UNIVERSITY of NICOSIA

Course Syllabus

Course Code	Course Title	ECTS Credits
BLOC-522	Smart Contracts Programming	10
Prerequisites	**Department**	**Semester**
BLOC-512 BLOC-514	Digital Innovation	Fall/Spring
Type of Course	**Field**	**Language of Instruction**
Elective	Smart Contracts, Programming	English
Level of Course	**Lecturer(s)**	**Year of Study**
2nd Cycle	Dr. Klitos Christodoulou	2nd
Mode of Delivery	**Work Placement**	**Corequisites**
Face to face	N/A	N/A

Course Objectives:

This course is designed for developers that have familiarity with other high-level programming languages. The main element of this course is to provide students with a solid understanding of the many opportunities for building decentralized applications using the Web3 stack and the Turing-complete Solidity language over the Ethereum Virtual Machine (EVM).

Thus, the course is mainly structured over the following three pillars:

The Evolution of the Web Infrastructure by decomposing and understanding the Decentralized Web 3.0 Technology Stack. This part will focus on the vision of realizing Web 3.0 and the promise of eliminating intermediates and pushing ownership of data to the user.

A deep understanding of the design aspects for defining and implementing smart contracts with Solidity.

Ethereum Smart Contracts for building Decentralized Applications (dApps). This part will focus on the hands-on dimension of the course with the very technical and software engineering aspects.

Learning Outcomes:

Upon completion of this course students are expected to be able to:

- Analyze and evaluate the stack of protocols that will form the future Web 3.0 and its decentralized nature;
- Analyze and evaluate the components of blockchain-based technologies which support Turing-complete languages;
- Survey in detail the architecture of Ethereum and the structure of the Ethereum Virtual Machine (including Byte Code interpretation);
- Dissect the Ethereum model, its consensus model, code execution, operation of its network, storage options and main actors that participate on its protocol;
- Discover the inner workings of smart contracts as means for developing decentralized applications;
- Build smart contracts using the Solidity programming language (including a deep understanding of the provided Libraries)
- Build a local Ethereum Network with Geth, and get familiar with a various development environments (e.g., Truffle, Remix - Ethereum IDE);
- Examine the interaction between the enclosed smart contract network and the external world, be aware of further implications these interactions pose to the aspect of decentralization;
- Build common implementation patterns, like modifiers and contract driven development;
- Analyze the smart contract development lifecycle (contract implementation, testing, deploying, and migrating a contract);
- Discover a set of technologies that support the backbone decentralized storage network (e.g., IPFS, Swarm).

Course Content:

- **Introduction to Blockchain and Ethereum**

- What is a Blockchain and why should I care?

- Blockchain Architectural Overview

- The Web of Trust

- Ethereum’s main components

- Ethereum’s sub-protocols

- The new generation of the Web (i.e., Web3.0)

- Smart Contracts and Decentralized Applications (dApps) o Web apps vs. dApps

- **Introduction to Smart Contracts**

- An overview to the history of smart contracts o An intro to the life-cycle of a smart contract o Ethereum's smart contract languages

- Interfacing with Ethereum Networks (overview of Ethereum Networks, Clients,

Wallets, Transactions etc.)

- The Solidity Programming Language o Development Environments

- **Blockchain technology Supporting Turing-Complete Languages**

- A comparison of Ethereum and Bitcoin

- Overview of Ethereum's tech stack, architecture

- The Ethereum reward scheme, Mist, EVM, Swarm, Whisper, Eth, Gas

- A simple Solidity Contract (Contract Walk-through)

- The Solidity compiler o Ethereum Contract ABI

- Deployment with the Web3.js or Web3J library

- **Virtual Machines and Beyond**

- History of Virtual Machines

- State replication, consensus and the Ethereum Architecture

- Introduction to the Ethereum Virtual Machine and EVM Byte Code interpretation

- Incentivisation structures, rewards schemes, and gas pricing

- **Deep-dive into Solidity**

- Understanding the different compiler versions and pragmas

- Authoring smart contracts o Contract definitions

- Basic data types

- Local and State Variables

- **Global Variables and Functions**

- Predefined Global Variables

- Structs and Enums o Mapping and Arrays

- Build-in Functions (e.g., addmod, keccak256)

- User Functions

- **Expressions and Control Structures**

- Valid expressions of the language

- Exception Handling (e.g., assert, require, revert, throw)
- Events and Logging
- Conditional logic
- Implementation of loops
- **Object Oriented Constructs**
- Contract constructor and selfdestruct
- Function Modifiers and Fallback functions
- Calling other contracts o Inheritance and Multiple Inheritance
- Declaring Abstract Classes and Interfaces
- Implementation of Abstract interfaces
- Function Overloading
- **Experimenting with Front-end Libraries**
- Intro to front-end web interfaces
- Decentralized Data Storage
- The Ethereum Name Services (ENS)
- **Unit Testing and Debugging Contracts**
- Estimating Gas Costs
- Basics of using Truffle for testing
- Troubleshooting and Debugging
- Common design patterns
- Smart Contract Security – overview of attacks on Ethereum smart contracts
- **Deployment Considerations and Other Smart Contract Platforms**
- Smart Contacts Quality Assurance
- Beyond Ethereum
- Blockchain-as-a-Service (BaaS) and the Dark Market
- Secure smart contracts with OpenZeppelin
- Experimenting with Hyperledger Besu
- Future Outlook and the Road Ahead (e.g., graph-based blockchain protocols, distributed autonomous organizations, quantum secured blockchains etc)

Learning Activities and Teaching Methods:

Teaching material including PowerPoint (PPT) presentations with extended descriptions and explanations, asynchronous video presentations, additional readings (journal articles and/or e-books), access to additional videos related to the course, synchronous meetings (Engageli), forums, chats, case studies and other formative and summative assessments.

Assessment Methods:

Formative Self-Assessment (not graded): A number of formative self-assessment questions (not graded) will be provided during each lecture. Summative Assessments: Assignment: assigned on session 6, deadline on session 10, assessed out of 100. This assignment corresponds to 10% of the total mark. In this assignment students will be asked to run Hyperledger Besu in a Docker environment and deploy several contracts on the EVM offered. Interactive Summative Activities: Assigned every session starting from Session 1 on-going until Session 12, assessed out of 100. Each interactive activity corresponds to 2.5% of the total mark. Overall, the total grade from all summative interactive activities corresponds to 12 sessions x 2.5% = 30% of the total mark. Purpose of interactive summative activities: Overall, these activities are designed to engage students with the learning outcomes leveraging on an interactive character. Different interactive approaches are utilized as follows: Interactive discussions during the class Interactive use of command line tools (including simulators) or related computational tools Interactive quiz (questions) Interactive use of a Wiki for sharing content generated by the students Final exam: Takes place after session 12, assessed out of 100, contributes 60% of the total course mark.

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Mastering ethereum: building smart contracts and dapps	Antonopoulos, Andreas M., and Gavin Wood	O'Reilly Media	2018	
Ethereum: A secure decentralised generalised transaction ledger	Wood, Gavin	Ethereum project yellow paper 151, no. 2014 (2014): 1-32. http://gavwood.com/paper.pdf	2014	
The science of the blockchain	Wattenhofer, Roger	CreateSpace Independent Publishing Platform	2016	
Swap, Swear, and Swindle: Incentive System for Swarm	Trón, Viktor, Aron Fischer, Dániel A. Nagy, Zsolt Felföldi, and Nick Johnson		2016	
A survey of attacks on ethereum smart contracts (sok)	Atzei, Nicola, Massimo Bartoletti, and Tiziana Cimoli	Springer, Berlin, Heidelberg	2017	
In International Conference on Principles of Security and Trust, pp. 164-186				

Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Scripting smart contracts for distributed ledger technology	Seijas, Pablo Lamela, Simon J. Thompson, and Darryl McAdams	IACR Cryptology ePrint Archive 2016 (2016): 1156.	2016	
Blockchains and databases: A new era in distributed computing	Mohan, C	In 2018 IEEE 34th International Conference on Data Engineering (ICDE), pp. 1739-1740. IEEE, 2018.	2018	

Selected online readings:

- ConsenSys, Decentralized Storage: The Backbone of the Third Web. https://media.consensys.net/decentralized-storage-the-backbone-of-the-third-webd4bc54e79700

- ConsenSys, A guide to available tools and platforms for developing on Ethereum. https://github.com/ConsenSys/ethereum-developer-tools-list

- ConsenSys, Ethereum Ecosystem Resources, https://github.com/ConsenSys/ethereumdeveloper-tools-list/blob/master/EcosystemResources.md

- A curated list of awesome Solidity resources, libraries, tools and more https://github.com/bkrem/awesome-solidity

Note: an updated list of readings in provided at the end of each lecture given the fact that Digital Currency and Blockchain Technologies constitute recent and rapidly evolving disciplines.