ITP 356 – Blockchain: Decentralized Applications
Units: 4  
Spring 2023

Lecture: TuTh 2:00pm-3:50pm  
Location: THH 106

Instructor: Jack Boyuan Xu  
Office: Remote  
Office Hours: 1 hour after each class  
Contact Info: boyuanxu@usc.edu

Teaching Assistant(s): Jordan Bettencourt, Gaurav Makasare, Dylan Patterson  
Office: Remote / By appointment  
Office Hours: See Piazza  
Contact Info: jbettenc@usc.edu, makasare@usc.edu, dylanpat@usc.edu

IT Help:  
USC IT (ITS): https://itservices.usc.edu/contact/  
Viterbi IT: https://viterbi.usc.edu/resources/vit/contact-us.htm
Catalogue Description
Learn the technical skills required to build decentralized applications on public blockchains. Applications that execute business transactions without the need of a trusted third-party.

Course Description
- Main components of a decentralized application (DApp)
- Social and design challenges to adoption of DApps
- Smart contracts written in the Solidity programming language
- Development environment required to write, test, and deploy an Ethereum DApp

Learning Objectives
After completing the course, students will be able to
- Explain what smart contracts and decentralized applications (DApps) are and the role they play in the Web 3.0 stack
- Comprehend the major technical, social, and economic implications of decentralized applications
- Develop smart contracts written in the Solidity programming language
- Develop a decentralized application with ethers.js, Truffle/Hardhat, etc.
- Deploy a decentralized application onto the Ethereum testnet

Prerequisite(s): ITP 256
Co-Requisite(s): None
Concurrent Enrollment: None
Recommended Preparation: https://www.codecademy.com/learn/introduction-to-javascript
Extra Course Information (MUST READ): https://itp356.notion.site/Disk-A-3699117da3154219abe416263ee2db4c

Course Notes
Lectures are delivered face to face in classroom when possible and over Zoom when COVID policies disallow in-person attendance. Lectures are Zoom-enabled and recorded. All course materials will be made available through Blackboard. These include:
- Lecture slides
- Homework Assignments
- Readings
- Software details and instructions
- Grades and feedback
- Office hours
- Online discussion forums on Piazza

Announcements made in class and content posted in Blackboard will supersede the contents of this syllabus.

Technological Proficiency and Hardware/Software Required
The assignments for this class will include both reading assignments as well as hands-on computer assignments. Students must bring their laptop computers to lecture sessions to participate in hands-on activities (loaner laptops are available for check out). Students will be given tutorials to gain familiarity with software tools.
VMware Player – Ubuntu Virtual Machine
If you are on Windows, it is recommended to utilize the provided Ubuntu VM to avoid environment setup issues. A VM image will be provided and VMware Player is needed. See Notion for additional installation instructions.

Required Readings and Supplementary Materials
Title: Mastering Ethereum
- Author/Publisher: Andreas Antonopoulos & Gavin Wood/O’Reilly Media
- https://github.com/ethereumbook/ethereumbook

Title: Ethereum Development Documentation
- Author/Publisher: Ethereum Community
- https://ethereum.org/en/developers/docs/

Title: Solidity Documentation
- Author/Publisher: Ethereum Community

In addition to the required reading and supplementary materials listed in the weekly breakdown section of this syllabus, additional materials will be announced in class and published on Blackboard.

Description and Assessment of Assignments
Homework: Most homework is computer based. Homework should be turned in to Blackboard or GitHub classroom on time. Grading will be based on completeness, accuracy, and timeliness. These are individual effort assignments. All homework assignments are due exactly before the beginning of class of the following week (e.g., an assignment due on Week 2 is due Week 1 Tuesday 3:59PM PST).

Exams are entirely open everything. However, no communication of any kind is permitted.

Grading Breakdown
<table>
<thead>
<tr>
<th>Assignment</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>29%</td>
</tr>
<tr>
<td>Programming basics quiz</td>
<td>1%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>30%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>40%</td>
</tr>
<tr>
<td>TOTAL</td>
<td>100%</td>
</tr>
</tbody>
</table>

Grading Scale
Course final grades will be determined using the following scale

<table>
<thead>
<tr>
<th>Grade</th>
<th>Score Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93-100</td>
</tr>
<tr>
<td>A-</td>
<td>90-93</td>
</tr>
<tr>
<td>B+</td>
<td>87-89</td>
</tr>
<tr>
<td>B</td>
<td>83-86</td>
</tr>
<tr>
<td>B-</td>
<td>80-82</td>
</tr>
<tr>
<td>C+</td>
<td>77-79</td>
</tr>
</tbody>
</table>
C    73-76  
C-   70-72  
D+   67-69  
D    63-66  
D-   60-62  
F    59 and below  

Assignment Submission Policy
It is the responsibility of the student to make sure problem solutions and assignments are turned in on time. Make sure you follow the procedures outlined in each assignment (Blackboard submissions & GitHub Classroom).

Students are encouraged to work with their classmates. However, students must turn in their own, original work. Late homework submissions are NOT accepted due to in-class code reviews. You will be given a chance to correct your code after the review, but no additional points will be given if it is clear no effort was made before the code review.

Grading Timeline
Assignment grading will typically be completed 7 days after submission. Any exceptions will be announced in class or on Blackboard.

Additional Policies
The use of mobile devices or any communication software is not permitted during the exam. No make-up exams (except for documented medical or family emergencies) will be offered nor will there be any changes made to the Final Exam schedule, except as permitted by university rules.
<table>
<thead>
<tr>
<th>Week 1</th>
<th>Topics/Daily Activities</th>
<th>Readings</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bitcoin Review &amp; Ethereum Intro</td>
<td>Mastering Ethereum, Chapters “What is Ethereum” &amp; “Introduction” &amp; “Keys and Addresses” &amp; “Transactions”</td>
<td>HW0 assigned</td>
</tr>
<tr>
<td></td>
<td>• Bitcoin Blockchain Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• What is Ethereum</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Ethereum vs. Bitcoin</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Philosophy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Account &amp; State</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Block Structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Transaction &amp; Gas</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>o Applications</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 2</td>
<td>Smart Contracts, DApps, Web 3.0, Tokens</td>
<td>Mastering Ethereum, Chapters “Ethereum Basics”, “Tokens”</td>
<td>HW1 assigned</td>
</tr>
<tr>
<td></td>
<td>• Smart Contracts</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Ethereum Virtual Machine (EVM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DApp vs. App</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Web 3 vs Web 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Tokens on Ethereum</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 3</td>
<td>Development Environment</td>
<td></td>
<td>HW0 due HW1 due HW2 assigned</td>
</tr>
<tr>
<td></td>
<td>• State of Developer Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Web3 Base Layer Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Deployment Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Smart Contract Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Remix IDE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Truffle &amp; Hardhat</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• MetaMask</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Local vs. Test vs. Main Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Programming basics quiz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 4</td>
<td>Solidity Language</td>
<td>Mastering Ethereum, Chapter “Smart Contracts”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Structure of a Contract</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Variables, Types &amp; Structs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Functions &amp; Parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Visibility &amp; Getters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Solidity Programming I</td>
<td>Solidity Programming II</td>
<td>Contract Deployment</td>
</tr>
<tr>
<td>------</td>
<td>------------------------</td>
<td>-------------------------</td>
<td>---------------------</td>
</tr>
</tbody>
</table>
| 5    | • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance |  
      • Command Line  
      • Setting up Truffle & Hardhat  
      • Connecting Blockchain and Smart Contracts  
      • Ethers.js  
      • Local Network Deployment  
      • Testnet Deployment | | |  
      • Reentrancy  
      • Race Conditions  
      • Front Running  
      • Timestamp Dependence  
      • Over/Underflow |  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Opcodes  
      • Gas Efficiency |
| 6    |  
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance |  
      • Command Line  
      • Setting up Truffle & Hardhat  
      • Connecting Blockchain and Smart Contracts  
      • Ethers.js  
      • Local Network Deployment  
      • Testnet Deployment | | | | |
| 7    | In-class HW3 & HW4 code review |  
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance | | | | |
| 8    |  
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance |  
      • Command Line  
      • Setting up Truffle & Hardhat  
      • Connecting Blockchain and Smart Contracts  
      • Ethers.js  
      • Local Network Deployment  
      • Testnet Deployment |  
      • Testing in TypeScript | | | |
| 9    |  
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance |  
      • Command Line  
      • Setting up Truffle & Hardhat  
      • Connecting Blockchain and Smart Contracts  
      • Ethers.js  
      • Local Network Deployment  
      • Testnet Deployment |  
      • Testing in TypeScript | | | |
| 10   | Midterm Exam |  
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance | | | | |
| 11   |  
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance |  
      • Command Line  
      • Setting up Truffle & Hardhat  
      • Connecting Blockchain and Smart Contracts  
      • Ethers.js  
      • Local Network Deployment  
      • Testnet Deployment |  
      • Testing in TypeScript | | | |
| 12   |  
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance |  
      • Command Line  
      • Setting up Truffle & Hardhat  
      • Connecting Blockchain and Smart Contracts  
      • Ethers.js  
      • Local Network Deployment  
      • Testnet Deployment |  
      • Testing in TypeScript | | | |
|      | Mastering Ethereum, Chapter “Dev Tools and Frameworks” & “Consensus” |  
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance |  
      • Command Line  
      • Setting up Truffle & Hardhat  
      • Connecting Blockchain and Smart Contracts  
      • Ethers.js  
      • Local Network Deployment  
      • Testnet Deployment |  
      • Testing in TypeScript | | | |
      • Static vs. Dynamic State  
      • Arrays & Memory Structure  
      • Loops  
      • Scoping and Declarations  
      • Function Modifiers |  
      • Error Handling with Require, Revert  
      • Conditionals  
      • Events  
      • Inheritance |  
      • Command Line  
      • Setting up Truffle & Hardhat  
      • Connecting Blockchain and Smart Contracts  
      • Ethers.js  
      • Local Network Deployment  
      • Testnet Deployment |  
      • Testing in TypeScript | | | |
<p>|      | HW2 due | HW3 due | HW3 due | HW4 due | HW4 due | HW5 assigned | HW5 assigned | HW6 assigned |</p>
<table>
<thead>
<tr>
<th>Week 13</th>
<th>In-class HW6 code review</th>
<th>In-class HW7 workshop</th>
<th>&quot;Mastering Ethereum, Chapter &quot;Ethereum Standards (EIPs/ERCs)&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenZeppelin Library</td>
<td>• Ownable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Pausable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Upgradeable (Proxy pattern)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating your own token</td>
<td>• ERC20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ERC721</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://docs.openzeppelin.com/contracts/4.x/tokens">https://docs.openzeppelin.com/contracts/4.x/tokens</a></td>
</tr>
<tr>
<td>Week 14</td>
<td>Working with contract interfaces</td>
<td>In-class HW7 workshop</td>
<td><a href="https://consensys.net/blog/blockchain-explained/what-is-a-dao-and-how-do-they-work/">https://consensys.net/blog/blockchain-explained/what-is-a-dao-and-how-do-they-work/</a></td>
</tr>
<tr>
<td></td>
<td>Decentralized decision making</td>
<td></td>
<td><a href="https://ethereum.org/en/developers/docs/smart-contracts/anatomy/">https://ethereum.org/en/developers/docs/smart-contracts/anatomy/</a></td>
</tr>
<tr>
<td></td>
<td>• The rise of DAOs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Community governance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 15</td>
<td>Advanced topics</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Indexing the blockchain with The Graph</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Using Chainlink Price Feeds</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Storing data on IPFS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Alchemy Gateway</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web3 Infrastructures</td>
<td>• Gateways</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Oracles</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Indexing services</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Decentralized file storage</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://www.web3.university/article/what-is-a-node-provider">https://www.web3.university/article/what-is-a-node-provider</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://chain.link/education/blockchain-oracles">https://chain.link/education/blockchain-oracles</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://medium.com/graphprotocol/graphql-will-power-the-decentralized-web-d7443a69c69a">https://medium.com/graphprotocol/graphql-will-power-the-decentralized-web-d7443a69c69a</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://docs.ipfs.io/concepts/what-is-ipfs/">https://docs.ipfs.io/concepts/what-is-ipfs/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://arweave.medium.com/what-is-arweave-explain-like-im-five-425362144eb5">https://arweave.medium.com/what-is-arweave-explain-like-im-five-425362144eb5</a></td>
</tr>
<tr>
<td>Week 16</td>
<td>Decentralized Application</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Degrees of decentralization</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Sample DAapps</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Working with other EVM chains</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://polygon.technology">https://polygon.technology</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://www.avax.network">https://www.avax.network</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://www.bnbchain.world/">https://www.bnbchain.world/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://fantom.foundation">https://fantom.foundation</a></td>
</tr>
<tr>
<td>Week 17</td>
<td>Final exam review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Exam</td>
<td></td>
<td></td>
<td>Date: For the date and time of the final for this class, consult the USC Schedule of Classes at <a href="http://www.usc.edu/soc">www.usc.edu/soc</a>.</td>
</tr>
</tbody>
</table>
Statement on Academic Conduct and Support Systems

Academic Conduct
Plagiarism – presenting someone else’s ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in Scampus in Part B, Section 11, “Behavior Violating University Standards” https://policy.usc.edu/scampus-part-b/. Other forms of academic dishonesty are equally unacceptable. See additional information in Scampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Support Systems
Student Counseling Services (SCS) - (213) 740-7711 – 24/7 on call
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention. https://engemannshc.usc.edu/counseling/

National Suicide Prevention Lifeline - 1-800-273-8255
Provides free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week. http://www.suicidepreventionlifeline.org

Relationship and Sexual Violence Prevention Services (RSVP) - (213) 740-4900 - 24/7 on call
Free and confidential therapy services, workshops, and training for situations related to gender-based harm. https://engemannshc.usc.edu/rsvp/

Sexual Assault Resource Center
For more information about how to get help or help a survivor, rights, reporting options, and additional resources, visit the website: http://sarc.usc.edu/

Office of Equity and Diversity (OED)/Title IX Compliance – (213) 740-5086
Works with faculty, staff, visitors, applicants, and students around issues of protected class. https://equity.usc.edu/

Bias Assessment Response and Support
Incidents of bias, hate crimes and microaggressions need to be reported allowing for appropriate investigation and response. https://studentaffairs.usc.edu/bias-assessment-response-support/

The Office of Disability Services and Programs
Provides certification for students with disabilities and helps arrange relevant accommodations. http://dsp.usc.edu

Student Support and Advocacy – (213) 821-4710
Assists students and families in resolving complex issues adversely affecting their success as a student EX: personal, financial, and academic. https://studentaffairs.usc.edu/ssa/

Diversity at USC
Information on events, programs and training, the Diversity Task Force (including representatives for each school), chronology, participation, and various resources for students. https://diversity.usc.edu/

USC Emergency Information
Provides safety and other updates, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible, http://emergency.usc.edu

USC Department of Public Safety – 213-740-4321 (UPC) and 323-442-1000 (HSC) for 24-hour emergency assistance or to report a crime.
Provides overall safety to USC community. http://dps.usc.edu