International Programs Department

Academic Year: 2019-2020

<Insert Course Name>

Description

Blockchain, Bitcoin & Security introduces the possibilities of blockchain technology to the students. They will learn how a decentralised organisation can make the technology secure by design, and what the benefits and constraints come from such a technology.

The core of the course will be focused on Ethereum smart contract development, with an emphasis on development best practices. The entire ecosystem will also be introduced, from the development to deployment.

Learning Objectives and Outcomes

- Learn about the origins of Blockchain, Bitcoin, and fundamental notions such as proof of work
- Solidity language fundamentals 1: Truffle, smart contracts, variables, arrays, structs, functions, conditions, loops, function visibility, events
- Solidity language fundamentals 2: mappings, addresses, inheritance, storage vs memory, interfaces, immutability, ownable contracts, function modifiers, gas, public function and security
- Introduction to OpenZeppelin library and token standards: ERC20, ERC721, introduction to others
- Unit testing smart contracts, test net faucets, deployment to a test net, real-life testing on My Ether Wallet
- Introduction to front-end interface with web3.js

Course Schedule and Contents

Session#1
Total: 3h

1h30

- *Acquaintance, assessment and expectations*
- *Course description and objectives*

1h30

- *History of bitcoin*
- *Proof of work and security*
- *Ethereum and smart contracts*

Session#2
Total: 3h

- Quiz Test (at home or before class)

1h30

- *Configuration of VSCode, Solidity linter, Truffle, code structure*

1h30

- *Live coding: CryptoZombies with modern Solidity: contracts, state variables, integers, math operations, structs, arrays, functions, private/public functions, function modifiers, natspec comment norm, keccak256 and typecasting, events, testing with Mocha and Chai*

Session#3
Total: 3h

1h30

- *Live coding: Mappings and addresses, msg.sender, require, inheritance, import, storage vs memory, function visibility, interaction between contracts, interfaces, multiple return values*

1h30

- *Immutability of smart contracts, OpenZeppelin and Ownable contracts, onlyOwner modifier, Gas, Time units, Public Functions and Security, view functions, storage cost, for loops,*

Session#4
Total: 2h

1h30

- *Live coding: Payable functions, withdraws, random numbers*
- *Guided exercises, refactoring*

30 min

- *Introduction to OpenZeppelin tokens and standards, OpenZeppelin's SafeMath and security*

Session #5
Total: 2h

45 min

- *Live coding: Deployment to Ropsten test net, introduction to Metamask, My Ether Wallet*

1h

- *Live coding: Introduction to web3.js to interact with deployed smart contracts*

Session #6
Total: 2h

2h

- *Assignment #2 class presentation*

Assignment #1 - Group class project

- Development of a mini-Twitter on Ethereum: group project for groups of 5 students
- Users should be able to attach an account name to their address
- Users should be able to post even without an account name
- Users should be able to edit their posts

- Users should be able to delete their own posts
- Any user should be able to read any other users’ posts
- Smart contracts should be deployed on Ropsten
- Smart contracts should be tested with at least 80% coverage
- Any front-end technology is allowed
- The design aspect will only be considered as a bonus

Assignment #2 - Research and oral presentation

- Report (1 page) + oral presentation (10 min presentation + 5 min questions) for groups of 5 students
- Possible topics:
  - Private blockchain
  - IPFS
  - Filecoin, SiaCoin (storage-based protocols)
  - Other alternative protocols
  - Tezos
  - DeFi (Decentralised Finance)
  - Binance
  - The biggest blockchain hacks
  - Stable coins
  - Security tokens
  - Non Fungible Tokens (NFT)
  - ICO, STO
  - Oracles
- Assignments will be graded on the interest raised among the other students:
  - Quality of questions asked
  - Number of questions asked

  - Global class reaction

- General advice:
  - Topics are quite broad (even with overlaps): feel free to choose a specific topic (technical approach or not) to focus on after a general introduction
  - Your grade will depend on the reaction of the rest of the class, so listen in the interest of your classmates

Grading <Feel free to modify/adapt…)

Quiz:	10%
Ass#1 Group class project:	50%
Ass#2 Group Research and Presentation:	40%

Policies

- I expect you to turn-in your reports on time to receive proper credit/grade.
- Any work submitted must be your own.
- I expect everyone to contribute equally to group assignments
- Attendance in every class is expected and class participation and discussion is strongly encouraged.
- Late work will not be accepted unless prior arrangements have been made directly with me.
- Cases will be decided on an individual basis.

Good Luck!