CS 521

Technological Foundations of Blockchain and Cryptocurrency

Course Structure and Syllabus — Spring 2026

Grigore Rosu

The Grainger College of Engineering • University of Illinois

Course Overview

CS 521 explores the technological foundations of blockchain and cryptocurrency, from the history of money and cryptographic primitives through Bitcoin, Ethereum, consensus mechanisms, and modern applications. The course combines **instructor-led lectures** on core topics with **student-driven projects** that deepen understanding through research, presentation, and implementation.

Logistics

Schedule

Day	In Person	Online
Wednesdays	Urbana (~45 students)	~15 students
Fridays	Chicago (~15 students)	~45 students

The course runs in hybrid format across both campuses. All lectures and presentations are accessible to both in-person and online students.

Course Structure

The course has two main components: **instructor lectures** covering foundational material, and a **semester-long pair project** combining research, presentation, and coding on a single topic.

Component 1: Instructor Lectures

The instructor will deliver lectures on core topics throughout the semester, including:

- Short History of Money
- Basic Crypto Primitives
- Bitcoin
- Ethereum and Smart Contracts
- Consensus Mechanisms
- Scaling and Layer 2 Solutions
- DeFi, NFTs, and Applications
- Security, Attacks, and Formal Verification

These lectures provide the shared foundation that all students are expected to master.

Component 2: Pair Project

Students form **pairs (groups of 2)**. In exceptional cases, a group of 3 may be approved, but pairs are strongly preferred. Each pair selects one topic from a curated list provided by the instructor. The same pair works on **both the presentation and the coding project** for that topic throughout the entire semester.

Each topic on the curated list comes with a defined scope for both the conceptual presentation and the coding implementation, ensuring both dimensions have sufficient depth.

Topic sharing: Since there are approximately 30 pairs and 15–20 curated topics, two pairs may work on the same topic. Each pair works independently, but awareness of the other pair’s approach is encouraged. Presentations and implementations must be distinct.

Semester Timeline

Period	Activity	Deliverables
Weeks 1–4	Foundations	Instructor lectures on core topics. Pairs form and select topics from the curated list.
Week 5	First Presentation	Each pair submits a recorded presentation (10–12 min) providing a conceptual overview of their topic. Captions required.
Week 6	Reflection #1	Individual written reflection (2 pages): connect your topic to the foundational material from lectures.
Weeks 6–11	Implementation	Instructor continues lectures. Pairs build their coding project. Brief progress check-in with instructor.
Week 12	Second Presentation	Improved recorded presentation (10–12 min) incorporating feedback, plus a live demo of the coding project. Captions required.
Week 13	Reflection #2	Individual written reflection (2 pages): technical challenges, design decisions, and lessons learned from coding.
Weeks 14–15	Live Showcase	Selected pairs (6–8) present live to the full class. Serves as a showcase of the best and most interesting projects.
Week 16	Final Deliverables	Final report (pair) + Reflection #3 (individual, 2 pages): critical self-evaluation of your contribution.

Deliverables in Detail

Recorded Presentations

- Each pair submits two recorded presentations over the semester: an initial conceptual overview (Week 5) and an improved version with a coding demo (Week 12).
- Presentations should be 10–12 minutes in length and must include captions (auto-generated is fine).
- All presentations will be made available to the entire class as a shared learning resource.
- The instructor will review all presentations and provide written feedback. Selected presentations may be invited for a live showcase in Weeks 14–15.

Coding Project

- The coding project implements concepts from your chosen topic. The scope is defined in the curated topic list.
- Code must be maintained in a Git repository with meaningful commit history from both partners. The commit history serves as evidence of individual contribution.
- A live demo is included in the second recorded presentation (Week 12).

Individual Reflections

Each student submits **three individual written reflections** over the semester (2 pages each). These are the primary mechanism for individual assessment. Reflections must demonstrate your personal understanding and engagement with the material.

- Reflection #1 (Week 6): Connect your project topic to the foundational course material. Demonstrate understanding of how the crypto primitives, blockchain structures, and concepts from lectures relate to your specific topic.
- Reflection #2 (Week 13): Discuss the technical challenges encountered during implementation, design decisions made, and what you learned from building the code.
- Reflection #3 (Week 16): Critically evaluate your own contribution to the pair project. What would you do differently? What did you learn about the topic that surprised you?

Note on AI usage: You may use AI tools to assist with writing. However, reflections that demonstrate only surface-level understanding will receive lower grades regardless of writing quality. The purpose is to show *your* understanding, not to produce polished prose.

Final Report

- Each pair submits one final report (approximately 10 pages) combining a conceptual synthesis of the topic with a technical description of the implementation.
- The report should incorporate all feedback received over the semester and serve as a comprehensive reference on the chosen topic.
- Both partners are credited equally on the report; individual differentiation comes from the reflections and git history.

Grading

Component	Weight	Type
Recorded Presentations (2)	**25%**	Pair
Coding Project + Git History	**25%**	Pair*
Individual Reflections (3)	**35%**	Individual
Final Report	**10%**	Pair
Participation + Live Showcase	**5%**	Individual

***Coding Project:** While the project is a pair deliverable, the git commit history will be reviewed to assess individual contribution. Significant imbalance in contributions may result in different grades for each partner.

Policies

Pair Formation

- Pairs should be formed by the end of Week 2. Students may form pairs across campuses (Chicago/Urbana), but should be aware this means fully remote collaboration.
- If you cannot find a partner, contact the instructor for assistance.
- In exceptional circumstances, a group of 3 may be approved. Solo projects are not permitted.

Topic Selection

- Topics are selected from the curated list on a first-come, first-served basis.
- Each topic can be taken by at most two pairs. Pairs on the same topic work independently.
- Topic selection opens in Week 2 and closes at the end of Week 3.

Pair Issues

If a pair experiences collaboration difficulties, contact the instructor before Week 5. After the first presentation, restructuring pairs becomes significantly harder. Early communication is essential.

Academic Integrity

All individual reflections must represent your own understanding. Code must be original work by the pair, with appropriate attribution for any external libraries or references used. AI tools may be used as assistants for both writing and coding, but you are responsible for understanding and being able to explain everything you submit.

Late Submissions

Recorded presentations and reflections are due on the specified dates. Late submissions receive a 10% penalty per day, up to 3 days. After 3 days, the submission receives zero credit. Extensions may be granted for documented emergencies.