UNIVERSITY of NICOSIA

Course Syllabus

Course Code	Course Title	ECTS Credits
COMP-539DL	Smart Contracts	10
Prerequisites	**Department**	**Semester**
COMP-514DL, COMP-515DL	Computer Science	Fall
Type of Course	**Field**	**Language of Instruction**
Required for Blockchain Technologies concentration	Computer Science	English
Level of Course	**Lecturer(s)**	**Year of Study**
2nd Cycle	Dr Harald Gjermundrød	2nd
Mode of Delivery	**Work Placement**	**Corequisites**
Distance Learning	N/A	None

Course Objectives:

The main objectives of the course are to:

- provide a deep understanding of smart contracts and their role in the emergence of decentralized applications
- introduce the concept of Decentralized Autonomous Organizations (DAO) and how technologies like blockchain and smart contract can support such organizations
- provide a deep understanding of blockchain technology which supports Turing-complete programming language
- critically compare and differentiate blockchain frameworks which support Turing-complete languages and those which support a weaker programming model with respect to scalability and computational abilities
- critically compare different reward schemes for the miners in blockchain frameworks and how this influences the architecture of the frameworks.
- provide deep knowledge of the architecture of the Ethereum system including the Ethereum Virtual Machine and Byte Code interpretation
- provide deep knowledge of how to develop smart contracts using Turing-complete languages, such as Solidity on blockchain infrastructures
- expose the students to the full smart contract development lifecycle and be able to critically assess the different trade-offs when using different development “stacks”
- expose the students to development of DApps (Decentralized application development) including native integration of Ethereum.

- make students aware of various security and scalability issues when developing DApps and using smart contracts.

Learning Outcomes:

After completion of the course students are expected to be able to:

1. understand and evaluate the concept of smart contracts and how they can be used to construct Decentralized Autonomous Organizations
2. compare and evaluate different solutions to problems that may arise when using automated immutable systems such as DAO
3. understand and evaluate the components of blockchain-based technologies which support Turing-complete languages
4. explain in detail the architecture of Ethereum and the structure of the Ethereum Virtual Machine (including Byte Code interpretation)
5. critically compare and evaluate different reward schemes in blockchain technologies and how this can influence the development of smart contracts
6. develop smart contracts using Solidity (including have a deep understanding of the provided API) and various development tools
7. identify and resolve security issues/problems with smart contracts and be able to demonstrate the correctness of the resulting smart contract
8. demonstrate how to use various formal verification tools/techniques on smart contracts to assure their correctness
9. develop decentralized applications using various tools and languages, including native application development.
10. be aware of problems and challenges in DApps deployments, especially with relation to security and scalability issues, and have a deep understanding of the different tradeoffs that proposed solutions entails.

Course Content:

1. Introduction to smart contracts
   a) Basic description of smart contracts and blockchain technology
   b) History of smart contracts
   c) Smart contracts operations and management
2. Decentralized Autonomous Organizations (DAO)
   a) Basic description of DAO and Futarchy
   b) Problems with automated immutable systems, when problems/issues are discovered
3. Blockchain technology supporting Turing-complete languages

a) Ethereum architecture

b) Ethereum Virtual Machine and EVM Byte Code interpretation

c) Ethereum mining reward scheme, gas pricing

d) Ethereum development including: Whisper, Swarm, and Raiden Network and State Channels

4. Smart contract development

a) Introduction to development with Solidity

b) Development environments like: MIX (The DApp IDE), Ether.camp, and Truffle + Sublime + testRPC

c) Development and reuse of common patterns, like modifiers and contract driven development

d) Security related issues when developing smart contracts

e) Development issues like handling of exceptions and debugging of applications

f) Formal verification of smart contracts using tools like: Oyente, Why3, and Solgraph

5. Decentralized application development

a) Introduction to DApps development

b) Native application development using Java (with RPC) verses JavaScript applications

Learning Activities and Teaching Methods:

Lectures, Practical Exercises, and Assignments.

Assessment Methods:

Final Exam, Individual Assignments, Individual Programming Assignments, and Individual Lab Assignments.

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Ethereum: A Secure Decentralised Generalised	Gavin Wood	Online	2015	http://gavwood.com/paper.pdf

Transaction Ledger Final Draft				
Ethereum Studio IDE documentation	Ether.camp	Online	2016	https://nogo10.gitbooks.io/ether-camp-live-studio-primer/

Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations	Henning Diedrich	CreateSpace Independent Publishing Platform	2016	978-1523930470
The Science of the Blockchain	Roger Wattenhofer	CreateSpace Independent Publishing Platform	2016	978-1522751830
Mastering Bitcoin	Andreas Antonopoulos	O'Reilly Publishing	2014	978-1449374044

Other resources:

1. Nick Szabo (1997). Formalizing and Securing Relationships on Public Networks, *First Monday*, 2(9). (http://pear.accc.uic.edu/ojs/index.php/fm/rt/printerFriendly/548/469)
2. Christoph Jentzsch (2016). Decentralized Autonomous Organization To Automate Governance Final Draft. (https://download.slock.it/public/DAO/WhitePaper.pdf)
3. Dino Mark, Vlad Zamfir, Emin Gün Sirer (2016). A Call for a Temporary Moratorium on "The DAO" (http://hackingdistributed.com/2016/05/27/dao-call-for-moratorium/)
4. Solidity Programming Language (https://solidity.readthedocs.io/en/develop/)