Spring 2019 – CS Special Topics Courses

1. **CS 494* - Hummel (BioE/CS) –**
   - **This class/section is for CS undergrads only**
2. **CS 494* – Mansky - Software Foundations**
3. **CS 494* – Polakis - Network Security**
4. **CS 494* – Stephens - Cloud Data Center Systems**
5. **CS 494* – Vishwanath - Introduction to High Performance Data-Centric Systems**
6. **CS 594** – Kanich - Advanced Computer Security and Online Privacy**
7. **CS 594 – Kash - Economics in Computer Science**
8. **CS 594 – Parde - Language and Vision**
9. **CS 594 – Sun - Algorithm Methods for Big Data**
10. **CS 594 – Wu - High-Performance NoSQL Databases**
11. **CS 594 – Zuck - Security Foundations**

*CS Undergraduate students must submit a modification of major to use the class as a technical elective.

**CS 594 taught by Chris Kanich in Spring 2019 will be converted to CS 568. So, the class will count as a regular CS 5xx coursework and not as a special topics CS 594 for graduation requirements.

Dept. of Computer Science
University of Illinois at Chicago

Spring 2019 – CS Special Topics Courses

CS 494 – Software Foundations

- Instructor: William Mansky
- Meeting time: MWF 2-250
- CRNs: 42280 (ugrad) & 42281 (grad)

Course Description:
Software as written today inevitably contains bugs: array bounds and memory errors, mishandled corner cases, and logic errors where the program's behavior differs from the programmer's expectations. This course will teach the logical principles and tools that can be used to precisely describe the intended behavior of a program and prove that its actual behavior meets its specification. Students will learn to use the Coq proof assistant to build machine-checked proofs of mathematical properties, both in general and specifically about the behavior of programming languages.

Topics:
The course will use the Software Foundations series of online textbooks, available at https://softwarefoundations.cis.upenn.edu, and in particular the first two volumes, Logical Foundations and Programming Language Foundations. The topics covered will include: interactive theorem proving in Coq, propositional and predicate logic, proof by induction (mathematical, structural, and rule), semantics of programming languages, proving equivalence of programs, Hoare logic and program verification, type systems and type safety, practical program verification. Each class session will be divided into a lecture portion and a lab portion.

Coursework and Grading:
Students will participate in in-class exercises in each session, and there will be outside-of-class proof assignments as well. There will be two in-class midterm exams and a final exam. The tentative grading scheme is as follows:

- Classwork/Homework: 30%
- Midterm 1: 20%
- Midterm 2: 20%
- Final Exam: 30%

Prerequisites:
CS 301 or equivalent. The course will assume familiarity with propositional logic and the basic concepts of mathematical proof, especially proof by induction. Functional programming experience (as in CS 341) is useful but not required. Students who have taken CS 476 may find the course particularly interesting, but no prior experience with programming language semantics will be assumed.

Spring 2019 – CS Special Topics Courses

CS 494 – Network Security

- Instructor: Jason Polakis
- Meeting time: TR 2-315
- CRNs: 42279 (ugrad) & ## (grad)

Course Description:
This course will cover the principles and practice of network security. We will provide an overview of cryptographic foundations, followed by a discussion of network security applications. Given the ubiquitous nature of network communications in modern computing, we will cover a wide range of systems and applications, and the security threats that arise.

Topics:
- Security Principles
- Network Security Infrastructure
- Protocol Security
- Network Security Applications
- Network Defenses
- Advanced Topics

Grading:
- Assignments: 40%
- Exams: 40%
- Reading homework & participation: 20%

Pre-requisite:

Spring 2019 – CS Special Topics Courses

CS 494 – Cloud Data Center Systems

- Instructor: Brent Stephens
- Meeting time: TR 1230-145
- CRNs: 42278 (ugrad) & 42289 (grad)

Course Description:
Cloud data center systems provide services like mail, search, maps, and ridesharing that have become ubiquitous in today's modern life. This class will introduce students to key concepts and state-of-the-art in the design of cloud data center systems. The topics that will be covered include big data systems, batch and streaming analytics, and online latency-sensitive applications like search. In this course, students will get hands on experience with these systems via CloudLab.

Graduate (4 credit hours) students will be expected to form small groups and collaborate on a project related to data center systems.

Topics:
This course will both cover the basics of modern cloud hardware and software infrastructures that these cloud systems leverage and explore the design and implementation of these cloud systems themselves from the ground up. The goal of this course is to survey the entire cloud computing landscape. First, this class will cover big data stacks, including Hadoop, YARN, Spark, and Hive. Next, this class will cover streaming analytics like Spark Streaming, graph processing stacks like GraphProc and GraphX, and search frameworks like Solr. After that, this course will cover serverless platforms like OpenLambda and approximation in BlinkDB. Finally, this class will briefly cover machine learning frameworks like ParamServ and Tensorflow.

This class will consist of four components: 1) lectures on the foundations of cloud data center systems, 2) reading and discussing previous research, 3) homework assignments that utilize CloudLab to cover the practical use of cloud data center systems, and 4) students who enroll in the 4 credit hour version of this class will form small groups and collaborate on a project related to data center systems.

Spring 2019 – CS Special Topics Courses

CS 494 – Introduction to High Performance Data-Centric Systems

- Instructor: Venkatram Vishwanath
- Meeting time: F 2-430
- CRNs: 42287 (ugrad) & 42288 (grad)

Course Description:
Data-centric systems and applications are critical to glean insights in application domains including neuroscience, engineering, material science, biosciences and national security. High Performance computing (HPC) systems have rapidly evolved from solving compute intensive applications to include data-driven applications. The course will provide an overview of data-centric HPC systems, architectures, including storage, and applications. We will cover various requirements and challenges relevant to data-intensive applications. We will discuss how these are being addressed on HPC systems and the recent advances and new directions being pursued in the design of data-centric system architectures and algorithms. The course aims to provide a solid basis to work on the research and development of next generation of data-centric systems, distributed systems and applications.

Topics:
Introduction to High performance computing (HPC) architectures, Storage systems, Programming models and runtimes for many-core systems, GPU-based accelerators and distributed memory systems, Introduction to MPI and SPARK for data-centric systems, Machine learning on HPC systems, Data-parallel deep learning training on HPC systems, Application case studies, including neuroscience, high energy physics, Materials science, Introduction to Neuromorphic computing, Architectures for graph processing.

Tentative Course structure and Grading policy:
10% Class participation
20% Paper/Reading Reviews
20% Paper/Reading Presentations and Survey report (for graduate students) 50% Project

Student Deliverables and Project:
Students will be required to undertake a project of their interest, relevant to any of the general themes of the course. I will plan to have several project ideas and relevant datasets in scientific domains in neuroscience, material science, high energy physics, image processing, etc. The project will be in groups of 2 or 3 students (depending on student enrollment) and will focus on using the HPC cluster systems. The project can either consist of taking one of the topics in the course and doing a deeper exploration with a use-case or by exploring a topic based on their research interests. The project will be required to present an overview and planned approach in Week 7, prototype and early results in Week 11 and a final project presentation in Week 15.

Students are expected to actively participate in the class. They will be expected to have a written report of the readings prior to the class. They are expected to present the readings and participate in the discussions.

Pre-requisites:
Students should have a solid grasp of programming in a language like C / C++ / Python and basic data structures to be able to implement the projects in the course. Ideally, you would have completed a course in data structures (CS251), Machine Organization (CS261), Software design (CS342), system programming (CS361).

Spring 2019 – CS Special Topics Courses

CS 594 – Advanced Computer Security and Online Privacy

- Instructor: Chris Kanich
- Meeting time: TR 330-445
- CRN: 33792

Course Description:
Students will be able to independently understand and evaluate modern scholarship in security and privacy research; understand and communicate to others the key ideas, challenges, and solutions; and identify future research directions. Students will be able to perform original research where computer security and privacy intersects with their own research interests.

Topics:

- Online user tracking
- User deanonymization
- Cybercrime
- Specialized (i.e. IoT, medical, embedded, etc.) device security
- Web security
- Empirical security: Web/Network Measurement
- Cryptography/Blockchain
- Machine learning security
- Usable security

Grading system:

- Participation in in-class discussions: 15%
- In-class presentations of others' scholarsip: 25%
- Final project: 60%

Spring 2019 – CS Special Topics Courses

CS 594 – Economics in Computer Science

- Instructor: Ian Kash (iankash@uic.edu)
- Meeting time: TR 2-315
- CRN: 33649

Course Description:
In this course, students will become broadly familiarized with research topics at the interface between economics and computer science through reading and discussing important papers. The course is self-contained and does not assume prior economics background, so there will also be a lecture component covering the basics of the relevant mathematical techniques from economics. Upon completion of this course, students will have a foundation for further research using economic techniques, both in computer science broadly and as applied to their specific research interests.

Tentative Weekly Schedule:

1 Game Theory I (Equilibria and Equilibrium Computation)
2 Game Theory II (Congestion Games and the Price of Anarchy)
3 Game Theory III (Poker and Security Games)
4 Mechanism Design
5 Dynamic Mechanism Design (Scheduling)
6 Auction Theory (Internet Advertising)
7 Combinatorial Auctions (Spectrum Auctions)
8 Mechanism Design without Money (Fair Division)
9 Matching Markets (School Choice, Kidney Exchanges)
10 Studying Markets in the Wild (Penny auctions, Ride Sharing)
11 Learning in Games I (Best response Dynamics
12 Learning in Games II (No-regret learning)
13 Information Elicitation (Scoring Rules and Prediction Markets)
14 Financial Applications (Blockchain and High Frequency Trading)
15 Privacy and Mechanism Design
16 Final project presentations

Assignments and Grading:
Students will have to read all assigned readings (typically one published paper per class) and be expected to actively participate in class discussions. Students will be required to undertake a project of their interest, relevant to any of the general themes of the course. The project will consist of original research. This can either consist of taking one of the topics in the course and doing a deeper exploration or by exploring a topic based on their research interests where economic techniques can be applied. Grades will be evenly split between the participation component and the project component.

Spring 2019 – CS Special Topics Courses

CS 594 – Language and Vision

- Instructor: Natalie Parde
- Meeting time: TR 1230-145
- CRN: 33648

Course Description:
Researchers in artificial intelligence are increasingly applying multimodal solutions to traditional problems, particularly within the realm of natural language processing (NLP). In particular, synthesizing NLP with computer vision (CV) allows intelligent systems to harness both visual and linguistic information to generate content and derive meaning. This seminar course will introduce students to current research in fundamental language + vision problems, and provide students with a scientific background in relevant application areas. By the end of the course, students will have gained exposure to core concepts through a combination of lectures on fundamental principles, paper discussions, and semester-long projects in student-selected focus areas.

Topics: Grounded Language Learning, Physically Situated Dialogue, Automated Image Captioning and Video Description, Visual Question Answering, Text-to-Image Generation, Visual Story Entailment, and Language Disambiguation via Images.

Grading System:

- 15% - Paper Reviews: Each week that papers are discussed, students will be required to submit a short (~ one page double-spaced) critique of one of the papers. The critique should include a brief summary of the paper, highlights of aspects of the paper that are particularly good or should be improved, an analysis of the soundness of the methodology and evaluation, and an explanation of whether or not the conclusions drawn by the authors are justified.
- 30% (10% each) - Paper Presentations: Each paper discussion will be led by three students: one will provide an overview of the paper, one will argue in favor of the paper, and one will (diplomatically!) argue against the paper. Students will be required to fill each role once.
- 20% (5%, 5%, and 10%, respectively) - Project Presentations: Students will be required to give three short presentations about their semester-long project: a proposal, a mid-semester status update, and a final presentation.
- 15% - Project Write-Up: Students will be required to write a conference/journal style paper about their project, including a literature review, methodology, evaluation, and conclusions.
- 20% - Project: A central component of this course is the semester-long project. Students will complete their projects independently (students working collaboratively

on ongoing research must clearly define separate sub-projects), and the projects must be relevant to the central theme of the course. However, students will otherwise be afforded considerable flexibility in selecting their projects. Ideally, for thesis or dissertation students, the work resulting from this course will be able to be incorporated in some way into their thesis or dissertation research.

Pre-requisite: Enrollment upon consent of the instructor

Spring 2019 – CS Special Topics Courses

CS 594 – Algorithm Methods for Big Data

- Instructor: Xiaorui Sun
- Meeting time: TR 11-1215
- CRN: 42282

Course Overview:
Recently there has been a lot of glorious hullabaloo about Big Data and how it is going to revolutionize the way of computation. As part of the general excitement, it has become clear that for truly massive datasets and digital objects of various sorts, even algorithms which run in linear time (linear in the size of the relevant dataset or object) may be much too slow.

Traditional models and paradigms of sequential computation has become obsolete. Different computation scenarios ask for different trade-offs between solution quality, space usage, running time, and communication. These paradigms are formalized as sublinear, streaming, parallel, and other distributed algorithmic models of computation. New algorithm techniques within these models have been developed, and have played an important role in the modern data analysis.

This course aims at timely dissemination of foundational algorithmic developments for big data analysis and exposing students to cutting edge research in this area. The course will involve deep theoretical analysis with the goal of developing practical algorithms for a variety of applications.

Pre-requisite:
The course will be accessible to students with a wide range of backgrounds, including both theoretical and applied areas of computer science, as well as mathematics. Some familiarity with algorithms (CS 401-level) will be assumed.

Spring 2019 – CS Special Topics Courses

CS 594 – High-Performance NoSQL Databases

- Instructor: Xingbo Wu
- Meeting time: MW 430-545
- CRN: 34724

Course Description:
NoSQL (“not-only-sql”) databases are being designed and deployed for demanding applications. Different from the “one size fits all” design principle rooted in SQL databases, each NoSQL database is tailored and optimized to provide a minimum set of functionalities with maximum efficiency, with certain capabilities on scalability, consistency, persistency, etc.. In this course, students will become familiar with the designs principles and core techniques of real-world NoSQL databases.

Topics:

- Index and storage structures: hash-based and ordered indexes, concurrency control, logging & journaling, COW.
- Hardware/OS interfaces: network I/O (polling, multiplexing, and kernel-bypassing), file system and storage interactions (direct I/O, asynchronous I/O, file-system bypassing, multi-level caching), and in-kernel designs (kernel-level KV cache).
- Distributed designs: consistent hashing, distributed transaction processing, load balancing, fault-tolerance (consensus protocols, replication, erasure code).

The class will involve regular lectures (~7 weeks), paper presentation and discussions (~8 weeks), and two or three programming projects.

Grading:
5% Class Participation
15% homeworks
40% paper presentation
40% programming projects
Grading:

Pre-requisite: CS 251, CS 361

Spring 2019 – CS Special Topics Courses

CS 594 – Security Foundations

- Instructor: Lenore Zuck
- Meeting time: TR 5-615
- CRN: 38551

Course Description:
Foundations course in security: Introduction to cryptography, public-key crypto, Encryption/Decryption/Digital signatures using RSA and Diffie-Hellman, encryption and digital signature, Dolev-Yao model, authentication protocols, access control, E-voting, Crypto currency, Privacy, and cloud security.

Topics:

- Intro to Cryptography
- RSA
- Dolev Yao Model
- Diffie Hellman Model
- Authentication
- Access Control
- E-voting
- Crypto Currency
- Privacy
- Cloud Security
- Grading:

Pre-requisite:
CS301 or similar, CS401 or similar or consent of instructor