Syllabus

Writing Smart Contracts v 2.0

Peter H. Gruber*

Course Description

Blockchain technology solves the important problem of automating transactions between agents in an efficient, flexible and trustworthy manner. It is one of the pillars of the digital finance revolution. New and specific skills are needed to fuel this revolution: Smart contracts require a distinct way of thinking and a novel programming approach, different from standard procedural programming.

This course introduces students to the relevant theoretical concepts in economics and cryptography and provides a hands-on introduction to writing smart contracts on a modern blockchain. To do so, the course uses only one programming language – Python – for all programming examples: be it sending transactions, writing smart contracts or analyzing blockchain data. This makes it possible to introduce students to the field in a coherent and concise way. The goal is to equip students with the necessary skills and useful background knowledge for creating a token, an NFT or to develop and deploy simple smart contracts. The Algorand blockchain has been chosen for all practical examples, as it unites speed and low cost with sustainability and easy-to-use programming tools.

Target Group and Prerequisites

The course is aimed at master or advanced bachelor students in economics, finance and law, as well as partitioners from the legal and finance industries. Participants should have basic programming skills in Python and a keen interest in the topic. (Students with no Python experience can alternatively complete online tutorials before the beginning of the winter school.) Basic knowledge of statistics and financial contracts is useful, but not mandatory. No experience with the blockchain is required. While many examples will be from finance and economics, the course is sufficiently general for students from other fields to profit from it.

Course Objectives

After this course, students will be able to interact programmatically with the Algorand blockchain as well as to create their own tokens and NFTs. They will have learnt to express simple governance

*Contact: peter.gruber@usi.ch

structures, business processes and financial assets as smart contracts. They will be able to implement these smart contracts in Python and to deploy them on the Algorand blockchain. They will understand the differences between functional and contract-oriented programming, as well as the risks specific to smart contract programming. They will be able to access and analyze blockchain data. Beyond the technical skills, they will have learnt the following soft skills: understanding the legal implications of smart contracts, organizing the own work on smart contracts, navigating the market for programming talent and starting a blockchain business.

Methods, Course Work

Classes are based on realistic applications and take students from theory to mastery in five steps:

1. Lecture with short presentation of a new concept
2. Guided tour: students and lecturer work together on applying the new concept
3. Short rationalization of the lessons learned
4. Individual exercises with possibility to ask questions during the exercise session
5. Discussion of students' solutions to exercises in class

Additional group work between students, guest lectures and discussion sessions are aimed at fostering creative application of the skills learned.

Class material and literature

Literature, slides, sample programs in Jupyter notebooks, solutions to exercises and class videos are made available digitally. Literature consists of original papers and original documentation.

Grading and crediting

Grading is based on individual homeworks (40%) and a project after the end of the course (60%), that is completed individually or in small groups.

Homeworks are simple applications of what had been learnt in class. The exercises are handed in via the blockchain.

Projects consist of (a) writing and deploying a smart contract for a real-world problem and (b) a five page PDF documentation.

Coursework and structure (28 units of 45 minutes each plus homeworks and project work) as well as the exam are commensurate with a 3 ECTS course. Crediting of the course at USI is ensured as a 3 ECTS elective course for the faculty of Economics and several master programs of the faculty of Informatics. Participants from other universities must ensure crediting themselves.

Organisation

The course will be held in English. Students will have to bring:

- A laptop with the Anaconda environment installed
- A mobile phone

All necessary software is free and open source. Additional software will be installed at the beginning of the course.

Contents

1. Introduction to blockchain theory
    - Distributed ledgers, consensus algorithms, transactions
    - Basic cryptography: hashing, signing, encrypting
    - Overview and classification of major blockchains
    - The economics of digital money and digital goods
2. Tools
    - Python and Anaconda
    - Setting up a wallet and making a transfer
    - Blockchain indexers and APIs
3. Wallets
    - Accessing the blockchain with Python
    - Creating and managing credentials
    - Understanding blockchain addresses
    - Standard blockchain transactions
4. Tokens
    - Creating and managing a fungible token
    - Simple transactions with tokens
    - Grouped transactions and atomic swaps
5. NFTs
    - The NFT economy
    - Creating an NFT
6. Stateless Smart Contracts (Smart Signatures)
    - Python for smart contracts: PyTEAL
    - Deploy, fund and interact with smart signatures
    - Integer arithmetic on the blockchain
    - Attacks on smart contracts
7. Stateful Smart Contracts
    - Structure and lifetime of a smart contract
    - Write, deploy and interact with a “hello world” smart conract
    - Global and local states
    - Arguments
    - Interacting with accounts and payment transactions
8. Examples for Smart Contracts
    - Social Security
    - American Options
    - Vending
9. Oracles
10. Coding principles and debugging
11. Blockchain empirics and blockchain economics
    - Extract, manage and visualize blockchain data
    - Major variables in tokenomics
    - Decentralized Organizations and Governance
12. Projects
13. Soft skills
    - Team work

- The market for programming talent
- Launching a blockchain business
- Social and environmental dimensions of blockchain technology

14. Legal aspects of DLT technology
15. Security

About

Università della Svizzera italiana (USI) is the public university of the canton of Ticino in southern Switzerland. Established in 1996, it currently has about 3'350 students. USI is ranked among the top 250 universities world-wide, according to the THE ranking. Its master program in Finance ranks number 37 world-wide in the Business School Ranking of the Financial Times. USI's main campus is located at Lugano, home of the famous plan B.

The Algorand blockchain is a latest-generation, flexible and energy-efficient blockchain. Since 2021, Algorand has been carbon-neutral. It allows programming a vast range of smart contacts in a developer-friendly way and has very low transaction costs. Algorand is based on the work of Silvio Micali (MIT), winner of the Turing prize, dubbed as the "Nobel prize in informatics".

Peter H. Gruber has PhDs in financial economics and particle physics. He is Senior Lecturer at the Università della Svizzera italiana (USI) in Lugano. He has been teaching programming to economics and finance students at USI, St. Gallen, Geneva and Bocconi since 2005. Before this, he was part of the neutrino physics efforts at CERN. His current research interests are asset pricing, blockchain economics, data science with non-traditional data sets, and data visualization. He represents USI in the Algorand Global University program, of which USI is a founding member.