates the SHA-256 hash of a block.
2. Write a function that checks if a hash is valid.
3. Write a function that prints a block in a readable format.
4. Test your functions with different inputs.

Part 02: Full blockchain with DB

WHAT YOU'LL DO TODAY

Today we will add a database to store the blocks and build a full blockchain. We will write functions to add blocks, check the chain, and find the longest valid chain.

DATABASE

We will use SQLite to store the blocks. SQLite is a lightweight database that stores data in a single file. It is easy to use and does not require a server.

BLOCKCHAIN

A blockchain is a chain of blocks. Each block contains a hash of the previous block. If someone changes a block in the past, the hashes break and everyone sees it immediately.

TASK

1. Create a database to store the blocks.
2. Write a function that adds a block to the database.
3. Write a function that checks if the chain is valid.
4. Write a function that finds the longest valid chain.
5. Test your functions with different inputs.

Part 03: Proof-of-Work

WHAT YOU'LL DO TODAY

Today we will add mining to our blockchain. We will write a function that finds a hash with a specified number of leading zeros. We will also write a function that checks if a hash is valid.

MINING

Mining is the process of finding a hash that meets the difficulty requirement. The difficulty is the number of leading zeros in the hash. The more zeros, the harder it is to find a valid hash.

TASK

1. Write a function that finds a hash with a specified number of leading zeros.
2. Write a function that checks if a hash is valid.
3. Test your functions with different inputs.

Part 04: Interface

WHAT YOU'LL DO TODAY

Today we will add a web interface to our blockchain. We will write a function that starts a web server and a function that handles requests. We will also write a function that renders the blockchain in a readable format.

WEB INTERFACE

We will use Flask to write a web interface. Flask is a lightweight framework that makes it easy to write web applications. We will use it to display the blockchain and to add new blocks.

TASK

1. Write a function that starts a web server.
2. Write a function that handles requests.
3. Write a function that renders the blockchain in a readable format.
4. Test your functions with different inputs.

Part 05: Full node

WHAT YOU'LL DO TODAY

Today we will add a full node to our blockchain. We will write a function that validates blocks and a function that stores the blockchain. We will also write a function that broadcasts new blocks to other nodes.

FULL NODE

A full node is a node that stores the blockchain and validates blocks. It does not mine blocks. It only checks if blocks are valid and stores them.

TASK

1. Write a function that validates blocks.
2. Write a function that stores the blockchain.
3. Write a function that broadcasts new blocks to other nodes.
4. Test your functions with different inputs.

Part 06: Wallet

WHAT YOU'LL DO TODAY

Today we will add a wallet to our blockchain. We will write a function that creates a wallet and a function that signs transactions. We will also write a function that verifies signatures.

WALLET

A wallet is a pair of keys. The private key is used to sign transactions. The public key is used to verify signatures. The address is a hash of the public key.

TASK

1. Write a function that creates a wallet.
2. Write a function that signs transactions.
3. Write a function that verifies signatures.
4. Test your functions with different inputs.

Part 07: Peer-to-peer

WHAT YOU'LL DO TODAY

Today we will add a peer-to-peer network to our blockchain. We will write a function that connects to other nodes and a function that handles messages. We will also write a function that synchronizes the blockchain.

PEER-TO-PEER

A peer-to-peer network is a network of nodes that connect to each other. Each node stores the blockchain and validates blocks. There is no central server.

TASK

1. Write a function that connects to other nodes.
2. Write a function that handles messages.
3. Write a function that synchronizes the blockchain.
4. Test your functions with different inputs.