

# CSDS (CSDS)

---

## Courses

### CSDS 132. Introduction to Programming in Java. 3 Units.

An introduction to modern programming language features, computer programming and algorithmic problem solving with an emphasis on the Java language. Computers and code compilation; conditional statements, subprograms, loops, methods; object-oriented design, inheritance and polymorphism, abstract classes and interfaces; types, type systems, generic types, abstract data types, strings, arrays, linked lists; software development, modular code design, unit testing; strings, text and file I/O; GUI components, GUI event handling; threads; comparison of Java to C, C++, and C#. Offered as CSDS 132 and ECSE 132. Counts for CAS Quantitative Reasoning Requirement.

### CSDS 133. Introduction to Data Science and Engineering for Majors. 3 Units.

This course is an introduction to data science and analytics. In the first half of the course, students will develop a basic understanding of how to manipulate, analyze and visualize large data in a distributed computing environment, with an appreciation of open source development, security and privacy issues. Case studies and team project assignments in the second half of the course will be used to implement the ideas. Topics covered will include: Overview of large scale parallel and distributed (cloud) computing; file systems and file i/o; open source coding and distributed versioning, data query and retrieval; basic data analysis; visualization; data security, privacy and provenance. Prereq: ENGR 131 or EECS 132.

### CSDS 233. Introduction to Data Structures. 4 Units.

Different representations of data: lists, stacks and queues, trees, graphs, and files. Manipulation of data: searching and sorting, hashing, recursion and higher order functions. Abstract data types, templating, and the separation of interface and implementation. Introduction to asymptotic analysis. The Java language is used to illustrate the concepts and as an implementation vehicle throughout the course. Offered as CSDS 233 and ECSE 233. Prereq: CSDS 132 or ECSE 132.

### CSDS 234. Structured and Unstructured Data. 3 Units.

This course is an introduction to types of data and their representation, storage, processing and analysis. The course has three parts. In the first part of the course, students will develop a basic understanding and the ability to represent, store, process and analyze structured data. Structured data include catalogs, records, tables, logs, etc., with a fixed dimension and well-defined meaning for each data point. Suitable representation and storage mechanisms include lists and arrays. Relevant techniques include keys, hashes, stacks, queues and trees. In the second part of the course, students will develop a basic understanding and the ability to represent, store, process and analyze semi-structured data. Semi-structured data include texts, web pages and networks, without a dimension and structure, but with well-defined meaning for each data point. Suitable representation and storage mechanisms include trees, graphs and RDF triples. Relevant techniques include XML, YAML, JSON, parsing, annotation, language processing. In the third part of the course, students will develop a basic understanding and the ability to represent, store, process and analyze unstructured data. Unstructured data include images, video, and time series data, without neither a fixed dimension and structure, nor well-defined meaning for individual data points. Suitable representation and storage mechanisms include large matrices, EDF, DICOM. Relevant techniques include feature extraction, segmentation, clustering, rendering, indexing, and visualization. Prereq: CSDS 133.

### CSDS 236. Introduction to C/C++ Programming. 1 Unit.

This course provides an introduction to C and C++ programming languages. Prior knowledge of these languages is not expected. However, students are expected to know at least one programming language prior to the class. The lectures will cover low-level programming techniques and object-oriented class design. Students will learn specific topics in C/C++ including required background knowledge, pointers & references, memory management, inheritance, polymorphism, templates, exceptions, use of debugging and compiler tools, and others as time permits. Prereq: CSDS 132.

### CSDS 275. Fundamentals of Robotics. 4 Units.

The Fundamentals of Robotics course will expose students to fundamental principles of robotics. Students will explore high level conceptual foundations of robotics beginning with Braitenberg vehicles and apply this knowledge to simulated and physical robot hardware in laboratory experiences and in a final project. Laboratory experiences will guide students through applying theory to practice increasingly complex tasks in a project oriented, group work environment. The course culminates in a robotics challenge project at the end of the semester. Topics covered are: sensors, actuators, kinematics, control, planning and programming. Programming languages and concepts (e.g., C++, object oriented programming) used in robotics will be introduced and used with modern robotics programming toolboxes and frameworks. Prior experience with these languages will not be necessary. Previous experience with robotics is not required for this course. Offered as CSDS 275 and ECSE 275. Prereq: (ENGR 131 or EECS 132) and PHYS 121 and MATH 121.

### CSDS 281. Logic Design and Computer Organization. 4 Units.

Fundamentals of digital systems in terms of both computer organization and logic level design. Organization of digital computers; information representation; boolean algebra; analysis and synthesis of combinational and sequential circuits; datapaths and register transfers; instruction sets and assembly language; input/output and communication; memory. Offered as CSDS 281 and ECSE 281. Prereq: ENGR 131 or CSDS 132.

### CSDS 285. Linux Tools and Scripting. 3 Units.

A fast paced introduction to linux that brings the project-oriented student from little or no linux experience to be fully capable of using scripting languages as tools in a linux command-line environment. This includes systems administration and agile web services with server-side scripting and server-side management. This also involves data cleaning, scraping, wrangling, and processing user data with regular expressions. Current main languages include php, bash, awk, with some client-side javascript, and very brief looks at tcl/lua, powershell, and python. Tools include vim, make, grep/sed, and curl/wget. Recommended preparation: one year of computer programming and familiarity with raw HTML/CSS. Prereq: CSDS 132.

**CSDS 290. Introduction to Computer Game Design and Implementation. 3 Units.**

This class begins with an examination of the history of video games and of game design. Games will be examined in a systems context to understand gaming and game design fundamentals. Various topics relating directly to the implementation of computer games will be introduced including graphics, animation, artificial intelligence, user interfaces, the simulation of motion, sound generation, and networking. Extensive study of past and current computer games will be used to illustrate course concepts. Individual and group projects will be used throughout the semester to motivate, illustrate and demonstrate the course concepts and ideas. Group game development and implementation projects will culminate in classroom presentation and evaluation. Offered as CSDS 290 and ECSE 290. Prereq: EECS 132 or CSDS 132.

**CSDS 293. Software Craftsmanship. 4 Units.**

A course to improve programming skills, software quality, and the software development process. Software design; Version control; Control issues and routines; Pseudo-code programming process and developer testing; Defensive programming; Classes; Debugging; Self-documenting code; Refactoring. Offered as CSDS 293 and CSDS 293N. Prereq: Computer Science Major and (CSDS 233 or ECSE 233 or EECS 233) with a C or higher.

**CSDS 293N. Software Craftsmanship. 4 Units.**

A course to improve programming skills, software quality, and the software development process. Software design; Version control; Control issues and routines; Pseudo-code programming process and developer testing; Defensive programming; Classes; Debugging; Self-documenting code; Refactoring. Offered as CSDS 293 and CSDS 293N. Prereq: (CSDS 233 or ECSE 233 or EECS 233) with a C or higher.

**CSDS 296. Independent Projects. 1 - 3 Units.**

Independent projects in Computer and Data Science. Recommended preparation: ENGR 131 or EECS/CSDS/ECSE 132. Prereq: Limited to freshmen and sophomore students.

**CSDS 297. Special Topics. 1 - 3 Units.**

Special topics in Computer and Data Science. Prereq: Limited to freshmen and sophomores.

**CSDS 302. Discrete Mathematics. 3 Units.**

A general introduction to basic mathematical terminology and the techniques of abstract mathematics in the context of discrete mathematics. Topics introduced are mathematical reasoning, Boolean connectives, deduction, mathematical induction, sets, functions and relations, algorithms, graphs, combinatorial reasoning. Offered as CSDS 302, ECSE 302 and MATH 304. Prereq: MATH 122 or MATH 124 or MATH 126.

**CSDS 305. Files, Indexes and Access Structures for Big Data. 3 Units.**

Database management become a central component of a modern computing environment, and, as a result, knowledge about database systems has become an essential part of education in computer science and data science. This course is an introduction to the nature and purpose of database systems, fundamental concepts for designing, implementing and querying a database and database architectures. Objectives: -An expert knowledge of basic data structures, basic searching, sorting, methods, algorithm techniques, (such as greedy and divide and conquer) -In-depth knowledge on Search and Index Structures for large, heterogeneous data including multidimensional data, high dimensional data and data in metric spaces (e.g., sequences, images), on different search methods (e.g. similarity searching, partial match, exact match), and on dimensionality reduction techniques. Prereq: CSDS 234 or ECSE 233.

**CSDS 310. Algorithms. 3 Units.**

Fundamentals in algorithm design and analysis. Loop invariants, asymptotic notation, recurrence relations, sorting algorithms, divide-and-conquer, dynamic programming, greedy algorithms, basic graph algorithms. Offered as CSDS 310 and CSDS 310N. Prereq: (Computer Science Major/Minor or Data Science Major or Biomedical Engineering Major on the Computing and Analysis Track) and (CSDS 302 or ECSE 302 or MATH 304 with a C or higher) and (CSDS 233 or ECSE 233 with a C or higher).

**CSDS 310N. Algorithms. 3 Units.**

Fundamentals in algorithm design and analysis. Loop invariants, asymptotic notation, recurrence relations, sorting algorithms, divide-and-conquer, dynamic programming, greedy algorithms, basic graph algorithms. Offered as CSDS 310 and CSDS 310N. Prereq: (CSDS 302 or ECSE 302 or MATH 304) with a C or higher and (CSDS 233 or ECSE 233) with a C or higher.

**CSDS 312. Introduction to Data Science Systems. 3 Units.**

An introduction to the software and hardware architecture of data science systems, with an emphasis on Operating Systems and Computer Architecture that are relevant to Data Sciences systems. At the end of the course, the student should understand the principles and architecture of storage systems, file systems (especially, HDFS), memory hierarchy, and GPU. The student should have carried out projects in these areas, and should be able to critically compare various design decisions in terms of capability and performance. Prereq: CSDS 132.

**CSDS 313. Introduction to Data Analysis. 3 Units.**

In this class we will give a broad overview of data analysis techniques, covering techniques from data mining, machine learning and signal processing. Students will also learn about probabilistic representations, how to conduct an empirical study and support empirical hypotheses through statistical tests, and visualize the results. Course objectives: -Expose students to different analysis approaches. -Understand probabilistic representations and inference mechanisms. -Understand how to create empirical hypotheses and how to test them. Prereq: EECS 340 and DSCI 234.

**CSDS 314. Computer Architecture. 3 Units.**

This course provides students the opportunity to study and evaluate a modern computer architecture design. The course covers topics in fundamentals of computer design, performance, cost, instruction set design, processor implementation, control unit, pipelining, communication and network, memory hierarchy, computer arithmetic, input-output, and an introduction to RISC and super-scalar processors. Offered as CSDS 314 and ECSE 314. Prereq: EECS 281.

**CSDS 315. Scalable Parallel Data Analysis. 3 Units.**

This course provides an introduction to scalable and parallel data analysis using the most common frameworks and programming tools in the age of big data. Covered topics include parallel programming models, parallel hardware architectures, multi-threaded, multi-core programming, cluster computing and GPU programming. The course is designed to provide a heavily hands-on experience with several programming assignments. Prereq: CSDS 312.

**CSDS 325. Computer Networks I. 3 Units.**

An introduction to computer networks and the Internet. Applications: http, ftp, e-mail, DNS, socket programming. Transport: UDP, TCP, reliable data transfer, and congestion control. Network layer: IP, routing, and NAT. Link layer: taxonomy, Ethernet, 802.11. Offered as CSDS 325 and CSDS 325N. Prereq: Computer Science Major with minimum Junior Standing and (CSDS 233 or ECSE 233 or EECS 233) with a C or higher.

**CSDS 325N. Computer Networks I. 3 Units.**

An introduction to computer networks and the Internet. Applications: http, ftp, e-mail, DNS, socket programming. Transport: UDP, TCP, reliable data transfer, and congestion control. Network layer: IP, routing, and NAT. Link layer: taxonomy, Ethernet, 802.11. Offered as CSDS 325 and CSDS 325N. Prereq: Minimum Junior Standing and (CSDS 233 or ECSE 233 or EECS 233) with a C or higher.

**CSDS 335D. Data Mining for Big Data. 3 Units.**

With the unprecedented rate at which data is being collected today in almost all fields of human endeavor, there is an emerging economic and scientific need to extract useful information from it. Data mining is the process of automatic discovery of patterns, changes, associations and anomalies in massive databases, and is a highly interdisciplinary field representing the confluence of several disciplines, including database systems, data warehousing, machine learning, statistics, algorithms, data visualization, and high-performance computing. This course is an introduction to the commonly used data mining techniques. In the first part of the course, students will develop a basic understanding of the basic concepts in data mining such as frequent pattern mining, association rule mining, basic techniques for data preprocessing such as normalization, regression, and classic matrix decomposition methods such as SVD, LU, and QR decompositions. In the second part of the course, students will develop a basic understanding of classification and clustering and be able to apply classic methods such as k-means, hierarchical clustering methods, nearest neighbor methods, association based classifiers. In the third part of the course, students will have a chance to study more advanced data mining applications such as feature selection in high-dimensional data, dimension reduction, and mining biological datasets. Prereq: CSDS 234 and CSDS 313.

**CSDS 337. Compiler Design. 4 Units.**

Design and implementation of compilers and other language processors. Scanners and lexical analysis; regular expressions and finite automata; scanner generators; parsers and syntax analysis; context free grammars; parser generators; semantic analysis; intermediate code generation; runtime environments; code generation; machine independent optimizations; data flow and dependence analysis. There will be a significant programming project involving the use of compiler tools and software development tools and techniques. Offered as CSDS 337 and ECSE 337. Prereq: (CSDS 233 or ECSE 233 or EECS 233) and (CSDS 281 or ECSE 281 or EECS 281).

**CSDS 338. Intro to Operating Systems and Concurrent Programming. 4 Units.**

Intro to OS: OS Structures, processes, threads, CPU scheduling, deadlocks, memory management, file system implementations, virtual machines, cloud computing. Concurrent programming: fork, join, concurrent statement, critical section problem, safety and liveness properties of concurrent programs, process synchronization algorithms, semaphores, monitors. UNIX systems programming: system calls, UNIX System V IPCs, threads, RPCs, shell programming. Offered as CSDS 338, ECSE, 338, CSDS 338N and ECSE 338N. Prereq: Computer Science Major or Minor and (CSDS 233 or ECSE 233 or EECS 233) with a C or higher.

**CSDS 338N. Intro to Operating Systems and Concurrent Programming. 4 Units.**

Intro to OS: OS Structures, processes, threads, CPU scheduling, deadlocks, memory management, file system implementations, virtual machines, cloud computing. Concurrent programming: fork, join, concurrent statement, critical section problem, safety and liveness properties of concurrent programs, process synchronization algorithms, semaphores, monitors. UNIX systems programming: system calls, UNIX System V IPCs, threads, RPCs, shell programming. Offered as CSDS 338, ECSE, 338, CSDS 338N and ECSE 338N. Prereq: (CSDS 233 or ECSE 233 or EECS 233) with a C or higher.

**CSDS 339. Web Data Mining. 3 Units.**

Web crawling technology, web search and information extraction, unsupervised and semi-supervised learning techniques and their application to web data extraction, social network analysis, various pagerank algorithms, link analysis, web resource discovery, web, resource description framework (RDF), XML, Web Ontology Language (OWL). Prereq: (CSDS 338 or ECSE 338) and CSDS 341 and (CSDS 302 or ECSE 302 or MATH 304).

**CSDS 340D. Machine Learning for Big Data. 3 Units.**

Machine learning is a sub-field of Artificial Intelligence that is concerned with the design and analysis of algorithms that "learn" and improve with experience. While the broad aim behind research in this area is to build systems that can simulate or even improve on certain aspects of human intelligence, algorithms developed in this area have become very useful in analyzing and predicting the behavior of complex systems. Machine learning algorithms have been used to guide diagnostic systems in medicine, recommend interesting products to customers in e-commerce, play games at human championship levels, and solve many other very complex problems. This course is an introduction to algorithms for machine learning and their implementation in the context of big data. We will study different learning settings, the different algorithms that have been developed for these settings, and learn about how to implement these algorithms and evaluate their behavior in practice. We will also discuss dealing with noise, missing values, scalability properties and talk about tools and libraries available for these methods. At the end of the course, you should be able to: –Understand when to use machine learning algorithms; –Understand, represent and formulate the learning problem; –Apply the appropriate algorithm(s) or tools, with an understanding of the tradeoffs involved including scalability and robustness; –Correctly evaluate the behavior of the algorithm when solving the problem. Prereq: CSDS 234 and CSDS 313.

**CSDS 341. Introduction to Database Systems. 3 Units.**

Relational model, ER model, relational algebra and calculus, SQL, OBE, security, views, files and physical database structures, query processing and query optimization, normalization theory, concurrency control, object relational systems, multimedia databases, Oracle SQL server, Microsoft SQL server. Offered as CSDS 341 and CSDS 341N. Prereq: Computer Science Major or Data Science and Analytics Major and (CSDS 302 or ECSE 302 or MATH 304) and (CSDS233 or ECSE 233) with a C or higher.

**CSDS 341D. Introduction to Databases: DS Major. 3 Units.**

Database management become a central component of a modern computing environment, and, as a result, knowledge about database systems has become an essential part of education in computer science and data science. This course is an introduction to the nature and purpose of database systems, fundamental concepts for designing, implementing and querying a database and database architectures. Weeks 1-6 provide an overview of basic database systems concepts including database design, database systems architecture, and database querying, using relational model and SQL as query language. Weeks 7-10 Objects, Semi structured data, XML and RDF basics. Weeks 11-14 provide an overview of more advanced topics including Database System Architectures (Parallel Databases and Distributed Databases), and Data Warehousing and Information Retrieval. Students cannot receive credit for both CSDS341 and CSDS341D. Prereq: CSDS 233 or ECSE 233 or CSDS 234.

**CSDS 341N. Introduction to Database Systems. 3 Units.**

Relational model, ER model, relational algebra and calculus, SQL, OBE, security, views, files and physical database structures, query processing and query optimization, normalization theory, concurrency control, object relational systems, multimedia databases, Oracle SQL server, Microsoft SQL server. Offered as CSDS 341 and CSDS 341N. Prereq: (CSDS 302 or ECSE 302 or MATH 304) with a grade of C or higher and (CSDS 233 or ECSE 233) with a C or higher.

**CSDS 343. Theoretical Computer Science. 3 Units.**

Introduction to different classes of automata and their correspondence to different classes of formal languages and grammars, computability, complexity and various proof techniques. Offered as CSDS 343 and MATH 343. Prereq: EECS 302 or MATH 304.

**CSDS 344. Computer Security. 3 Units.**

General types of security attacks; approaches to prevention; secret key and public key cryptography; message authentication and hash functions; digital signatures and authentication protocols; information gathering; password cracking; spoofing; session hijacking; denial of service attacks; buffer overruns; viruses, worms, etc., principles of secure software design, threat modeling; access control; least privilege; storing secrets; socket security; firewalls; intrusions; auditing; mobile security. Recommended preparation: EECS/CSDS/ECSE 132, EECS/CSDS 293, EECS/CSDS 325 and EECS/CSDS 338. Offered as CSDS 344 and CSDS 444.

**CSDS 345. Programming Language Concepts. 3 Units.**

This course examines the four main programming paradigms: imperative, object-oriented, functional, and logical. It is assumed that students will come to the course with significant exposure to object-oriented programming and some exposure to imperative programming. The course will teach the functional paradigm in depth, enhance the students' knowledge of the object-oriented and imperative paradigms, and introduce the logical paradigm. The course will explore language syntax, semantics, names/scopes, types, expressions, assignment, subprograms, abstraction and inheritance. This exploration will have several forms. Students will study the programming language concepts at a theoretical level, use the concepts in functional language programming, and implement the concepts by designing language interpreters. Offered as CSDS 345 and CSDS 345N. Prereq: Computer Science Major and (CSDS 302 or ECSE 302 or MATH 304) with a C or higher and (CSDS 233 or ECSE 233) with a C or higher.

**CSDS 345N. Programming Language Concepts. 3 Units.**

This course examines the four main programming paradigms: imperative, object-oriented, functional, and logical. It is assumed that students will come to the course with significant exposure to object-oriented programming and some exposure to imperative programming. The course will teach the functional paradigm in depth, enhance the students' knowledge of the object-oriented and imperative paradigms, and introduce the logical paradigm. The course will explore language syntax, semantics, names/scopes, types, expressions, assignment, subprograms, abstraction and inheritance. This exploration will have several forms. Students will study the programming language concepts at a theoretical level, use the concepts in functional language programming, and implement the concepts by designing language interpreters. Offered as CSDS 345 and CSDS 345N. Prereq: (CSDS 302 or ECSE 302 or MATH 304) with a C or higher and (CSDS 233 or ECSE 233) with a C or higher.

**CSDS 356. Data Privacy. 3 Units.**

Introduction to privacy, economics and incentives, crypto-based solution for privacy, hiding data from the database user, hiding access patterns from the database owner, anonymous routing and TOR, privacy in online social networks, privacy in cellular and Wi-Fi networks, location privacy, privacy in e-cash systems, privacy in e-voting, genomic privacy. Offered as CSDS 356 and CSDS 456. Prereq: (EECS 132 and EECS 233 and any STAT course) or (Graduate student in Computing & Info Science or Computer Engineering).

**CSDS 364. Computational Perception. 3 Units.**

An introduction to the information processing and computational algorithms that underlie perception. The course focuses on vision and audition but also covers other senses and various types of perceptual processing in biological systems. The motivating question is: What are the fundamental computational problems faced by perceptual systems in complex environments? The course draws from fields in engineering and the study of biological perception. Specific topics include signal detection and processing; sound localization; motion perception and estimation; sensory coding; perception of structure; active perception; perceptual invariance; attention; object, sound and speech recognition; visual and auditory scene analysis. Offered as CSDS 364 and CSDS 464. Prereq: CSDS 132 or ENGR 131.

**CSDS 366. Computer Graphics. 3 Units.**

Theory and practice of computer graphics: object and environment representation including coordinate transformations image extraction including perspective, hidden surface, and shading algorithms; and interaction. Covers a wide range of graphic display devices and systems with emphasis in interactive shaded graphics. Offered as CSDS 366, ECSE 366, CSDS 466 and ECSE 466. Prereq: EECS 233.



**CSDS 373. Modern Robot Programming. 3 Units.**

The goal of this course is to learn modern methods for building up robot capabilities using the Robot Operating System (ROS). Through a sequence of assignments, students learn how to write software to control both simulated and physical robots. Material includes: interfacing software to robot I/O; path and trajectory planning for robot arms; object identification and localization from 3-D sensing; manipulation planning; and development of graphical interfaces for supervisory robot control. Laboratory assignments are scheduled in small groups to explore implementations on specific robots. Graduate students will also perform an independent project. Offered as CSDS 373, ECSE 373, CSDS 473 and ECSE 473.

**CSDS 376. Mobile Robotics. 4 Units.**

Design of software systems for mobile robot control, including: motion control; sensory processing; localization and mapping; mobile-robot planning and navigation; and implementation of goal-directed behaviors. The course has a heavy lab component involving a sequence of design challenges and competitions performed in teams. Offered as CSDS 376 and ECSE 376. Prereq: CSDS 373 or CSDS 473.

**CSDS 377. Introduction to Connected Devices. 3 Units.**

Introduction to Connected Devices (e.g., Internet of Things). Undergraduates work in pairs to build a complete connected-device system, an embedded device with wireless networking, cloud and web, and mobile, and then develop hands-on experience with systems-level aspects of the connected-device system, including analytics, remote firmware update, load testing, and essential security. Students learn about current architectures, languages, and technologies, such as Pub/Sub (MQTT), Python, Objective-C, Python Django, JavaScript, HTML/CSS, and Bluetooth Low Energy. Offered as CSDS 377 and ECSE 377.

**CSDS 386. Quantum Computing, Information, and Devices. 3 Units.**

An introduction to the math, physics, engineering, and computer science underlying the rapidly emerging fields of quantum computing, quantum information, and quantum devices. The course is taught by a group of faculty from physics, engineering, computer science, and math, and is geared towards students with diverse backgrounds and interests in these fields. Students will select a concentration in one of these four areas, and the coursework, while still covering all topics, will be adjusted to focus on the selected area in the most detail. Note that the listed prerequisites depend on choice of concentration. Topics will include: 1. (Mathematics) Introduction to linear algebra, convex geometry, fundamental theory of quantum information. 2. (Physics) Introduction to the quantum mechanics of two-level systems (qubits). Survey of physics and materials for qubit technologies. 3. (Computer Science) Basic quantum gates and circuits, introduction to the theory of algorithms, survey of quantum algorithms. 4. (Engineering) Quantum architectures, mapping algorithms onto circuits. The course consists of lectures, homework, and group projects. Group projects will aim to synthesize the diverse backgrounds of the students and instructors to capture the interdisciplinary nature of the field. Students taking the course for graduate credit will complete an additional literature research project and presentation, in addition to enhanced problem sets. Offered as CSDS 386, CSDS 486, ECSE 386, ECSE 486, MATH 386, MATH 486, PHYS 386, and PHYS 486. Prereq: (CSDS 233 or ECSE 233) and (CSDS 281 or ECSE 281) and (MATH 201 or MATH 307) and (MATH 223 or MATH 227) and (PHYS 122 or PHYS 124).

**CSDS 390. Advanced Game Development Project. 3 Units.**

This game development project course will bring together an inter-professional group of students in the fields of engineering, computer science, and art to focus on the design and development of a complete, fully functioning computer game as an interdisciplinary team. The student teams are given complete liberty to design their own fully functional games from their original concept to a playable game published in an online marketplace. Student teams will experience the entire game development cycle as they execute their projects. Responsibilities include creating a game idea, writing a story, developing the artwork, designing characters, implementing music and sound effects, programming and testing the game, and publishing the final project. Students enrolled in 487 will develop a healthcare or education virtual environment or video game in collaboration with a mentor who has expertise in the chosen area. Offered as CSDS 390, ECSE 390, CSDS 487, and ECSE 487. Prereq: EECS 233 and EECS 290.

**CSDS 391. Introduction to Artificial Intelligence. 3 Units.**

This course is an introduction to artificial intelligence. We will study the concepts that underlie intelligent systems. Topics covered include problem solving with search, constraint satisfaction, adversarial games, knowledge representation and reasoning using propositional and first order logic, reasoning under uncertainty, introduction to machine learning, automated planning, reinforcement learning and natural language processing. Recommended: basic knowledge of probability and statistics. Prereq: ENGR 131 or CSDS 132 or ECSE 132.

**CSDS 392. App Development for iOS. 3 Units.**

This course is an introduction to app development for iPhone and iPad using Cocoa Touch Framework and Xcode development environment. Topics include Swift programming language and iOS SDK/foundations, object-oriented design and model-view-controller framework, user interface design using Xcode. Additional topics may include data management, map applications, animations and some recent developments in iOS. Recommended preparation: experiences in object-oriented programming and Mac OS; knowledge in software engineering and databases. Prereq: EECS 293 and Junior or Senior standing.

**CSDS 393. Software Engineering. 3 Units.**

Topics: Introduction to software engineering; software lifecycle models; development team organization and project management; requirements analysis and specification techniques; software design techniques; programming practices; software validation techniques; software maintenance practices; software engineering ethics. Undergraduates work in teams to complete a significant software development project. Graduate students are required to complete a research project. Offered as CSDS 393, CSDS 393N, and CSDS 493. Prereq: (Computer Science Major/Minor or Data Science Major) and (CSDS 233 or ECSE 233) with a C or higher.

**CSDS 393N. Software Engineering. 3 Units.**

Topics: Introduction to software engineering; software lifecycle models; development team organization and project management; requirements analysis and specification techniques; software design techniques; programming practices; software validation techniques; software maintenance practices; software engineering ethics. Undergraduates work in teams to complete a significant software development project. Graduate students are required to complete a research project. Offered as CSDS 393, CSDS 393N, and CSDS 493. Prereq: CSDS 233 or ECSE 233 with a C or higher.

**CSDS 394. Introduction to Information Theory. 3 Units.**

This course is intended as an introduction to information and coding theory with emphasis on the mathematical aspects. It is suitable for advanced undergraduate and graduate students in mathematics, applied mathematics, statistics, physics, computer science and electrical engineering. Course content: Information measures-entropy, relative entropy, mutual information, and their properties. Typical sets and sequences, asymptotic equipartition property, data compression. Channel coding and capacity: channel coding theorem. Differential entropy, Gaussian channel, Shannon-Nyquist theorem. Information theory inequalities (400 level). Additional topics, which may include compressed sensing and elements of quantum information theory. Recommended preparation: MATH 201 or MATH 307. Offered as MATH 394, CSDS 394, ECSE 394, MATH 494, CSDS 494 and ECSE 494. Prereq: MATH 223 and MATH 380 or requisites not met permission.

**CSDS 395. Senior Project in Computer Science. 4 Units.**

Capstone course for computer science seniors. Material from previous and concurrent courses used to solve computer programming problems and to develop software systems. Professional engineering topics such as project management, engineering design, communications, and professional ethics. Requirements include periodic reporting of progress, plus a final oral presentation and written report. Scheduled formal project presentations during last week of classes. Counts as SAGES Senior Capstone. Prereq: Senior standing.

**CSDS 396. Independent Projects. 1 - 6 Units.**

Independent projects in Computer and Data Science. Prereq: Limited to juniors and seniors.

**CSDS 397. Special Topics. 1 - 6 Units.**

Special topics in Computer and Data Science. Prereq: Limited to juniors and seniors.

**CSDS 400T. Graduate Teaching I. 0 Unit.**

This course will provide the Ph.D. candidate with experience in teaching undergraduate or graduate students. The experience is expected to involve direct student contact but will be based upon the specific departmental needs and teaching obligations. This teaching experience will be conducted under the supervision of the faculty member who is responsible for the course, but the academic advisor will assess the educational plan to ensure that it provides an educational experience for the student. Students in this course may be expected to perform one or more of the following teaching related activities: grading homeworks, quizzes, and exams, having office hours for students, tutoring students. Recommended preparation: Ph.D. student in CSDS department.

**CSDS 405. Data Structures and File Management. 3 Units.**

Fundamental concepts: sequential allocation, linked allocation, lists, trees, graphs, internal sorting, external sorting, sequential, binary, interpolation search, hashing file, indexed files, multiple level index structures, btrees, hashed files. Multiple attribute retrieval; inverted files, multi lists, multiple-key hashing, hd trees. Introduction to data bases. Data models. Recommended preparation: EECS/CSDS/ECSE 233 and MATH 304. Prereq: EECS 341 or Graduate standing.

**CSDS 410. Analysis of Algorithms. 3 Units.**

This course covers fundamental topics in algorithm design and analysis in depth. Amortized analysis, NP-completeness and reductions, dynamic programming, advanced graph algorithms, string algorithms, geometric algorithms, local search heuristics. Offered as CSDS 410 and OPRE 454. Prereq: EECS 340.

**CSDS 425. Computer Networks I. 3 Units.**

An introduction to computer networks and the Internet. Applications: http, ftp, e-mail, DNS, socket programming. Transport: UDP, TCP, reliable data transfer, and congestion control. Network layer: IP, routing and NAT. Link layer: taxonomy, Ethernet, 802.11. Recommended preparation: EECS/CSDS/ECSE 338.

**CSDS 427. Internet Security and Privacy. 3 Units.**

This course introduces students to research on Internet security and privacy. Covered topics include denial of service attacks, attacks enabled by man-in-the-middle surveillance, communication hijacking, botnet and fast-flux networks, email and Web spam, threats to privacy on the Internet, and Internet censorship. The course will be based on a collection of research papers. Students will be required to attend lectures, read the materials, prepare written summaries of discussed papers, present a paper in class, complete a course project and take the final exam (in the form of the course project presentation). Prereq: EECS 325 or EECS 425 or graduate standing in Computer Science or Computer Engineering.

**CSDS 428. Computer Communications Networks II. 3 Units.**

Introduction to topics and methodology in computer networks and middleware research. Traffic characterization, stochastic models, and self-similarity. Congestion control (Tahoe, Reno, Sack). Active Queue Management (RED, FQ) and explicit QoS. The Web: overview and components, HTTP, its interaction with TCP, caching. Overlay networks and CDN. Expected work includes a course-long project on network simulation, a final project, a paper presentation, midterm, and final test. Recommended preparation: EECS/CSDS 425.

**CSDS 433. Database Systems. 3 Units.**

Basic issues in file processing and database management systems. Physical data organization. Relational databases. Database design. Relational Query Languages, SQL. Query languages. Query optimization. Database integrity and security. Object-oriented databases. Object-oriented Query Languages, OQL. Recommended preparation: MATH 304. Prereq: EECS 341 or Graduate standing.

**CSDS 435. Data Mining. 3 Units.**

Data Mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories. Topics to be covered includes: Data Warehouse and OLAP technology for data mining, Data Preprocessing, Data Mining Primitives, Languages, and System Architectures, Mining Association Rules from Large Databases, Classification and Prediction, Cluster Analysis, Mining Complex Types of Data, and Applications and Trends in Data Mining. Prereq: EECS 341 or Graduate standing.

**CSDS 438. High Performance Computing. 3 Units.**

High performance computing (HPC) leverages parallel processing in order to maximize speed and throughput. This hands-on course will cover theoretical and practical aspects of HPC. Theoretical concepts covered include computer architecture, parallel programming, and performance optimization. Practical applications will be discussed from various information and scientific fields. Practical considerations will include HPC job management and Unix scripting. Weekly assessments and a course project will be required. Offered as CSDS 438 and ECSE 438. Prereq: EECS 233 or graduate standing.

**CSDS 439. Web Data Mining. 3 Units.**

Web crawling technology, web search and information extraction, unsupervised and semi-supervised learning techniques and their application to web data extraction, social network analysis, various pagerank algorithms, link analysis, web resource discovery, web, resource description framework (RDF), XML, Web Ontology Language (OWL). Recommended preparation: EECS/CSDS/ECSE 338, EECS/CSDS 341.

**CSDS 440. Machine Learning. 3 Units.**

Machine learning is a subfield of Artificial Intelligence that is concerned with the design and analysis of algorithms that "learn" and improve with experience. While the broad aim behind research in this area is to build systems that can simulate or even improve on certain aspects of human intelligence, algorithms developed in this area have become very useful in analyzing and predicting the behavior of complex systems. Machine learning algorithms have been used to guide diagnostic systems in medicine, recommend interesting products to customers in e-commerce, play games at human championship levels, and solve many other very complex problems. This course is focused on algorithms for machine learning: their design, analysis and implementation. We will study different learning settings, including supervised, semi-supervised and unsupervised learning. We will study different ways of representing the learning problem, using propositional, multiple-instance and relational representations. We will study the different algorithms that have been developed for these settings, such as decision trees, neural networks, support vector machines, k-means, harmonic functions and Bayesian methods. We will learn about the theoretical tradeoffs in the design of these algorithms, and how to evaluate their behavior in practice. At the end of the course, you should be able to: –Recognize situations where machine learning algorithms are applicable; –Understand, represent and formulate the learning problem; –Apply the appropriate algorithm(s), or if necessary, design your own, with an understanding of the tradeoffs involved; –Correctly evaluate the behavior of the algorithm when solving the problem. Prereq: CSDS 391 or CSDS 491.

**CSDS 442. Causal Learning from Data. 3 Units.**

This course introduces key concepts and techniques for characterizing, from observational or experimental study data and from background information, the causal effect of a specific treatment, exposure, or intervention (e.g., a medical treatment) upon an outcome of interest (e.g., disease status). The fundamental problem of causal inference is the impossibility of observing the effects of different and incompatible treatments on the same individual or unit. This problem is overcome by estimating an average causal effect over a study population. Making valid causal inferences with observational data is especially challenging, because of the greater potential for biases (confounding bias, selection bias, and measurement bias) that can badly distort causal effect estimates. Consequently, this topic has been the focus of intense cross-disciplinary research in recent years. Causal inference techniques will be illustrated by applications in several fields such as computer science, engineering, medicine, public health, biology, genomics, neuroscience, economics, and social science. Course grading will be based on quizzes, homeworks, a class presentation, and a causal data analysis project. Specific topics: treatments, exposures, and interventions; causal effects and causal effect measures; confounding bias; potential outcomes and counterfactuals; randomized experiments; observational studies; causal directed acyclic graphs (DAGs); exchangeability and conditional exchangeability; effect modification; causal interactions; nonparametric structural equations; Pearl's Back-Door Criterion, Front-Door Criterion, and related results; covariate adjustment; matching on covariates; selection bias; measurement bias; instrumental variables; causal modeling; inverse probability weighting; marginal structural models; standardization; structural nested models; outcome regression; propensity scores; sensitivity analysis. Prereq: EECS 440 or MATH 380 or STAT 312 or STAT 313 or STAT 332 or STAT 333 or Requisites Not Met permission.

**CSDS 444. Computer Security. 3 Units.**

General types of security attacks; approaches to prevention; secret key and public key cryptography; message authentication and hash functions; digital signatures and authentication protocols; information gathering; password cracking; spoofing; session hijacking; denial of service attacks; buffer overruns; viruses, worms, etc., principles of secure software design, threat modeling; access control; least privilege; storing secrets; socket security; firewalls; intrusions; auditing; mobile security. Recommended preparation: EECS/CSDS/ECSE 132, EECS/CSDS 293, EECS/CSDS 325 and EECS/CSDS 338. Offered as CSDS 344 and CSDS 444.

**CSDS 448. Smartphone Security. 3 Units.**

This course is designed to better prepare undergraduate and graduate students for researching and developing in the neighborhood of mobile and software security. Lectures, paper readings and presentations, in-class discussions, and projects are the main components. The course covers the basics of Android programming and a wide range of security issues and solutions concerning mobile platforms, including permission analysis, textual artifacts analysis, malware analysis, program analysis, and UI analysis. Students should expect one literature survey paper and one system-building or empirical study project on one selected security solution in mobile app security. Prereq: EECS 132 and a Graduate or Undergraduate Computer Science major.

**CSDS 455. Applied Graph Theory. 3 Units.**

This course serves as an introduction to many of the important aspects of graph theory. Topics include connectivity, flows, matchings, planar graphs, and graph coloring with additional topics selected from extremal graphs, random graphs, bounded treewidth graphs, social networks and small world graphs. The class will explore the underlying mathematical theory with a specific focus on the development and analysis of graph algorithms. Prereq: Graduate Student or (EECS 302 or MATH 304 or MATH 305 or MATH 307).

**CSDS 456. Data Privacy. 3 Units.**

Introduction to privacy, economics and incentives, crypto-based solution for privacy, hiding data from the database user, hiding access patterns from the database owner, anonymous routing and TOR, privacy in online social networks, privacy in cellular and Wi-Fi networks, location privacy, privacy in e-cash systems, privacy in e-voting, genomic privacy. Offered as CSDS 356 and CSDS 456. Prereq: (EECS 132 and EECS 233 and any STAT course) or (Graduate student in Computing & Info Science or Computer Engineering).

**CSDS 458. Introduction to Bioinformatics. 3 Units.**

Fundamental algorithmic and statistical methods in computational molecular biology and bioinformatics will be discussed. Topics include introduction to molecular biology and genetics, DNA sequence analysis, polymorphisms and personal genomics, structural variation analysis, gene mapping and haplotyping algorithms, phylogenetic analysis, biological network analysis, and computational drug discovery. Much of the course will focus on the algorithmic techniques, including but not limited to, dynamic programming, hidden Markov models, string algorithms, graph theories and algorithms, and some representative data mining algorithms. Paper presentations and course projects are also required. Prereq: EECS 340 or Graduate standing.

**CSDS 459. Bioinformatics for Systems Biology. 3 Units.**

Description of omic data (biological sequences, gene expression, protein-protein interactions, protein-DNA interactions, protein expression, metabolomics, biological ontologies), regulatory network inference, topology of regulatory networks, computational inference of protein-protein interactions, protein interaction databases, topology of protein interaction networks, module and protein complex discovery, network alignment and mining, computational models for network evolution, network-based functional inference, metabolic pathway databases, topology of metabolic pathways, flux models for analysis of metabolic networks, network integration, inference of domain-domain interactions, signaling pathway inference from protein interaction networks, network models and algorithms for disease gene identification, identification of dysregulated subnetworks network-based disease classification. Offered as CSDS 459 and SYBB 459. Prereq: CSDS 458 or BIOL 250.

**CSDS 464. Computational Perception. 3 Units.**

An introduction to the information processing and computational algorithms that underlie perception. The course focuses on vision and audition but also covers other senses and various types of perceptual processing in biological systems. The motivating question is: What are the fundamental computational problems faced by perceptual systems in complex environments? The course draws from fields in engineering and the study of biological perception. Specific topics include signal detection and processing; sound localization; motion perception and estimation; sensory coding; perception of structure; active perception; perceptual invariance; attention; object, sound and speech recognition; visual and auditory scene analysis. Offered as CSDS 364 and CSDS 464. Prereq: CSDS 132 or ENGR 131.

**CSDS 465. Computer Vision. 3 Units.**

The goal of computer vision is to create visual systems that recognize objects and recover structures in complex 3D scenes. This course emphasizes both the science behind our understanding of the fundamental problems in vision and the engineering that develops mathematical models and inference algorithms to solve these problems. Specific topics include feature detection, matching, and classification; visual representations and dimensionality reduction; motion detection and optical flow; image segmentation; depth perception, multi-view geometry, and 3D reconstruction; shape and surface perception; visual scene analysis and object recognition. Offered as CSDS 465 and ECSE 465.

**CSDS 466. Computer Graphics. 3 Units.**

Theory and practice of computer graphics: object and environment representation including coordinate transformations image extraction including perspective, hidden surface, and shading algorithms; and interaction. Covers a wide range of graphic display devices and systems with emphasis in interactive shaded graphics. Offered as CSDS 366, ECSE 366, CSDS 466 and ECSE 466. Prereq: Graduate standing or Requisites Not Met permission.

**CSDS 473. Modern Robot Programming. 3 Units.**

The goal of this course is to learn modern methods for building up robot capabilities using the Robot Operating System (ROS). Through a sequence of assignments, students learn how to write software to control both simulated and physical robots. Material includes: interfacing software to robot I/O; path and trajectory planning for robot arms; object identification and localization from 3-D sensing; manipulation planning; and development of graphical interfaces for supervisory robot control. Laboratory assignments are scheduled in small groups to explore implementations on specific robots. Graduate students will also perform an independent project. Offered as CSDS 373, ECSE 373, CSDS 473 and ECSE 473.

**CSDS 476. Mobile Robotics. 3 Units.**

Design of software systems for mobile robot control, including: motion control; sensory processing; localization and mapping; mobile-robot planning and navigation; and implementation of goal-directed behaviors. The course has a heavy lab component involving a sequence of design challenges and competitions performed in teams. Offered as CSDS 476 and ECSE 476. Prereq: EECS 373 or EECS 473.

**CSDS 477. Advanced Algorithms. 3 Units.**

Design and analysis of efficient algorithms, with emphasis on network flow, combinatorial optimization, and randomized algorithms. Linear programming: duality, complementary slackness, total unimodularity. Minimum cost flow: optimality conditions, algorithms, applications. Game theory: two-person zero-sum games, minimax theorems. Probabilistic analysis and randomized algorithms: examples and lower bounds. Approximation algorithms for NP-hard problems: examples, randomized rounding of linear programs. Prereq: EECS 302 and EECS 340 and MATH 380 and (MATH 201 or MATH 307).

**CSDS 478. Computational Neuroscience. 3 Units.**

Computer simulations and mathematical analysis of neurons and neural circuits, and the computational properties of nervous systems. Students are taught a range of models for neurons and neural circuits, and are asked to implement and explore the computational and dynamic properties of these models. The course introduces students to dynamical systems theory for the analysis of neurons and neural learning, models of brain systems, and their relationship to artificial and neural networks. Term project required. Students enrolled in MATH 478 will make arrangements with the instructor to attend additional lectures and complete additional assignments addressing mathematical topics related to the course. Recommended preparation: MATH 223 and MATH 224 or BIOL 300 and BIOL 306. Offered as BIOL 378, COGS 378, MATH 378, BIOL 478, CSDS 478, EBME 478, ECSE 478, MATH 478 and NEUR 478.

**CSDS 486. Quantum Computing, Information, and Devices. 3 Units.**

An introduction to the math, physics, engineering, and computer science underlying the rapidly emerging fields of quantum computing, quantum information, and quantum devices. The course is taught by a group of faculty from physics, engineering, computer science, and math, and is geared towards students with diverse backgrounds and interests in these fields. Students will select a concentration in one of these four areas, and the coursework, while still covering all topics, will be adjusted to focus on the selected area in the most detail. Note that the listed prerequisites depend on choice of concentration. Topics will include: 1. (Mathematics) Introduction to linear algebra, convex geometry, fundamental theory of quantum information. 2. (Physics) Introduction to the quantum mechanics of two-level systems (qubits). Survey of physics and materials for qubit technologies. 3. (Computer Science) Basic quantum gates and circuits, introduction to the theory of algorithms, survey of quantum algorithms. 4. (Engineering) Quantum architectures, mapping algorithms onto circuits. The course consists of lectures, homework, and group projects. Group projects will aim to synthesize the diverse backgrounds of the students and instructors to capture the interdisciplinary nature of the field. Students taking the course for graduate credit will complete an additional literature research project and presentation, in addition to enhanced problem sets. Offered as CSDS 386, CSDS 486, ECSE 386, ECSE 486, MATH 386, MATH 486, PHYS 386, and PHYS 486. Prereq: (CSDS 233 or ECSE 233) and (CSDS 281 or ECSE 281) and (MATH 201 or MATH 307) and (MATH 223 or MATH 227) and (PHYS 122 or PHYS 124).



**CSDS 487. Advanced Game Development Project. 3 Units.**

This game development project course will bring together an inter-professional group of students in the fields of engineering, computer science, and art to focus on the design and development of a complete, fully functioning computer game as an interdisciplinary team. The student teams are given complete liberty to design their own fully functional games from their original concept to a playable game published in an online marketplace. Student teams will experience the entire game development cycle as they execute their projects. Responsibilities include creating a game idea, writing a story, developing the artwork, designing characters, implementing music and sound effects, programming and testing the game, and publishing the final project. Students enrolled in 487 will develop a healthcare or education virtual environment or video game in collaboration with a mentor who has expertise in the chosen area. Offered as CSDS 390, ECSE 390, CSDS 487, and ECSE 487. Prereq: Graduate student standing.

**CSDS 489. Robotics I. 3 Units.**

Orientation and configuration coordinate transformations, forward and inverse kinematics and Newton-Euler and Lagrange-Euler dynamic analysis. Planning of manipulator trajectories. Force, position, and hybrid control of robot manipulators. Analytical techniques applied to select industrial robots. Recommended preparation: EMAE 181. Offered as CSDS 489, ECSE 489 and EMAE 489.

**CSDS 490. Digital Image Processing. 3 Units.**

Digital images are introduced as two-dimensional sampled arrays of data. The course begins with one-to-one operations such as image addition and subtraction and image descriptors such as the histogram. Basic filters such as the gradient and Laplacian in the spatial domain are used to enhance images. The 2-D Fourier transform is introduced and frequency domain operations such as high and low-pass filtering are developed. It is shown how filtering techniques can be used to remove noise and other image degradation. The different methods of representing color images are described and fundamental concepts of color image transformations and color image processing are developed. One or more advanced topics such as wavelets, image compression, and pattern recognition will be covered as time permits. Programming assignments using software such as MATLAB will illustrate the application and implementation of digital image processing. Offered as CSDS 490 and ECSE 490.

**CSDS 491. Artificial Intelligence: Probabilistic Graphical Models. 3 Units.**

This course is a graduate-level introduction to Artificial Intelligence (AI), the discipline of designing intelligent systems, and focuses on probabilistic graphical models. These models can be applied to a wide variety of settings from data analysis to machine learning to robotics. The models allow intelligent systems to represent uncertainties in an environment or problem space in a compact way and reason intelligently in a way that makes optimal use of available information and time. The course covers directed and undirected probabilistic graphical models, latent variable models, associated exact and approximate inference algorithms, and learning in both discrete and continuous problem spaces. Practical applications are covered throughout the course. Prereq: CSDS 391 or Requisites Not Met permission.

**CSDS 493. Software Engineering. 3 Units.**

Topics: Introduction to software engineering; software lifecycle models; development team organization and project management; requirements analysis and specification techniques; software design techniques; programming practices; software validation techniques; software maintenance practices; software engineering ethics. Undergraduates work in teams to complete a significant software development project. Graduate students are required to complete a research project. Offered as CSDS 393, CSDS 393N, and CSDS 493. Prereq: (Computer Science Major/ Minor or Data Science Major) and (CSDS 233 or ECSE 233) with a C or higher.

**CSDS 494. Introduction to Information Theory. 3 Units.**

This course is intended as an introduction to information and coding theory with emphasis on the mathematical aspects. It is suitable for advanced undergraduate and graduate students in mathematics, applied mathematics, statistics, physics, computer science and electrical engineering. Course content: Information measures-entropy, relative entropy, mutual information, and their properties. Typical sets and sequences, asymptotic equipartition property, data compression. Channel coding and capacity: channel coding theorem. Differential entropy, Gaussian channel, Shannon-Nyquist theorem. Information theory inequalities (400 level). Additional topics, which may include compressed sensing and elements of quantum information theory. Recommended preparation: MATH 201 or MATH 307. Offered as MATH 394, CSDS 394, ECSE 394, MATH 494, CSDS 494 and ECSE 494.

**CSDS 496. Artificial Intelligence: Sequential Decision Making. 3 Units.**

This course will study the formulation and solution of decision making problems by automated agents. Topics covered include one-shot decision making (decision trees and influence diagrams), Markov decision processes (MDPs), automated classical and probabilistic planning, reinforcement learning (RL), hierarchical planning and RL, partially observable MDPs, Bayesian RL, collaborative multi-agent systems. Recommended preparation: EECS/CSDS 491 (Probabilistic Graphical Models). Prereq: EECS 391.

**CSDS 497. Artificial Intelligence: Statistical Natural Language Processing. 3 Units.**

This course gives students an overview of the state of the art in natural language processing. We will discuss computational aspects of language modeling through probabilistic models, computational approaches to syntax (parsing) and semantic representations, discourse and dialog. We will study the applications of these techniques to a variety of problems including information extraction, translation and summarization. At the end of the course a student should be able to (i) understand the various statistical models and algorithms for NLP (ii) modify them as needed or design novel approaches for specific NLP tasks and (iii) understand how to evaluate the performance of these models and compare them to alternatives. Prereq: EECS 440.

**CSDS 499. Algorithmic Robotics. 3 Units.**

This course introduces basic algorithmic techniques in robotic perception and planning. Course is divided into two parts. The first part introduces probabilistic modeling of robotic motion and sensing, Gaussian and nonparametric filters, and algorithms for mobile robot localization. The second part introduces fundamental deterministic and randomized algorithms for motion planning. Offered as CSDS 499 and ECSE 499. Prereq: Graduate Standing or Requisites Not Met permission.

**CSDS 500. CSDS Colloquium. 0 Unit.**

Seminars on current topics in Computer and Data Science.

**CSDS 500T. Graduate Teaching II. 0 Unit.**

This course will provide the Ph.D. candidate with experience in teaching undergraduate or graduate students. The experience is expected to involve direct student contact but will be based upon the specific departmental needs and teaching obligations. This teaching experience will be conducted under the supervision of the faculty member who is responsible for the course, but the academic advisor will assess the educational plan to ensure that it provides an educational experience for the student. Students in this course may be expected to perform one or more of the following teaching related activities: grading homeworks, quizzes, and exams, having office hours for students, running recitation sessions, providing laboratory assistance. Recommended preparation: Ph.D. student in CSDS department.

**CSDS 589. Robotics II. 3 Units.**

Survey of research issues in robotics. Force control, visual servoing, robot autonomy, on-line planning, high-speed control, man/machine interfaces, robot learning, sensory processing for real-time control. Primarily a project-based lab course in which students design real-time software executing on multi-processors to control an industrial robot. Recommended preparation: EECS/CSDS/ECSE 489. Offered as CSDS 589 and ECSE 589.

**CSDS 600. Special Topics. 1 - 18 Units.****CSDS 600T. Graduate Teaching III. 0 Unit.**

This course will provide Ph.D. candidate with experience in teaching undergraduate or graduate students. The experience is expected to involve direct student contact but will be based upon the specific departmental needs and teaching obligations. This teaching experience will be conducted under the supervision of the faculty member who is responsible for the course, but the academic advisor will assess the educational plan to ensure that it provides an educational experience for the student. Students in this course may be expected to perform one or more of the following teaching related activities running recitation sessions, providing laboratory assistance, developing teaching or lecture materials presenting lectures. Recommended preparation: Ph.D. student in CSES department.

**CSDS 601. Independent Study. 1 - 18 Units.****CSDS 620. Special Topics. 1 - 18 Units.****CSDS 621. Special Projects. 1 - 18 Units.****CSDS 651. Thesis M.S.. 1 - 18 Units.**

Credit as arranged.

**CSDS 695. Project M.S.. 1 - 9 Units.**

Research course taken by Plan B M.S. students. Prereq: Enrolled in the EECS Plan B MS Program.

**CSDS 701. Dissertation Ph.D.. 1 - 9 Units.**

Credit as arranged.