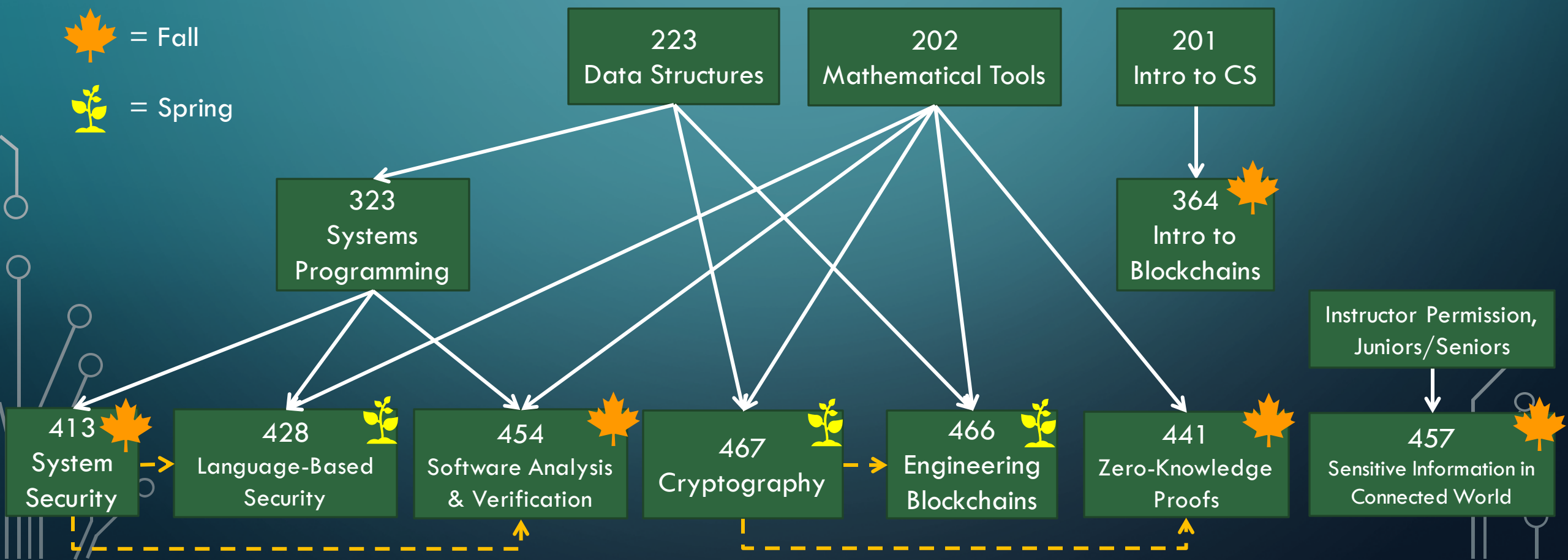
YALE SECURITY COURSES

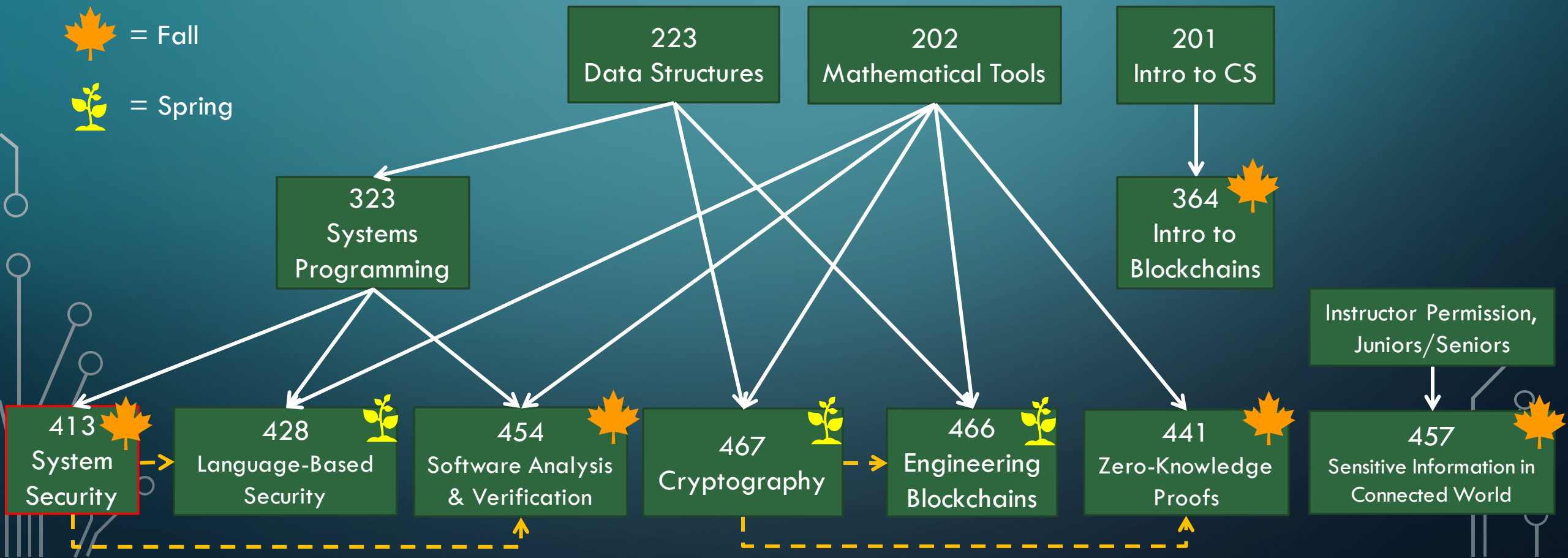
11/6/2023 ADVISING PANEL

COURSES OUTLINE



413:
What can go wrong when building software systems? How can bugs be exploited? How can we stop known attacks and reduce the risk of unknown future attacks? Topics include…

- Ethical hacking and responsible disclosure
- Secure design principles
- Authentication/Authorization
- Client/Server/Network-side attacks against web applications
- Memory corruption and control flow hijacking
- AI in security: challenges & solutions

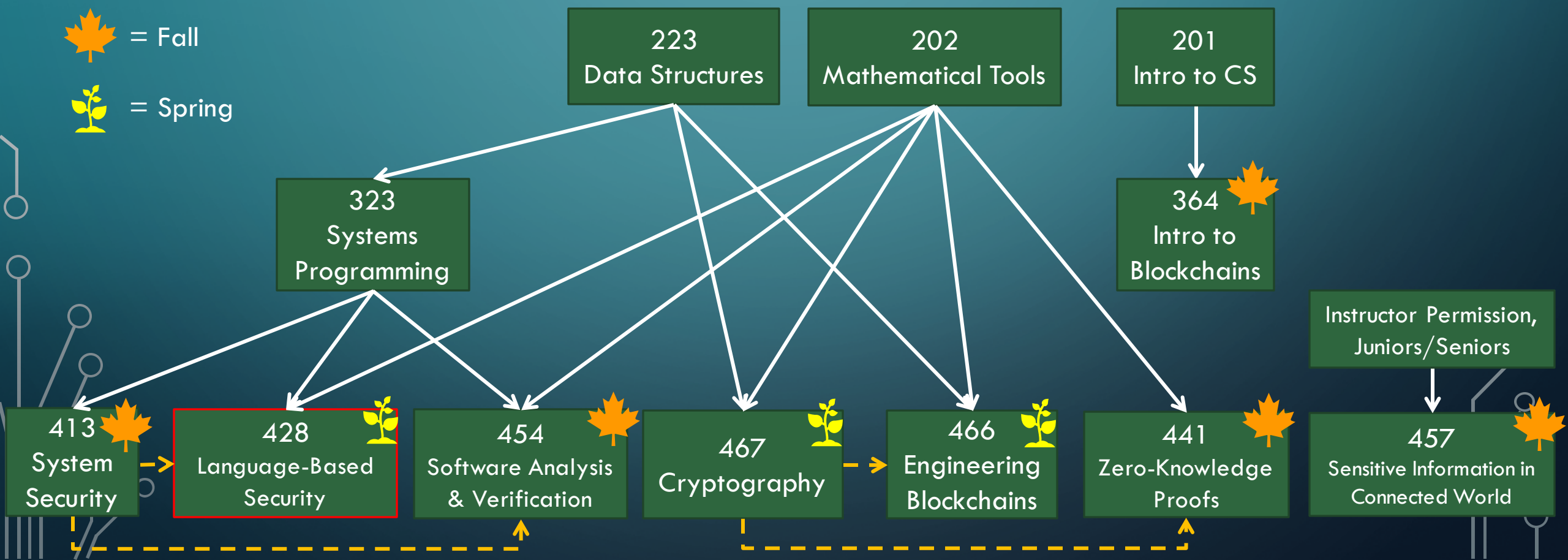


428: Tue/Thu 2:30-3:45pm

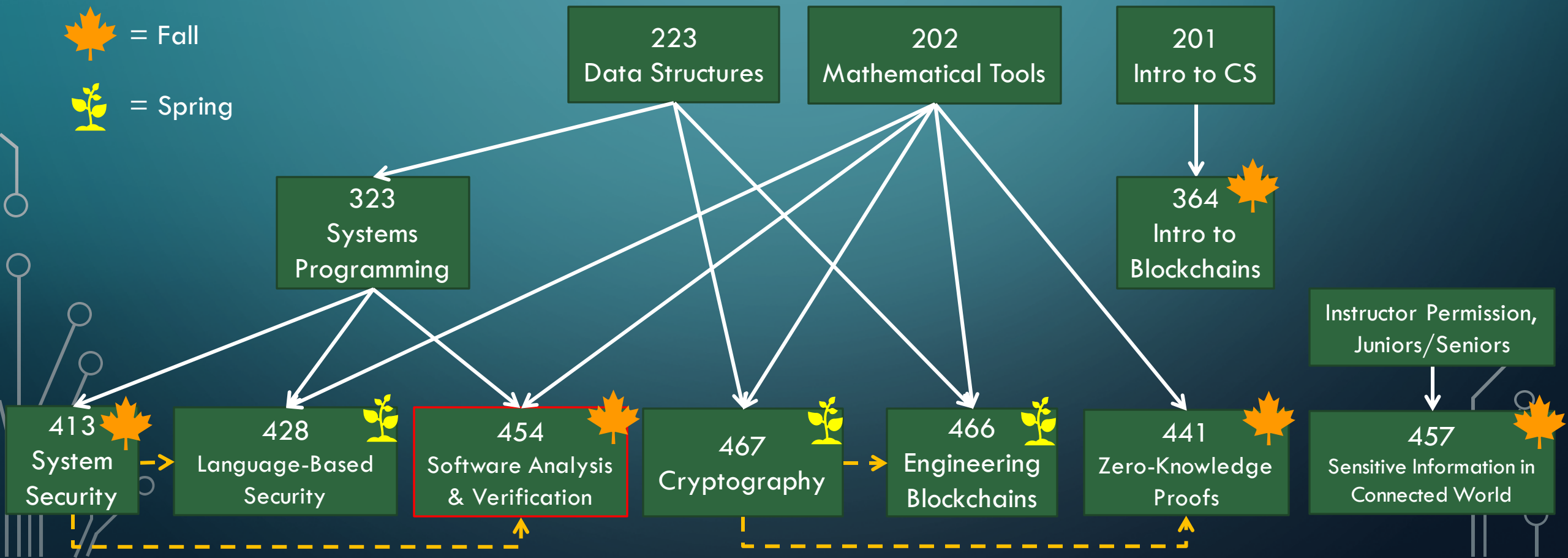
First time being offered since Spring 2020!

Design and implement features of programming languages, compilers, and runtimes to enable building secure and reliable systems. Topics include:

- Proof-carrying code
- Certifying compilation
- Typed assembly languages
- Runtime checking and monitoring
- High-confidence embedded systems and drivers
- Language support for verification of safety and liveness properties



454:
How can we formally prove that a program does what it is supposed to do?
Introduces concepts, tools and techniques to analyze software to answer this question.
The goals are similar to CPSC 428, but using different techniques.

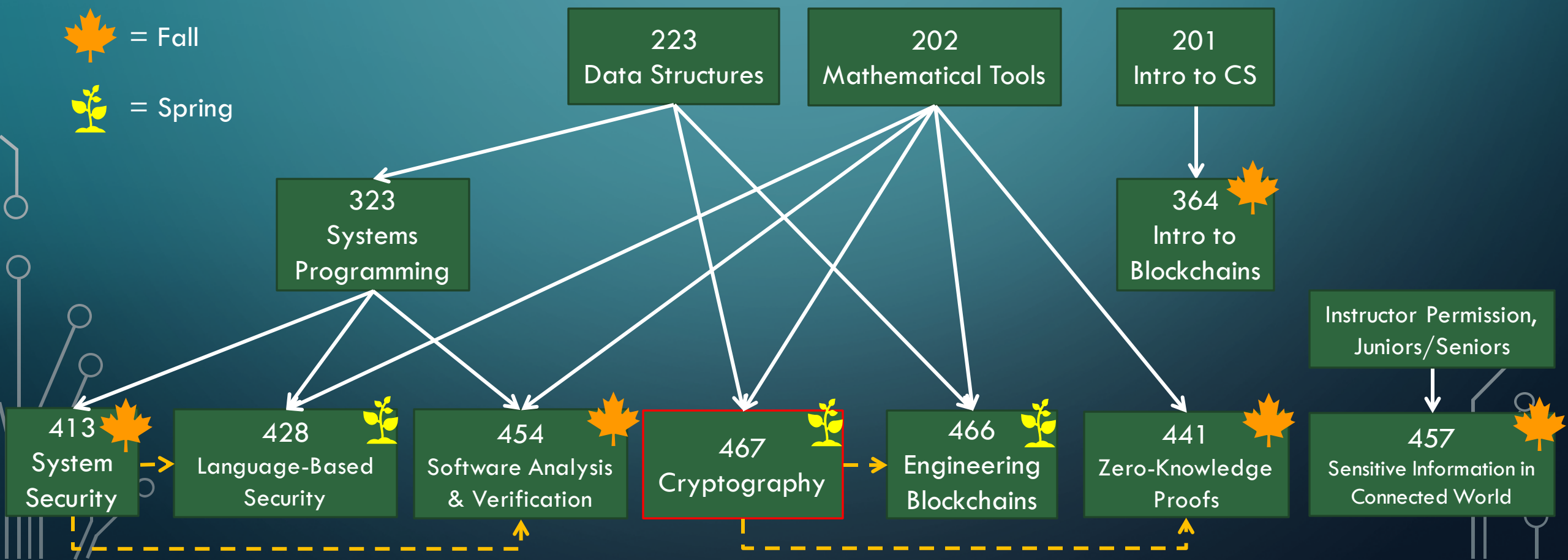


467: Mon/Wed 4-5:15pm

How can we communicate secretly and/or without someone modifying our messages when all communication is public and you've never met the other person before?

- Learn mathematical primitives that build up to cryptographic protocols.
- Prove that these protocols are secure (under certain assumptions).
- See how these protocols are used to solve real world problems.

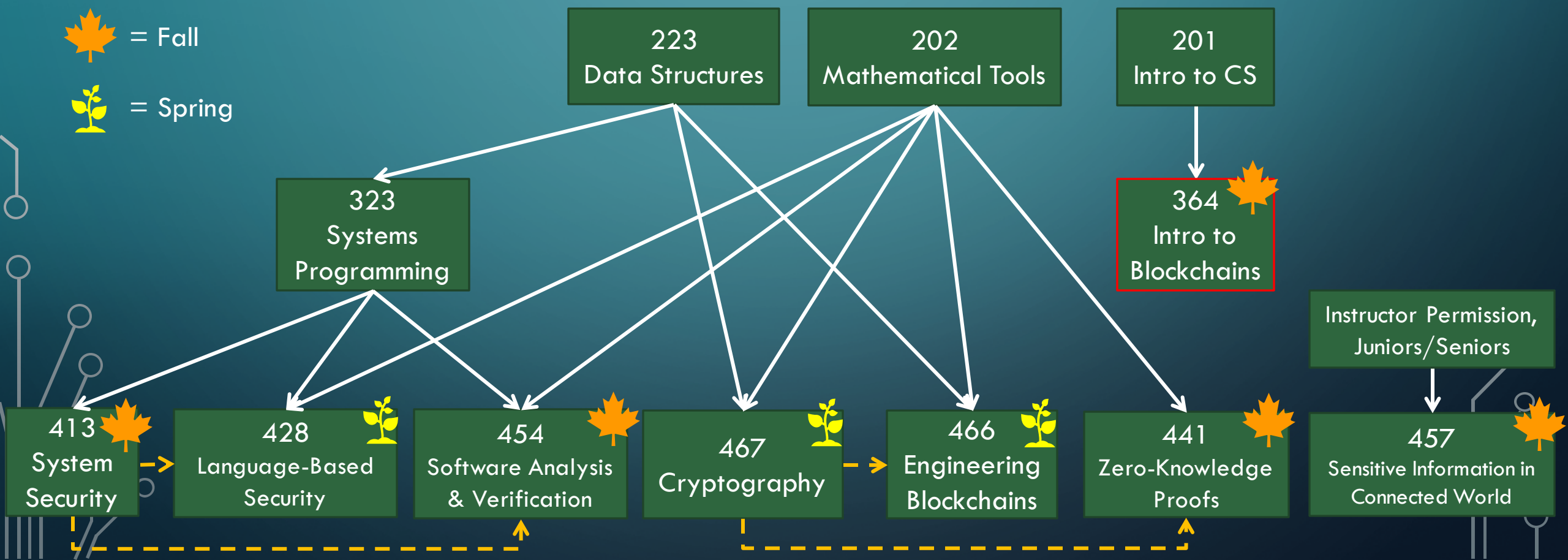
More theory. Pairs well with 413 for a more complete view of the security ecosystem. 413 uses protocols learned in 467, but we can abstract these details away so these courses can be taken in either order.



364:
What a blockchain is, what applications can be built on top of it, and how to program smart contracts securely.

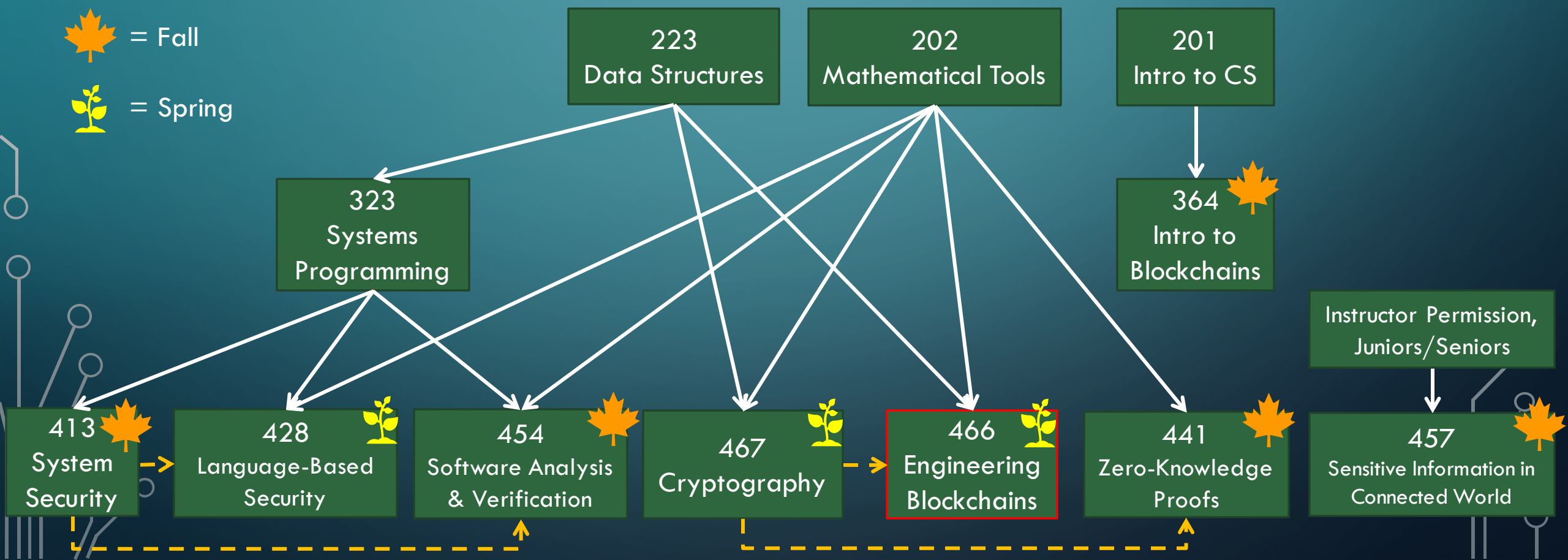
- Technological foundation of the blockchain stack (consensus layer, ordering layer, execution layer, etc.)
- The design of representative applications (cryptocurrencies, smart contracts, Decentralized Finance, etc.)
- The principles for writing secure smart contracts
- Overview of the latest research directions

Few prerequisites and a focus on applications makes this more accessible than alternatives.
If you take this and enjoy it, then you can go deeper with the 400 level courses.



466: Mon/Wed 11:35am-12:50pm
Fundamental building blocks of blockchains and core architectural considerations.
A bottom-up understanding of blockchains as opposed to the top-down approach in 364.
For students who want to get up to speed with the latest technical topics in the industry and is good preparation for those looking to either pursue research or engineering jobs related to blockchains.

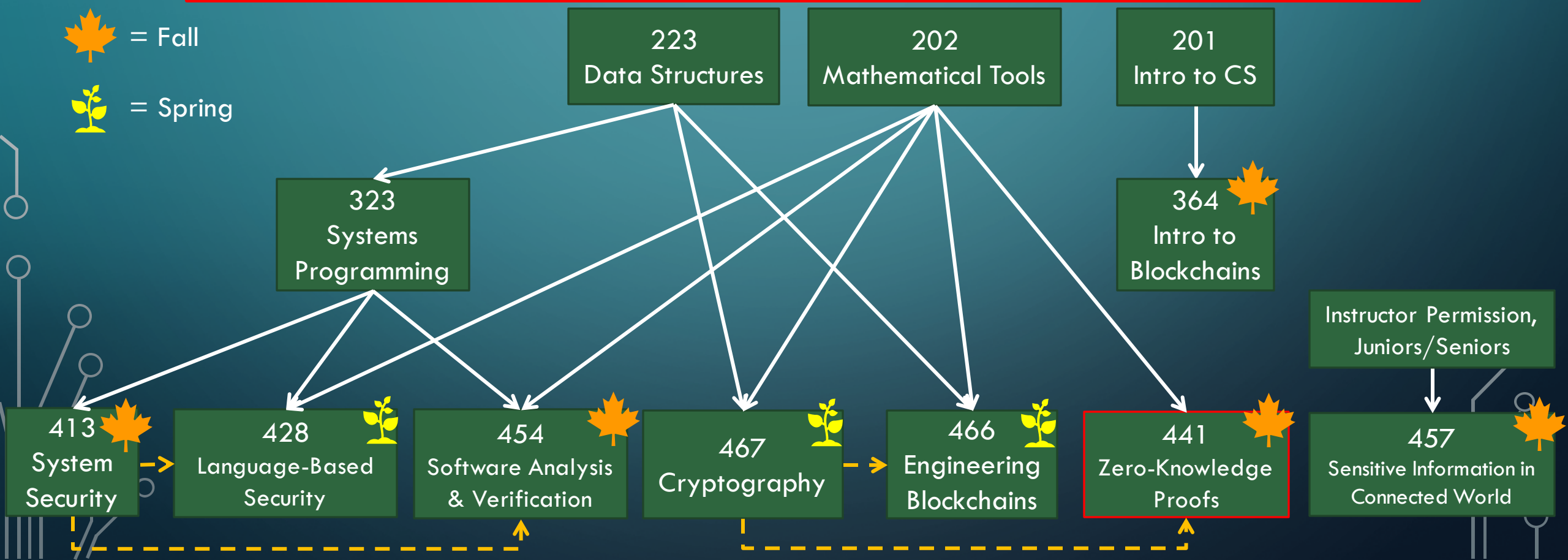
467 is a soft or recommended prerequisite. It can be taken without it, but you will probably get more out of the course with that cryptography background.



441:
How can we do computation in a private way while still proving that we got the correct result?
By using Cryptography!

This class covers cryptographic proof systems. Beyond correctness, a zero-knowledge proof system enables us to prove knowledge of secret information, including hidden inputs to a computation that achieves a certain output. Both types of proof systems have incredible applications to privacy and verifiability in a decentralized web.

467 is not a prerequisite, but it might be helpful to get the foundations of Cryptography this Spring if you are interested In 441 next year.



457:
20-person capped seminar course
Reading and discussing research papers that explore both the power and the limitations of existing privacy-enhancing technologies such as encryption and "trusted platforms."