

FIN579 Solidity Programming

Level: 5

Credit Units: 5 Credit Units

Language: ENGLISH

Presentation Pattern: EVERY JULY

Synopsis:

FIN579 Solidity Programming is a hands-on course in using the Solidity programming language to write and deploy Smart Contracts on Ethereum Virtual Machine (EVM) based blockchain networks. The course provides an introduction to Solidity and its background, including relevant concepts on development tools, libraries and Smart Contract standards (ERC). The course will also expose the students to the workflow and components involved in creating Smart Contracts and deploying it on the blockchain. The course will help students to appreciate the idiosyncrasies in writing Smart Contracts against traditional programming as well as its implications on security, cost, and other pitfalls.

Overall, this intensive course aims to build students' capabilities in reading, understanding, writing and deploying Smart Contracts on their own. After this course, students are expected to have grasped the knowledge and programming skills necessary as a Smart Contract programmer. This course will involve the use of Javascript and assumes that the students are familiar with blockchain, exposed to Linux-based operating systems and have some general programming knowledge.

Topics:

- Solidity Language Construct
- Solidity development tools
- Solidity development libraries
- Gas and transaction costs
- ERC20 and ERC20-related token standards
- ERC721 and ERC721-related non-fungible token (NFT) standards
- Deploying smart contracts
- Ethereum Testnets
- Testing and Debugging
- Application integration using Web3 library
- Inline assembly language
- Solidity compilers versions

Learning Outcome:

- Evaluate the Solidity language construct and the gas cost implications of Smart Contracts design based on Ethereum versions, transaction operations and storage
- Construct Smart Contracts utilising major Ethereum standards such as ERC20, ERC721, Gnosys MultiSigWallet
- Assess development workflow from compiling Solidity Smart Contracts into Application Binary Interface (ABI) and Ethereum Bytecodes, to deploying it to Ethereum
- Critique major software development components within the Ethereum ecosystem
- Design and develop Ethereum smart contracts using Solidity programming language, Web3 client and OpenZeppelin libraries
- Create, test and deploy Ethereum smart contracts using Truffle development tools

Assessment Strategies - Regular Semester (Evening Class):

Components	Description	Weightage Allocation (%)
Overall Continuous Assessment	PARTICIPATION 1	10
	GROUP BASED ASSIGNMENT 1	40
Overall Examinable Components	ECA	50
Total		100

*The information listed is subject to review and change.