<div style="border:1px solid black; padding:10px; text-align:center;">

Blockchain and Cryptocurrencies (CS2361)
Monsoon 2023

</div>

Instructor: Mahavir Jhawar. [mahavir.jhawar@ashoka.edu.in]

Course Overview: The course will provide a comprehensive introduction to principles and core topics in crypto-currencies and the wider blockchain space. It will begin with a fast-paced introduction to relevant concepts from cryptography. The course will then focus extensively on understanding the core concepts of Bitcoin and distributed consensus in general. In the process, we will explore a list of important concepts and enablers relating to blockchain technology. We will also introduce Ethereum, its architecture, and will prepare you to write smart contracts and build distributed applications using Ethereum. The course will also introduce the Hyperledger Fabric blockchain platform with focus on building distributed applications such as central bank digital currencies (CBDC).

Selected lecture slides of the previous iteration of the course are available here `https://sites.google.com/view/brgashoka/blockchain-course?authuser=0`. A few course project (a major component of the course) demos are also available.

Prerequisites: The course requires an overall maturity in Computer Science and Mathematics. The hard pre-requisite are

- Introduction to Computer Programming
- Data Structures

While the course on Computer Networks is recommended, it is not mandatory.

Grading: Your final grade in the course will be calculated as per the policy given in Table 1.

Evaluation Type	Weightage		Letter Grade	Percentage Bracket
Mid Term	15%		F	< 40
Assignments [1]	40%		D-	$40 - 44$
Final Term	15%		D	$45 - 49$
Course Project	30%		D+	$50 - 54$
			C-	$55 - 59$
			C	$60 - 64$
			C+	$65 - 69$
			B-	$70 - 74$
			B	$75 - 79$
			B+	$80 - 84$
			A-	$85 - 90$
			A	> 90

Table 1: Grading Policy

Course Format: Beside two 90-minute weekly lectures, the course will have a mandatory 60-minute session per week. A regular 60-minute slot will be decided in the first week of the course. This session will be used primarily to hold weekly test and to enforce concepts and techniques learned in class.

Course References:

- Lecture notes.
- Mastering Bitcoin: Unlocking Digital Cryptocurrencies, by Andreas Antonopoulos
- Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder. Princeton University Press.

---

[1] "Assignments" - will include all or some of the following: theory/programming home work, in-class quizzes, and weekly in-class tests.

- Bitcoin: A Peer-to-Peer Electronic Cash System. `https://bitcoin.org/bitcoin.pdfm`.
- `https://bitcoin.org/en/developer-reference#block-chain`
- `https://github.com/ethereum/wiki/wiki`
- `https://hyperledger-fabric.readthedocs.io/en/latest/tutorials.html`

Late homework policy: Late or improperly submitted homework will not be accepted. If you know in advance that you will be unable to submit your homework at the correct time, you must make special arrangements ahead of time. Theory assignments must be written neatly and well organized. If it's not readable, it won't be graded. You should strongly consider starting with a rough draft, especially on problems requiring a proof. You might consider taking the opportunity to learn LATEX.

Academic dishonesty: Many students find it helpful to consult their peers while doing assignments. This practice is legitimate and to be expected. However, it is not acceptable practice to pool thoughts and produce common answers. To avoid this situation, it is suggested that students not write anything down during such talks, but keep mental notes for later development of their own. Students who allow their files or assignments to be copied are as guilty of academic dishonesty as those who copy and will be treated accordingly. Major occurrences of academic dishonesty, such as the submission of work that is not the student's own, will be dealt with according to the Ashoka University's academic honesty document.

How much work is expected: The amount of work will vary, depending on your background in programming skill and the ease with which you follow mathematical ideas. However, 12-14 hours per week is a good guess.