University at Buffalo
The State University of New York

CSE426/526: Blockchain Application Development

COURSE INFORMATION

3 Credits
Lecture – 3 credits
Lab – N/A

INSTRUCTOR INFORMATION

Dr. Bina Ramamurthy

- *Email:* bina@buffalo.edu
- *Webpage:* http://www.cse.buffalo.edu/~bina/cse426/Fall2020
- *Office hours:* Tue and Thu: 10 A.M. to 11.30 A.M. (online: individual, private and/or group)

COURSE DESCRIPTION

This course is intended for students interested in learning about blockchain technology and developing applications using the blockchain concepts. It begins with the definition of the blockchain as a trust layer over the internet. Blockchain is meant for working with distributed resources, decentralized, and disintermediated control. Topics include: Definition of a blockchain in terms of transactions, blocks, and chain of blocks, data structures enabling the blockchain protocol and operational details involving algorithms and techniques such as peer-to-peer transactions, cryptography, digital signing and hashing, and consensus mechanisms. All of these concepts will be illustrated using Bitcoin and Ethereum blockchain. In the second part of the course, we introduce the idea of code execution on the blockchain and the program module called smart contract and a language, Solidity, for writing smart contracts, compiling, deploying, and testing the smart contracts on Ethereum blockchain. In the last part of the course, we introduce a decentralized application (Dapp) stack and explore problem-solving using blockchain. This process involves designing and developing a Dapp stack with the computational logic represented by the smart contract code, a user interface, and support for external data access through oracles and decentralized file systems. Students will work on hands-on end-to-end Dapp projects using the Ethereum blockchain and Truffle integrated development environment (IDE). The course will also discuss standards, best practices, and current challenges, such as scalability and interoperability, and the respective solutions. On completion of the course, a student will be able to analyze a problem, develop, and implement an end-to-end blockchain solution on Ethereum blockchain.

COURSE PREREQUISITE: CSE250 Data Structures, equivalent, or permission of the instructor

TEXT BOOK:

B. Ramamurthy. Blockchain in Action. Manning publishers. https://www.manning.com/books/blockchain-in-action, ISBN-13: 978-1617296338, 2020.

COURSE ORGANIZATION / SCHEDULE

Week#	Description
1	History of blockchain; Blockchain as a method for enabling peer-peer transactions; blocks of transactions, chain of blocks and distributed ledger technology; review transactions on real-world blockchain.
2	Bitcoin protocol: UTXO (unspent transaction output) model; Mining, verification and validation; and proof of work; Ethereum protocol: concept of account and addresses; incentive model and the cost of trust; immutability and anonymity.
3	Issues: block size, timestamp, scalability, hard and soft fork, governance.
4	Code execution on the blockchain; Introduction to smart contract and Ethereum Virtual Machine (EVM); alternative implementations of smart contract in chain code; application-specific validation and verification.
5	Introduction to Solidity, a Turing-complete language for writing smart contracts: Basic data types, blockchain-specific data structures such as mapping, address, message.
6	Problem solving using finite state machine design and transforming the design into smart contract; Solidity programming details: access modifiers and function modifiers. Defining events and pushing notifications and listeners.
7	Remix integrated development environment (IDE) for smart contract development and testing.

8	Best practices in development of smart contracts: when to use blockchain and what to store on the blockchain? Privacy, integrity, and confidentiality.
9	Blockchain server concept; setting up a blockchain network; working of a blockchain node; go-ethereum node (geth); Ethereum APIs: web3 API for integrating traditional web to a blockchain network.
10	Definition of decentralized application stack; Truffle development environment; End-end application development process using Truffle; test-driven application development.
11	Off chain and on-chain data. Interplanetary File system (IPFS) for decentralized data.
12	Application models, distributed apps (Dapps), side chains; Use of Multi-sig, escrow, oracles.
13	Blockchain platforms: Linux foundation's Hyperledger project: Intel Sawtooth; IBM's Fabric. Use cases in healthcare and Fintech application domains.
14	Advanced topics, if time permits; a week's time left open for exam and quizzes.

STUDENT LEARNING OUTCOMES

Course Learning Outcome	Program Outcomes (CS / CEN)	Assessment Method(s)	Assessment types
Explain the history of blockchain	1(CS); 1(CEN)	Summative quiz, Exam	Quizzes, Exam
Explain a transaction, block and blockchain and distributed ledger.	1(CS); 1(CEN)	Summative quiz, Exam	Quizzes, Exam
Discuss how immutability and consensus are achieved, and the transaction verification process.	2(CS); 2(CEN)	Summative quiz, Exam	Quizzes, Exam
Design a blockchain solution using a FSM design, implement a smart contract for the design using Solidity and test it on the Remix IDE.	1,2 (CS); 1,2 (CEN)	Graded lab, Exam	Programming Assignments, Exam

Program outcomes correspond to student outcomes provided by ABET: Blank: no coverage, 1: introduced 2: practices or reinforced 3: mastered
See this link for program outcomes:
https://docs.google.com/spreadsheets/d/1gjDtNAi4Tu4VyUUH7ECvaW_sAjH64vgH6zYhJQ4ixNU/edit#gid=708630931

CS:

PROGRAM OUTCOME	1	2	3	4	5	6
SUPPORT LEVEL	2	2	0	0	0	0

CEN:

PROGRAM OUTCOME	1	2	3	4	5	6	7
SUPPORT LEVEL	2	2	0	0	0	0	0

COURSE REQUIREMENTS

- Students will be assessed through lecture attendance, weekly quizzes, code modules developed during lecture and a term project.
- **Code development and submission:** There is no final exam. But every week there is a product (smart contract or Dapp) that will be demonstrated during the lecture. Students are required to submit the code that they developed to follow along the professor during the lecture. These items represent the parts of the design and development of a Dapp and will help you in imagining and developing the solution for your term-project.
- **Term project**: Students will setup, test, deploy and transact on a blockchain as well as design, program and test a smart-contract on this blockchain. They will work on problem solving using blockchain by designing, developing and test a Dapp.

- **Quizzes**: There will be a quiz at least once a week during the lecture time. The quizzes have to be completed during the class time and there are no makeup quizzes. Quizzes will be used to assess students on learning outcomes that are not amenable to be assessed through programming assignments.

Grading Policy

Learning assessments will be graded based on rubric criteria and weighted according to the following break-down.

Weight	Assessment / Assignment
10%	Class attendance
25%	Quizzes (online)
30%	Code snippet created during lectures (submit)
35%	Term project: incrementally developed over the semester with specific milestones to meet

Final Grades:

Grade	Quality Points	Percentage
A	4.0	93.0% -100.00%
A-	3.67	90.0% - 92.9%
B+	3.33	87.0% - 89.9%
B	3.00	83.0% - 86.9%
B-	2.67	80.0% - 82.9%
C+	2.33	77.0% - 79.9%
C	2.00	73.0% - 76.9%
C-	1.67	70.0% - 72.9%
D+	1.33	67.0% - 69.9%
D	1.00	60.0% - 66.9%
F	0	59.9 or below

Incompletes (I/IU): A grade of incomplete (“I”) indicates that additional course work is required to fulfill the requirements of a given course. Students may only be given an “I” grade if they have a passing average in coursework that has been completed and have well-defined parameters to complete the course requirements that could result in a grade better than the default grade. An “I” grade may not be assigned to a student who did not attend the course.

Incompletes (I/IU): Prior to the end of the semester, students must initiate the request for an “I” grade and receive the instructor’s approval. Assignment of an “I” grade is at the discretion of the instructor.

Academic Integrity

Academic integrity is a fundamental university value. Through the honest completion of academic work, students sustain the integrity of the university while facilitating the university's imperative for the transmission of knowledge and culture based upon the generation of new and innovative ideas. See the link for more details:
https://catalog.buffalo.edu/policies/integrity.html

Accessibility Resources

If you have any disability which requires reasonable accommodations to enable you to participate in this course, please contact the Office of Accessibility Resources, 25 Capen Hall, 645-2608, and also the instructor of this course. The office will provide you with information and review appropriate arrangements for reasonable accommodations. See http://www.buffalo.edu/studentlife/who-we-are/departments/accessibility.html

Course Fees

None