City University of Hong Kong
Course Syllabus

offered by Department of Computer Science
with effect from Semester B 2019/20

Part I Course Overview

Course Title:	Decentralized Applications Development
Course Code:	CS4389
Course Duration:	One semester
Credit Units:	3 credits
Level:	B4
Proposed Area: *(for GE courses only)*	☐ Arts and Humanities ☐ Study of Societies, Social and Business Organisations ☐ Science and Technology
Medium of Instruction:	English
Medium of Assessment:	English
Prerequisites: *(Course Code and Title)*	CS3343 Software Engineering Practice
Precursors: *(Course Code and Title)*	CS2204 Fundamentals of Internet Applications Development
Equivalent Courses: *(Course Code and Title)*	Nil
Exclusive Courses: *(Course Code and Title)*	Nil

Part II Course Details

1. Abstract

(A 150-word description about the course)

Decentralized applications (Dapps) are software applications that are run by multiple users on decentralized networks. Dapps have found successful usage scenarios in many domains such as software development, finance, customer services, and logistics. Many Dapps scenarios include the use of a kind of programs called *smart contracts* to perform transactions involving multiple users. The aim of this course is to provide a comprehensive study on the software design and development of Dapps, its associated software engineering practices, programming languages, development and testing environments, tools, evaluation, and current trends and issues in the aspect of software engineering. Students are expected to design, implement, test, maintain and evaluate programs that meet the constraints and requirements of high quality decentralized applications.

2. Course Intended Learning Outcomes (CILOs)

(CILOs state what the student is expected to be able to do at the end of the course according to a given standard of performance.)

No.	CILOs[#]	Weighting* (if applicable)	Discovery-enriched curriculum related learning outcomes (please tick where appropriate)		
			A1	*A2*	*A3*
1.	Describe the essential requirements, constraints and technology stack in developing Dapps.		✓		
2.	Demonstrate working knowledge on the development, test, and deployment environments of Dapps.			✓	
3.	Design, implement and maintain Dapps.				✓
4.	Evaluate the correctness and performance of Dapps.		✓		
	** If weighting is assigned to CILOs, they should add up to 100%.*	100%			

[#] Please specify the alignment of CILOs to the Gateway Education Programme Intended Learning outcomes (PILOs) in Section A of Annex.

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to self-life problems.

A3: *Accomplishments*
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

3. Teaching and Learning Activities (TLAs)

(TLAs designed to facilitate students' achievement of the CILOs.)

Teaching pattern:
Suggested lecture/tutorial/laboratory mix: 3 hrs lecture/tutorial.

TLA	Brief Description	CILO No.				Hours/week (if applicable)
		1	2	3	4	
Lectures	Lectures will cover the essential constraints, requirement knowledge, system models and frameworks on the development and deployment of Dapps.	✓	✓	✓	✓	
Tutorials	Technical questions and study cases are provided to lead students' discussions and practice of various skills in Dapps software development. A series of hands-on practices on developing and testing a Dapp from scratch will be used in tutorial sessions to strengthen students' skillset and knowledge.	✓	✓	✓	✓	
Quiz	A quiz will cover all topics learned in lectures and practices gained via tutorials as well as the working knowledge in setting up the decentralized applications' environment in the group project.		✓	✓	✓	
Project	A team-based, comprehensive software engineering project gives students an opportunity to collaborate and share in their learning process. All major topics from design to coding and testing learned in the course are required in the project.	✓	✓	✓	✓	
Presentation	Presentation will cover all aspects of the project including design decision and rationale as well as the justification, implementation and evaluation of the project.		✓	✓	✓	

4. Assessment Tasks/Activities (ATs)

(ATs are designed to assess how well the students achieve the CILOs.)

Assessment Tasks/Activities	CILO No.				Weighting*	Remarks
	1	2	3	4		
Continuous Assessment: 60%						
Quiz		✓	✓	✓	15%	
Project	✓	✓	✓	✓	40%	
Presentation	✓	✓	✓	✓	5%	
Examination^: 40% (duration: 2 hours)	✓	✓	✓	✓	40%	
** The weightings should add up to 100%.*					100%	

^ For a student to pass the course, at least 30% of the maximum mark for the examination must be obtained.

5. Assessment Rubrics

(Grading of student achievements is based on student performance in assessment tasks/activities with the following rubrics.)

Assessment Task	Criterion	Excellent (A+, A, A-)	Good (B+, B, B-)	Fair (C+, C, C-)	Marginal (D)	Failure (F)
1. Quiz	1.1 ABILITY to describe, analyse and apply software engineering techniques and write programs and tests for decentralized applications	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Project	2.1 ABILITY to set up and apply software engineering environment, tools, techniques, practices, and programming languages to develop and deploy decentralized applications 2.2 ABILITY to function as a team of developers 2.3 ABILITY to report in an organised and logical way	High	Significant	Moderate	Basic	Not even reaching marginal levels
3. Presentation	3.1 ABILITY to summarize and present complex technical ideas systematically	High	Significant	Moderate	Basic	Not even reaching marginal levels
4. Examination	4.1 ABILITY to explain software development and deployment technology stack 4.2 ABILITY to apply software design techniques and write code for decentralized applications 4.3 ABILITY to ensure the correctness and performance of decentralized applications	High	Significant	Moderate	Basic	Not even reaching marginal levels

Part III Other Information (more details can be provided separately in the teaching plan)

1. Keyword Syllabus

(An indication of the key topics of the course.)

Decentralized applications, blockchain, blockstack, transactions, smart contract, wallet, testing, debugging, performance, programming language, tools and frameworks.

Syllabus

1. Technology stack of decentralized applications
   Overview of different kinds of application styles including centralized, decentralized and distributed. Technology stacks including decentralized filesystem, decentralized networks, software architecture, programming language, platform, wallet, virtual machine, API framework and library, development and testing environments.

2. Problem solving through decentralized application programming
   Solidity programming language. Implementation of wallet, transaction, smart contract, and backend and frontend application logics.

3. Software correctness and performance
   Analysis, code review, debugging, testing, fuzzing, and maintenance.

2. Reading List

2.1 Compulsory Readings

(Compulsory readings can include books, book chapters, or journal/magazine articles. There are also collections of e-books, e-journals available from the CityU Library.)

There is no textbook for this course.

2.2 Additional Readings

(Additional references for students to learn to expand their knowledge about the subject.)

1.	*Ethereum Smart Contract Development: Build blockchain-based decentralized applications using solidity*. Mayukh Mukhopadhyay. Packt Publishing Ltd, 2018. ISBN 1788472624.
2.	*Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginner*. Chris Dannen. Apress; 1st ed. Edition, 2017. ISBN 1484225341.
3.	Solidity. https://github.com/ethereum/solidity and https://solidity.readthedocs.io/en/v0.4.24/
4.	Ethereum. https://www.ethereum.org/
5.	*Decentralized applications: Harnessing bitcoin's blockchain technology*. Siraj Raval. O'Reilly. 2016. ISBN 9781491924549.