CSIT6000Q (L1) - Blockchain and Smart Contracts

Jump to Today

Course Description:

This course provides a thorough technical introduction to pertinent subjects in the larger blockchain and smart contract ecosystem. This course is designed to focus on providing students with a solid understanding of extensive knowledge of what Blockchain is and how it works, with a focus on Ethereum and Smart Contracts. It will provide many opportunities for building decentralized applications using the Web3 stack and the Turing-complete Solidity language over the Ethereum Virtual Machine (EVM). The course also provides opportunities to design and deliver a final project targeting a specific blockchain use case using smart contracts of their choice (will be given options) running on an Ethereum testnet or private blockchain.

Course Prerequisites:

Students should have some basic programming knowledge.

Evaluation:

Evaluation of each student will be done based on the following: 1) Assignments- 45% (There will be 3 Assignment)
2) Term Project- 15%
3) Final- 40%

Note: the above evaluation scheme may change slightly during the course.

Term Project:

There will be a major term project (worth 15% of the final mark) on Smart Contract Design. The project topics will be discussed in class. Students will be asked to prepare, submit, and present materials (Word & PowerPoint & Source Code) related to the project throughout the course.

Course Contents:

Blockchain Fundamentals

This module introduces the foundations of blockchain technology and how it is used to solve problems in the real world. It covers the key technological components of the blockchain architecture through examples and case studies.

Topic

• Origin and Goals.
• Different Blockchain Types.
• Wallets and Keys.
• Consensus Protocols.
• Blockchain Transactions.
• Blockchain Architecture

Smart Contract Fundamentals

We'll also examine smart contract technology and go over examples of its implementation in various markets and sectors today. This module aims to introduce the reasons for a smart contract and its critical role in transforming blockchain technology from enabling decentralized systems. We will explore a smart contract's structure and basic concepts through examples and illustrate Remix (remix.ethereum.org) web IDE for deploying and interacting with a smart contract.

Topic

• Introduction to Ethereum and Smart Contracts. • Solidity.
• Decentralized Finance (DeFi).
• Non-Fungible item Tokens (NFTs).

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

Solidity

The foundations of Solidity, a high-level language that combines Javascript, Java, and C++, are what we aim to master. It is specifically made to target the Ethereum Virtual Machine and construct smart contracts. Students will be able to practice using Solidity and follow along with demos.

Topic

• Smart contract layout
• The structure of .sol source file
• Understanding the different compiler versions and pragmas • Authoring smart contracts
• Contract definitions
• Basic data types

• Local and State Variables

- Predefined Global Variables
- Structs and Enums
- Mapping and Arrays
- Build-in Functions (e.g., addmod, keccak256)
- User Functions
- Valid expressions of the language
- Exception Handling (e.g., assert, require, revert, throw)
- Events and Logging
- Conditional logic
- Implementation of loops
- Contract constructor and selfdestruct
- Function Modifiers and Fallback functions
- Calling other contracts
- Inheritance and Multiple Inheritance
- Declaring Abstract Classes and Interfaces
- Implementation of Abstract interfaces
- Function Overloading

Other Smart Contract Designing Languages

Topics will cover other popular smart contract designing languages such as Vyper, Rust.

Topic

- Fundamental of Vyper.
- Fundamental of Rust.

Best Practices: Smart Contract Development, Deployment, and Testing

The topics will include those relating to Remix IDE, writing Solidity smart contracts, and determining whether a blockchain-based solution is appropriate for your situation.

Topic

- Demonstration - Metamask
- Demonstration - Remix
- Smart Contract Design using Solidity
- Demonstration - Etherscan
- Alchemy
- Truffle vs Hardhat
- 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.

Project Topic

- OpenZeppelin Libraries
- ERC20 Token Standard
- Access Control library
- Demonstration - ERC20 Standard • Token minting
- Testing
- Deployment

Name	Role	Email Address
Pritom Rajkhowa	Instructor	pritom.rajkhowa@gmail.com pritomraj@ust.hk
Prasanta Saikia	Teaching Assistant	

Course Summary:

Date	Details	Due