Fall 2021 – CS Special Topics Courses

1. **CS 394* – Buy – Framework-based Software Development for Hand-held Devices**
2. **CS 394* – Mobasheri – Advanced Data Structure Practicum**
3. **CS 494* – Caragea –**
4. **CS 494* – Esmailbeigi (BioE)/Hummel (CS) – Wearable Technologies**
5. **CS 494* – Miao – Introduction to Cryptography**
6. **CS 594 – Michaelis – Social Robotics and Human-Robot Interaction**
7. **CS 594 – Miranda – Big Data Visualization and Analytics**
8. **CS 594 – Sidiropoulos – Foundations of Blockchains**

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

Fall 2021 – CS Special Topics Courses

CS 394 – Framework-based Software Development for Hand-held Devices

- Instructor: Ugo Buy
- Meeting time: TR 3:30-4:45pm (on-campus)
- CRN: 46370 (undergraduate only)

Course Description: Multi-platform app development using software frameworks; the integration of framework-generated code with native code of mobile operating systems; numerous programming assignments in multiple programming languages.

Prerequisites: CS 342

Topics: Mobile app development with a platform-independent framework; the framework's programming language; deployment and maintenance of framework-generated code on iOS and Android platforms.

Student responsibilities: 1 or 2 homeworks, 5 projects, 7 quizzes, a final exam (subject to change)

Fall 2021 – CS Special Topics Courses

CS 394– Advanced Data Structure Practicum

- Instructor: Nasim Mobasheri
- Meeting time: TR 5:00-6:15pm (on campus)
- CRN: 46371 (undergraduate only)

Course Description:
This course focuses on the design and implementation details of non-trivial algorithms. We will cover complex data structures with an emphasis on amortized analysis and implementation, as well as related non-trivial algorithms that use the same data structures in their implementation.

Topics:
Trees: { AVL, RedBlack trees, Tries}, Heaps: {Binary, Binomial, Fibonacci}, Disjoint Sets, String algorithms, Bitwise manipulation, Graph algorithms: {Traversal, shortest path, MST,...}, Dynamic Programming, Network Flow: {Bipartite matching, Min/Max cut}, Computational geometry, Cross product, convex Hull, plane intersections

Coursework (subject to change): biweekly whiteboard interview session (20%), 4 - 6. programming projects (30%), midterm (20%), final (30%)

Prerequisite: CS 251

Back to List

Fall 2021 – CS Special Topics Courses

CS 494–

- Instructor: Cornelia Caragea
- Meeting time: M 3:00-5:30pm (on campus)
- CRN: 44014/44015

Back to List

Fall 2021 – CS Special Topics Courses

CS 494– Wearable Technologies

- Instructor: Hananeh Esmailbeigi (BioE) and Joe Hummel (CS)
- Meeting time: M 2:00-4:50pm (on campus)
- CRN: 43965 (undergraduate only)

Course Description:
This course, taught primarily by BioE, is a lab-based course in the design and construction of wearable devices. Students work in teams of 4, 2 from CS and 2 from BioE, and use HW and SW skills to design, construct, and analyze wearable devices and the collected data. The class is run like a true lab, where class meets once/week for 3 hours, and class time is devoted mostly to project work. If necessary, projects are completed outside of class time.

Coursework:
3 assigned projects + a final project of team's own design. Each project involves HW and SW integration, some circuit design, data analysis, and a team report. Team work required.

Prerequisites:
CS 251 and 261, and CS 362 if possible.

Fall 2021 – CS Special Topics Courses

CS 494 - Introduction to Cryptography

- Instructor: Peihan Miao
- Meeting time: TR 2-3:15pm
- CRN: 45500/45501

Course Description:
Cryptography plays a central role in protecting information in computer systems, achieving various objectives including confidentiality, data integrity, and authentication. It is a widely used tool in communications, computer networks, and more generally computer security. This course is an introduction to modern cryptography from both foundational and practical perspectives.

We will present definitions and constructions of various foundational cryptographic primitives and algorithms including encryption schemes, hash functions, message authentication codes, digital signatures, and so on. Besides the theoretical foundations, we will also study various practical cryptosystems such as stream cipher, block cipher, and SHA. More advanced topics will also be covered such as zero-knowledge proofs, secure computation, blockchain and cryptocurrency.

Grading:

- Regular attendance and active participation in class 20%
- Biweekly homework assignments 30%
- Take-home midterm 20%
- Final exam 30%

Prerequisites:
(a) CS 251, and (b) STAT 381 or IE 342 or STAT 401.
CS 401 is highly recommended.

Fall 2021 – CS Special Topics Courses

CS 594 - Social Robotics and Human-Robot Interaction

- Instructor: Joseph Michaelis
- Meeting Time: W 3:00-5:30
- CRN: 27441
- Textbook: Human-Robot Interaction (Bartneck et al. 2020) – free and open source!

Course Description:
The field of social robotics is growing rapidly as robots with socially interactive capacities are being implemented as collaborative robotics in industrial manufacturing, consumer assistants in retail settings, healthcare aids in hospitals and outpatient care facilities, and as in-home companions and assistants. The growing demand for social robots in these fields has increased the demand for a workforce capable of designing human-robot interactions that incorporate the needs and workflows of stakeholders in each of these areas.
Throughout, we will focus on designing, creating, and testing *social* robots, to concentrate the discussion on human perception of robots in social ways, interacting with humans in socially adept and responsive ways, and testing and measuring the social experience in relationship to other outcomes (e.g. learning or following directions). This emphasis will require additional readings beyond the HRI book to supplement the course with findings and theory from social psychology, learning sciences, and social robotics.
This course will introduce students to the research methods, iterative design processes, and theory on human-robot interaction as it applies to social robotics and guide them through developing their own social robotic interaction model, programming a real or simulated robot to enact that model and design a research study to explore the human-robot interactions the robot was designed for. The combination of design, technical skills and research methods in HRI that you will engage with are applicable to many computer science, engineering, learning sciences and design sciences fields that consider a human-centered approach to technology design.

Course Outline:
Typically, each class session will include a reading discussion follow by an applied workshop, where students will spend time on guided project work or technical tutorials. The course will include two major themes: design and research. We will begin with the **research** theme to focus students on developing a knowledge of methods and research questions in the field to focus their robot design for a specific research question. Next the **design** theme will focus on generating design principles from theory and user input from a user centered design perspective, developing an interaction model (or flow-chart) based on those principles, and creating a wireframed or other low fidelity visualization of the interaction. Throughout the course, we will also support students in the **creating** theme will focus on implementing the interaction model as a functioning interactive prototype.

Course Topics:

- Intro to HRI research
- Research Questions and Methods Intro

Fall 2021 – CS Special Topics Courses

- Mode of Interaction + Measures
- Research Design Workshop
- Intro to HRI design
- Modes of Interaction
- Verbal and Non-verbal Interaction
- Applications and Ethic in Social Robots
- Design Workshop
- Robot Hardware, Sensors and Actuators
- Robotics Programming
- WeBots Programming*
- WeBots Languages*
- Programming Workshop/Lab

* If in-person and materials allow, students can use physical robots – including Misty II, Temi, NAO, Cue, and iRobot

Programming Requirements:
Programming requirement will be based on each student's prior experience and which robot is used. Most robots have robust Software Development Kits and APIs that serve as supports for programming.
For this course, we will use either physical robots (Misty, Temi, Cue, iRobot) or a suite of robot simulation tools through the Webots (https://cyberbotics.com) open-source robot simulator for students to program and test their prototype robots using C, C++, Java, Python, MATLAB, or ROS, and can use one of many existing templates to begin their work. There is also a robust training manual and tutorial set for the Webots environment (https://cyberbotics.com/doc/guide).

Course Work:
The course will be structured as a project-based curriculum where students will have deliverables due at three points in the course as summative work for each theme. All pieces will be built together in a single document to create a project document that can serve as a fully articulated research study proposal.

1. 3-page HRI research study proposal.
2. Interaction model, annotated with theoretical and user centered design principles, and a low fidelity prototype.
3. Final project simulated or physical robot prototype, with programming and demonstration that enacts their design.

Fall 2021 – CS Special Topics Courses

CS 594– Big Data Visualization and Analytics

- Instructor: Fabio Miranda
- Meeting time: MW 9:30-10:45am (on campus)
- CRN: 43915

Course rationale:

In the past decade, technological innovations have enabled the automatic collection of a diverse set of very large datasets, from user-generated content, such as tweets, to sensor data, such as noise decibel level or aerial surveys. As pointed out by Hal Varian, Google's Chief Economist, today data is not only free but ubiquitous, and the ability to understand, process, extract value from, and visualize it, is going to be hugely important in the next decades. As data grows in size and complexity, however, siloed solutions become less capable of conveying all the structure and information hidden in the data. Visual analytics systems have been successful at enabling users to obtain insights from large data [1,2]; Well-designed large-scale data visual analytics systems merge domain expertise and analytics techniques, enabling users to gain actionable insights into real-world problems by formulating and testing hypotheses, and to address domain-specific challenges.

This course will introduce students to big data visualization and analytics, focusing on the connections between visualization, data management, and data mining [3]. We will cover potential sources of big data, including audio, image, graph, geometry, and mobile data, both structured or unstructured. The course will discuss effective metaphors to visualize data that might span billions of records; data mining and machine learning techniques that can summarize the data, or extract patterns to guide users in their visual exploration; and the data management techniques that can effectively drive interactivity in the visual exploration of big data.

Course objectives:

This course will focus on the intersection between visualization, data management, and data mining, covering the necessary topics to build visual analytics tools to handle big data. For the purposes of this course, big data will include any dataset with size (or complexity) that goes beyond the ability of standard tools and techniques to interactively manage and process it with a latency below 0.5 second [4].

At the end of the course, students will have a good understanding of the state of the art in visual analytics of large data, limitations of current techniques, and promising research directions. They will also be able to design and implement visual analytics systems capable of handling large data, by combining visualization and data mining techniques, data structures and algorithms that work in tandem to enable interactive data exploration and tackle real-world problems.

Fall 2021 – CS Special Topics Courses

Topics (tentative):
Visualization
Uncertainty vis
Progressive vis
Data management
Data structures for vis, including data cubes
Spatial queries using GPUs
Approximate queries, hashing schemes, and learned indices
Data analytics
Computational topology
Wavelet-based techniques
Incremental analytics
Domain-specific applications and tools

Assignments and evaluation:
Students will be required to complete 2 to 3 assignments, where they will implement concepts and techniques presented in class using large datasets, from high-resolution timeseries to image data. There will be a final assignment where students will have to identify a research problem under my guidance, and propose and implement a small-scale system for the interactive exploration of a large dataset (or multiple datasets). Students will be able to use external datasets, or select from a list of available datasets, including spatiotemporal audio, image, graph, and geometry data. Students will have to produce a short paper reporting preliminary results, and give a brief final presentation in class.

Prerequisites:
Working knowledge of programming languages, preferably C/C++, Java, or Python.

Method of instruction:
For the first ~10 weeks, the course will be lecture based (75 minutes twice per week), followed by ~5 weeks of paper presentations and discussions.

Readings:
There is no required textbook. The vast majority of the coursework will be based on research papers from conferences and journals, including visualization (VIS, EuroVis, IEEE TVCG), database (SIGMOD, VLDB, IEEE Big Data), and related fields (IEEE TITS, AAAI).

[1] H. Doraiswamy, J. Freire, M. Lage, F. Miranda, and C. Silva, "Spatio-Temporal Urban Data Analysis: A Visual Analytics Perspective," IEEE Computer Graphics and Applications, 38(5):26-35, 2018

[2] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, "Visual exploration of big spatio-temporal urban data: A study of New York City taxi trips," IEEE Transactions on Visualization and Computer

Fall 2021 – CS Special Topics Courses

Graphics, 19(12):2149–2158, 2013

[3] R. Chang, J. D. Fekete, J. Freire, and C. E. Scheidegger, "Connecting visualization and data management research," Dagstuhl Reports, 7(11):46–58, 2017

[4] Z. Liu and J. Heer, "The effects of interactive latency on exploratory visual analysis," IEEE Transactions on Visualization and Computer Graphics, 20(12):2122–2131, 2014

Back to List

Fall 2021 – CS Special Topics Courses

CS 594– Foundations of Blockchains

- Instructor: Anastasios Sidiropoulos
- Meeting time: TR 11:00a-12:15pm (on campus)
- CRN: 43136

Method of instruction: The instruction will be based on the following main components:

- During the first half of the course, the instructor will present various fundamental methods and ideas used in the design of blockchains, cryptocurrencies, and related objects. Any necessary prerequisites will also be discussed during this time.
- During the second half of the course, the students will read and present research papers.
- The students will work on a project of their interest that incorporates ideas discussed in the class. The students will have the option to either conduct original research or experimentally evaluate prior work. The project will be performed in teams of 1–3 students. The students will be encouraged to start thinking about possible research topics early in the semester. The instructor will hold frequent meetings with each team to guide their progress.

Narrative description: A blockchain is a tamperproof sequence of data that can be read and augmented by everyone. The first widely used implementation of such a structure, proposed by Satoshi Nakamoto, is the Bitcoin protocol. Blockchains have found numerous applications, such as cryptocurrencies and smart contracts, and hold the potential to revolutionize the way a democratic society operates.

From the scientific point of view, blockchains present several new exciting opportunities, as well as technical challenges. The intellectual underpinnings of the design of such public ledgers lie in the intersection of cryptography, distributed computing, algorithms, game theory, and economics. This unique combination of diverse scientific disciplines necessitates the development of new theoretical foundations for this emerging area. Consequently, a new intellectual framework has started emerging for reasoning about public ledgers, their powers and limitations.

Goal: In this course, the students will be exposed to the theoretical foundations underpinning the design and operation of blockchains. Emphasis will be given on understanding how the properties of blockchains lead to several other important primitives, such as cryptocurrencies, smart contracts, digital assets, and so on. Furthermore, the students will learn about important technical advances, such as scaling, transaction routing, energy consumption, and so on.

Student deliverables: The students will have to read all the papers, and they will be expected to actively participate in all the lectures. Furthermore, each student will present at least one research paper to the class. For the final project, the students will have to submit a proposal of their selected topic within the first half of the course, a final report at the end of the class, and they will be asked to give a brief presentation on their findings.

Back to List

Fall 2021 – CS Special Topics Courses

Class meetings: There will be two 75’ meetings per week.

Prerequisites: The course will be accessible to students with a wide range of backgrounds, including both theoretical and applied areas of computer science. Some familiarity with discrete math (equivalent to CS 201), algorithms (equivalent to CS 401) and computability theory (equivalent to CS 301) will be assumed. All necessary cryptographic primitives (elements of public key cryptog- raphy, zero knowledge proofs, elliptic curve cryptography, and so on) will be introduced during the course.

Exams: There will be no exams.

Readings: Selected books and research papers from the following tentative list of topics:

Byzantine Agreement: What is the Byzantine generals problem? How does Nakamoto consensus solve the token distribution problem?

Game-theoretic aspects of blockchains: Is the Bitcoin protocol incentive-compatible? How can we mathematically analyze chain forks?

The Bitcoin protocol and its extensions: What is the Bitcoin Backbone protocol? Can we verify parts of a blockchain without reading the whole list of blocks? What are Non-interactive Proofs of Proof of Work?

Network-theoretic aspects of blockchains: What is a peer-to-peer network? What is an eclipse attack? How does this affect the security of cryptocurrencies?

Energy consumption: The Nakamoto consensus is based on a “proof of work” algorithm, which uses an enormous amount of energy. Several other “proof of stake” protocols have been proposed that try to mitigate this issue (Algorand, Fruitchains, Ouroboros, etc). How do these protocols work?

Scalability: The Bitcoin protocol can support only a limited number of transactions per second. Recent theoretical work suggests that this is an inherent limitation of blockchains. In order to bypass this obstacle, various other, so-called “second-layer” algorithms and protocols have been proposed. Some of these proposed solutions include the lightning network, plasma, rollups, side-chains, and so on. How do these work?

Turing-completeness: The Bitcoin protocol supports only a limited number of types of transactions, specified as the Bitcoin script language. This language is powerful enough to program various interesting primitives, such as inter-blockchain transactions, the lightning network, and so on. However, this language is not Turing-complete. Many other blockchains have been created (e.g., Ethereum, Cardano, and so on) that provide a Turing-complete set of transactions, and can be used to implement arbitrary mechanisms, such as voting, governance, public registries, and so on.

Fall 2021 – CS Special Topics Courses

Economic aspects of blockchains: The construction of blockchains with Turing-complete languages has lead to the implementation of a plethora of financial primitives that operate without a central authority. These include: decentralized autonomous organizations (DAOs), algorithmic stable coins, automated marked makers, trustless loans, and so on.

Beyond blockchains: Motivated by the success of applications that run on a single blockchain, researchers have proposed various mechanisms for transferring value and state across different blockchains. These include, Polkadot, Chainlink, and so on. How do these systems work? What are the underlying theoretical guarantees of such systems?

Sample of relevant papers:

- Xi Chen, Christos Papadimitriou, Tim Roughgarden, *An Axiomatic Approach to Block Rewards.*
- J Garay, A Kiayias, N Leonardos, *The bitcoin backbone protocol: Analysis and applications.*
- A Kiayias, A Russell, B David, R Oliynykov, *Ouroboros: A provably secure proof-of-stake blockchain protocol.*
- A Kiayias, A Miller, D Zindros, *Non-interactive proofs of proof-of-work.*
- J Chen, S Micali, *Algorand.*
- Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, *Algorand: Scaling Byzantine Agreements for Cryptocurrencies.*
- I Eyal and E G Sirer. *Majority is not Enough: Bitcoin Mining is Vulnerable.*
- S Tochner, A Zohar. *How to Pick Your Friends - A Game Theoretic Approach to P2P Overlay Construction.*
- Lewis Gudgeon, Sam Maximilian Werner, Daniel Perez, William J. Knottenbelt. *DeFi Protocols for Loanable Funds: Interest Rates, Liquidity and Market Efficiency.*