

# **Syllabus**

## **Writing Smart Contracts**

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 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, analyzing blockchain data or writing smart contracts. This makes it possible to introduce students to the field in a concise and consistent way. The goal is to equip students with the necessary skills and useful background knowledge for creating simple smart contracts and for deploying them on the Algorand blockchain.

### **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.

### **Target Group and Prerequisites**

The course is aimed at master or advanced bachelor students in economics or finance. Students 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 begin of the winter school.) No experience in 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 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 about organizing their work on smart contracts, about the market for programming talent and how to start 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

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

## Grading

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.

## 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.

## Contents

1. Introduction to blockchain theory
  - Distributed ledgers, consensus algorithms, transactions
  - Basic cryptography: hashing, signing, encrypting
  - Overview and classification of major blockchains
2. Tools
  - Python and Anaconda
  - Setting up a wallet and making a transfer
  - Blockchain indexers and APIs
3. Wallets
  - Accessing the blockchain with Python
  - Understanding blockchain addresses
  - Creating and managing credentials
  - 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 contract
  - 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
12. Projects
13. Soft skills
  - Team work
  - The market for talent
  - Launching a blockchain business
  - Social and environmental dimensions of blockchain technology
14. Security

## About

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 contracts 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, often 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 University program, of which USI is a founding member.