COMPUTER SCIENCE

http://cs.jhu.edu/

Computing has grown to be pervasive throughout engineering, science, business, society, and entertainment. Computer Science at Johns Hopkins University (CS@JHU) is a diverse, collaborative, and intensely research-focused (https://www.cs.jhu.edu/research/) department. Our mission in the university is to enhance discovery and innovation in science, engineering and society through computing research and education:

- To advance disciplinary science and engineering in core and applied computing with a applications focus on data intensive science and engineering, medicine, information security, language and robotics
- To enhance computing science research and education broadly in the school and the university through unique course offerings customized to students of varied backgrounds
- To identify and lead new computing-intensive initiatives for the school and the university

There are two dimensions to the field of computer science that establish it as a unique area. CS can be viewed as a stand-alone discipline worthy of study unto itself, and/or as an empowering discipline to be studied in conjunction with other areas. Core CS careers include (but are not limited to) software design and development, computer systems engineering or administration, and information security. Application areas span a wide range of fields and disciplines such as robotics, medical and health informatics, scientific research, data analytics, entrepreneurship, and human/computer interaction to name a few.

Because computer science is a highly diverse and broadly applied field, studies can proceed in many different directions. Accordingly, the undergraduate and graduate programs in the Department of Computer Science at Johns Hopkins are flexible curricula designed to accommodate a wide range of goals. A student at Johns Hopkins can pursue appropriately customized versions of the following computer science programs: minor, bachelor of science, bachelor of arts, masters of science in engineering, and doctor of philosophy. Most of this catalogue section is devoted to details regarding these programs.

Computer science research laboratories are currently active in the following areas at Hopkins: algorithm design and analysis, human-computer interaction, machine learning, data intensive computing, health informatics, computational medicine, computer vision and image processing, computer graphics, geometric modeling, programming languages, artificial intelligence, natural language and speech processing, information retrieval, cryptography and information security, secure and robust systems, storage systems, high-performance and scientific computing, computational genomics, networks and distributed systems, stream processing, robotics, and computer-integrated surgical systems.

Additionally, interdisciplinary research centers in the university have heavy involvement by Computer Science faculty: the Information Security Institute (ISI) (https://isi.jhu.edu/), the Laboratory for Computational Sensing and Robotics (LCSR) (https://lcsr.jhu.edu/), the Center for Language and Speech Processing (CLSP) (https://www.clsp.jhu.edu/), the Institute for Data Intensive Engineering and Science (IDIES) (http://idies.jhu.edu/), the Mathematical Institute for Data Science (MINDS) (https://www.minds.jhu.edu/), the Institute for Computational Medicine (ICM) (https://icm.jhu.edu/), the Center for Computational Biology (CCB) (http://ccb.jhu.edu/), the Institute for Assured Autonomy (IAA) (https://iaa.jhu.edu), and the Malone Center for Engineering in Healthcare (MCEH) (https://malonecenter.jhu.edu/). An important component of the educational process in the department is the opportunity for undergraduate and graduate student participation in the research programs of the faculty. In particular, original research in close association with individual faculty members is emphasized at the graduate level.

There are several closely related programs at the undergraduate and graduate levels which involve significant coursework and faculty involvement from the Department of Computer Science. The Laboratory for Computational Sensing and Robotics (LCSR) offers a minor in robotics and also a minor in computer integrated surgery through the Engineering Research Center for Computer Integrated Surgical Systems and Technology. Details of these programs may be found elsewhere in this catalogue in the section pertaining to the Laboratory for Computational Sensing and Robotics (https://e-catalogue.jhu.edu/engineering/robotics-computational-sensing/). Undergraduates with a strong interest in system design and performance may elect to pursue a bachelor degree in computer engineering (https://e-catalogue.jhu.edu/engineering/electrical-computer-engineering/). This field of study includes coursework in computer science, as well as electrical and computer engineering. Although programmatically shared by both departments, specific goals and requirements of the computer engineering degree may be found in the catalogue section pertaining to the Department of Electrical and Computer Engineering only.

At the graduate level, the LCSR (https://e-catalogue.jhu.edu/engineering/robotics-computational-sensing/) offers a Master of Science in Engineering (M.S.E.) in Robotics, designed for students from a wide variety of engineering, scientific, and mathematical backgrounds to advance their interdisciplinary knowledge in robotics. Details of this program may be found in the LCSR section of this catalogue, or on the web at www.lcsr.jhu.edu/MSE (http://www.lcsr.jhu.edu/MSE/). The M.S.E. in Computer Science offers an official concentration in Human Language and Technology (https://www.clsp.jhu.edu/human-language-technology-masters/) jointly administered through CLSP (https://www.clsp.jhu.edu/).

The CS department also has shared graduate programs with the Johns Hopkins University Information Security Institute (https://isi.jhu.edu/) (ISI) and the Department of Applied Mathematics (https://engineering.jhu.edu/ams/) and Statistics (https://engineering.jhu.edu/ams/):

- ISI and CS have a Dual Masters Program (DMP) combining the Master of Science in Security Informatics (MSSI) and the (MSE CS). The Master of Science in Security Informatics (MSSI) is a specialized graduate program offered through ISI. The field of security informatics is fundamentally based on information security and assurance technologies (hardware, software, and networks) as related to issues such as policy, management, privacy/trust, health care, and law, from both national and international perspectives. Interested students can obtain detailed information regarding the M.S.S.I. online at https://isi.jhu.edu or in the ISI section of this catalogue.
- The Data Science (https://engineering.jhu.edu/ams/data-science-masters-program/) program is a joint program between computer science and applied mathematics that aims to produce the next generation of leaders in data science by emphasizing mastery of the skills needed to translate real-world data-driven problems in mathematical ones, and then solving these problems by using a diverse collection of scientific tools.

For additional information regarding the academic programs available in Computer Science, and the facilities provided, please consult the sections which follow, or the departmental website cs.jhu.edu (https://cs.jhu.edu).

Combined Undergraduate/Graduate Program

As early as the beginning of their junior year, qualified students may apply for admission to a combined bachelor's/master's program which combines a B.S. or B.A. degree (in any department) with a master of science in engineering degree in Computer Science. This program allows students to simultaneously pursue both an undergraduate and a graduate degree program of study. Generally, the combined B.S./M.S.E. or B.A./M.S.E. program is accomplished in five years, although some students take more or less time. Applicants are judged on the basis of their performance in courses and their letters of recommendation. Double counting of at most two courses is subject to current WSE and departmental policies. Students may not take a 601.3xx or 601.4xx course as an undergraduate and the corresponding 601.6xx course for the M.S.E. degree. Combined students will have a graduate faculty advisor in the Computer Science Department who must approve the courses to be applied toward the master's degree. For information on the requirements of the M.S.E. degree, see the Graduate Programs tab on this page, or ask in the departmental office for the document that lists those requirements.

Facilities

The CS department is primarily housed in Malone Hall, a state-of-the-art, open-concept research facility. Additional department research space is located in the adjacent Hackerman Hall.

The general department computing facilities include numerous workstations and servers. Two undergraduate laboratories combine to provide approximately 24 Linux workstations. One of these is a collaboration room allowing students to work in a team-based environment, with several private breakout rooms as well. Both labs include a networked printer. At the graduate level, there is a Master's Lab consisting of a collaboration area and workstation area, both consisting of several Linux workstations and a networked printer. All Ph.D. students are assigned dedicated desks in their research labs.

Focused research laboratories have significant resources that provide greater specialization, including isolated networks of PCs for security studies, high-performance computing clusters, robots and computer vision systems, a mock operating room equipped with medical robots and imaging equipment, and more.

The general department computing facilities are tied together by our own LAN, and access to specialized hardware in other departments, labs, and institutions is available via the university intranet and the Internet. In addition, the university provides wireless access to the JHU intranet and the Internet, as well as server systems that provide email accounts for all students.

Programs

- Computer Science, Bachelor of Arts (https://e-catalogue.jhu.edu/engineering/full-time-residential-programs/degree-programs/computer-science/computer-science-ba/)
- Computer Science, Bachelor of Science (https://e-catalogue.jhu.edu/engineering/full-time-residential-programs/degree-programs/computer-science/computer-science-bs/)
- Computer Science, Master of Science in Engineering (https://e-catalogue.jhu.edu/engineering/full-time-residential-programs/degree-programs/computer-science/computer-science-master-science-engineering/)
- Computer Science, Minor (https://e-catalogue.jhu.edu/engineering/full-time-residential-programs/degree-programs/computer-science/computer-science-minor/)
- Computer Science, PhD (https://e-catalogue.jhu.edu/engineering/full-time-residential-programs/degree-programs/computer-science/computer-science-phd/)
- Security Informatics, Master of Science (https://e-catalogue.jhu.edu/engineering/full-time-residential-programs/degree-programs/information-security-institute/security-informatics-master-science/)

For current course information and registration go to https://sis.jhu.edu/classes/

Courses

EN.601.104. Computer Ethics. 1 Credit.
Students will examine a variety of topics regarding policy, legal, and moral issues related to the computer science profession itself and to the proliferation of computers in all aspects of society, especially in the era of the Internet. The course will cover various general issues related to ethical frameworks and apply those frameworks more specifically to the use of computers and the Internet. The topics will include privacy issues, computer crime, intellectual property law -- specifically copyright and patent issues, globalization, and ethical responsibilities for computer science professionals. Work in the course will consist of weekly assignments on one or more of the readings and a final paper on a topic chosen by the student and approved by the instructor.
Area: Humanities

EN.601.105. CS First-year Experience. 1 Credit.
This course provides first-year computer science majors with an introduction to the field and department. A variety of faculty members will lead weekly small group discussion sections on topics of interest related to the discipline. Upper-year majors will serve as peer mentors for each group (enrollment by permission only). Satisfactory/Unsatisfactory only; counts as elective credits only, not towards CS course credit requirement.

EN.601.124. The Ethics of Artificial Intelligence and Automation. 3 Credits.
The expansion of artificial intelligence (AI)-enabled use cases across a broad spectrum of domains has underscored the benefits and risks of AI. This course will address the various ethical considerations engineers need to engage with to build responsible and trustworthy AI-enabled autonomous systems. Topics to be covered include: values-based decision making, ethically aligned design, cultural diversity, safety, bias, AI explainability, privacy, AI regulation, the ethics of synthetic life, and the future of work. Case studies will be utilized to illustrate real-world applications. Students will apply learned material to a group research project on a topic of their choice.
Area: Humanities, Engineering

EN.601.220. Intermediate Programming. 4 Credits.
This course teaches intermediate to advanced programming, using C and C++. (Prior knowledge of these languages is not expected.) We will cover low-level programming techniques, as well as object-oriented class design, and the use of class libraries. Specific topics include pointers, dynamic memory allocation, polymorphism, overloading, inheritance, templates, collections, exceptions, and others as time permits. Students are expected to learn syntax and some language specific features independently. Course work involves significant programming projects in both languages.
Prerequisite(s): (EN.500.132 OR EN.500.133 OR EN.500.134) OR (EN.500.112 OR EN.500.113 OR EN.500.114)
Area: Engineering

EN.601.226. Data Structures. 4 Credits.
This course covers the design and implementation of data structures including arrays, stacks, queues, linked lists, binary trees, heaps, balanced trees (e.g. 2-3 trees, AVL-trees) and graphs. Other topics include sorting, hashing, memory allocation, and garbage collection. Course work involves both written homework and Java programming assignments.
Prerequisite(s): EN.500.132 OR (EN.500.112 or EN.601.220) or AP Computer Science or equivalent.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.229. Computer System Fundamentals. 3 Credits.
We study the design and performance of a variety of computer systems from simple 8-bit micro-controllers through 32/64-bit RISC architectures all the way to ubiquitous x86 CISC architecture. We'll start from logic gates and digital circuits before delving into arithmetic and logic units, registers, caches, memory, stacks and procedure calls, pipelined execution, super-scalar architectures, memory management units, etc. Along the way we'll study several typical instruction set architectures and review concepts such as interrupts, hardware and software exceptions, serial and other peripheral communications protocols, etc. A number of programming projects, frequently done in assembly language and using various processor simulators, round out the course. [Systems].
Prerequisite(s): EN.600.120/EN.601.220
Area: Engineering

EN.601.230. Mathematical Foundations for Computer Science. 4 Credits.
This course provides an introduction to mathematical reasoning and discrete structures relevant to computer science. Topics include propositional and predicate logic, proof techniques including mathematical induction, sets, relations, functions, recurrences, counting techniques, simple computational models, asymptotic analysis, discrete probability, graphs, trees, and number theory.
Prerequisite(s): EN.500.112 OR EN.500.113 OR EN.500.114 OR EN.500.132 OR EN.500.133 OR EN.500.134 OR EN.601.220;Student may not enroll if taken EN.601.231.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.231. Automata & Computation Theory. 3 Credits.
This course is an introduction to the theory of computing. topics include design of finite state automata, pushdown automata, linear bounded automata, Turing machines and phrase structure grammars; correspondence between automata and grammars; computable functions, decidable and undecidable problems, P and NP problems, NP-completeness, and randomization.
Prerequisite(s): EN.550.171/EN.553.171 OR EN.553.172;Students may not enroll if taken EN.601.230.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.270. Open Source Software Engineering (Semesters of Code I). 3 Credits.
The course will provide students a development experience focused on learning software engineering skills to deliver software at scale to a broad community of users associated with open source licensed projects. The class work will introduce students to ideas behind open source software with structured modules on recognizing and building healthy project structure, intellectual property basics, community & project governance, social and ethical concerns, and software economics.
Prerequisite(s): EN.601.220 AND EN.601.226
Area: Engineering

EN.601.277. Disinformation Self-Defense. 3 Credits.
Scientific, statistical and logical literacy is a necessary skill for evaluating policy proposals, reading news articles with an appropriately critical eye, and making informed choices as consumers and voters. Misunderstanding of claims made in scientific publications, online publishing platforms, and mass media drives, in part, the spread of malicious misinformation and propaganda online. Further, many actors have the means, the motive and the opportunity to mislead the public in a variety of subtle and not so subtle ways. This class will give you tools to discern valid and invalid forms of inference and discourse, and give you tools to communicate precisely, argue appropriately, and stay on top of research and news with an appropriately skeptical attitude. The class will draw on historical and modern literature on linguistic, logical, and probabilistic fallacies, statistical and logical inference, data visualization, cognitive biases, and the scientific method.
Prerequisite(s): (EN.553.171 OR EN.553.172) OR AS.150.118
Area: Quantitative and Mathematical Sciences, Social and Behavioral Sciences

EN.601.280. Full-Stack JavaScript. 3 Credits.
A full-stack JavaScript developer is a person who can build modern software applications using primarily the JavaScript programming language. Creating a modern software application involves integrating many technologies - from creating the user interface to saving information in a database and everything else in between and beyond. A full-stack developer is not an expert in everything. Rather, they are someone who is familiar with various (software application) frameworks and the ability to take a concept and turn it into a finished product. This course will teach you programming in JavaScript and introduce you to several JavaScript frameworks that would enable you to build modern web, cross-platform desktop, and native/hybrid mobile applications. A student who successfully completes this course will be on the expedited path to becoming a full-stack JavaScript developer.
Prerequisite(s): EN.601.220 OR EN.601.226
Area: Engineering

EN.601.290. User Interfaces and Mobile Applications. 3 Credits.
This course will provide students with a rich development experience, focused on the design and implementation of user interfaces and mobile applications. A brief overview of human computer interaction will provide context for designing, prototyping and evaluating user interfaces. Students will invent their own mobile applications and implement them using the Android SDK, which is JAVA based. An overview of the Android platform and available technologies will be provided, as well as XML for layouts, and general concepts for effective mobile development. Students will be expected to explore and experiment with outside resources in order to learn technical details independently. There will also be an emphasis on building teamwork skills, and on using modern development techniques and tools.
Prerequisite(s): EN.600.120 AND EN.600.226
Area: Engineering

EN.601.295. Developing Health IT Applications. 3 Credits.
This course is a project-based introduction to working on successful projects in health care. In the first half of the term, students perform reading and homework assignments designed to introduce: (1) the context of health care delivery and health IT, (2) techniques to overcome challenges to conducting health care data analyses, and (3) techniques to design meaningful applications around health care data. In the second half of the term, students work in small groups to solve a real-world problem of their choosing. Includes exercises in written and oral communication and team building. [Oral starting 2019]
Prerequisite(s): (EN.600.120 OR EN.601.220) AND (EN.600.226 OR EN.601.226)
Area: Engineering

EN.601.310. Software for Resilient Communities. 3 Credits.
This is a project-based course focusing on the design and implementation of practical software systems. Students will work in small teams to design and develop useful open-source software products that support our communities. Students will be paired with community partners and will aim to develop software that can be used after the course ends to solve real problems facing those partners today. Instructors will connect with the community partners and determine viable project areas prior to the course start. Students will meet with their community partners to analyze the challenges in their project area, agree on a concrete target project outcome, and gather requirements for their project. Based on these requirements, students will design and implement open-source software systems. [Oral]
Prerequisite(s): EN.600.120/EN.601.220 AND EN.600.226/EN.601.226
Area: Engineering

EN.601.315. Databases. 3 Credits.
Introduction to database management systems and database design, focusing on the relational and object-oriented data models, query languages and query optimization, transaction processing, parallel and distributed databases, recovery and security issues, commercial systems and case studies, heterogeneous and multimedia databases, and data mining. [Systems]
Prerequisite(s): EN.600.226/EN.601.226;Students may receive credit for only one of EN.600.315, EN.600.415, EN.601.315, EN.601.415, EN.601.615.
Area: Engineering

EN.601.318. Operating Systems. 3 Credits.
This course covers fundamental topics related to operating systems theory and practice. Topics include processor management, storage management, concurrency control, multi-programming and processing, device drivers, operating system components (e.g., file system, kernel), modeling and performance measurement, protection and security, and recent innovations in operating system structure. Course work includes the implementation of operating systems techniques and routines, and critical parts of a small but functional operating system.
Prerequisite(s): EN.600.120/EN.601.220 AND EN.600.226/EN.601.226 AND EN.600.233/EN.601.229;Students may receive credit for only one of EN.600.318, EN.600.418, EN.601.318, EN.601.418, EN.601.618.
Area: Engineering

EN.601.320. Parallel Programming. 3 Credits.
This course prepares the programmer to tackle the massive data sets and huge problem size of modern scientific and enterprise computing. Google and IBM have commented that undergraduate CS majors are unable to "break the single server mindset" (http://www.google.com/intl/en/ press/pressrel/20071008_ibm_univ.html). Students taking this course will abandon the comfort of serial algorithmic thinking and learn to harness the power of cutting-edge software and hardware technologies. The issue of parallelism spans many architectural levels. Even ``single server'' systems must parallelize computation in order to exploit the inherent parallelism of recent multi-core processors. The course will examine different forms of parallelism in four sections. These are: (1) massive data-parallel computations with Hadoop!; (2) programming compute clusters with MPI; (3) thread-level parallelism in Java; and, (4) GPGPU parallel programming with NVIDIA's Cuda. Each section will be approximately 3 weeks and each section will involve a programming project. The course is also suitable for undergraduate and graduate students from other science and engineering disciplines that have prior programming experience. [Systems]
Prerequisite(s): EN.600.226/EN.601.226 AND EN.600.233/EN.601.229;Students may receive credit for only one of EN.600.320, EN.600.420, EN.601.320, EN.601.420, EN.601.620.
Area: Engineering

EN.601.340. Web Security. 3 Credits.
This course begins with reviewing basic knowledge of the World Wide Web, and then exploring the central defense concepts behind Web security, such as same-origin policy, cross-origin resource sharing, and browser sandboxing. It will cover the most popular Web vulnerabilities, such as cross-site scripting (XSS) and SQL injection, as well as how to attack and penetrate software with such vulnerabilities. Students will learn how to detect, respond, and recover from security incidents. Newly proposed research techniques will also be discussed. [Systems]
Prerequisite(s): (EN.600.226 OR EN.601.226) AND (EN.600.233 OR EN.601.229)
Area: Engineering

EN.601.350. Genomic Data Science. 3 Credits.
This course will use a project-based approach to introduce undergraduates to research in computational biology and genomics. During the semester, students will take a series of large data sets, all derived from recent research, and learn all the computational steps required to convert raw data into a polished analysis. Data challenges might include the DNA sequences from a bacterial genome project, the RNA sequences from an experiment to measure gene expression, the DNA from a human microbiome sequencing experiment, and others. Topics may vary from year to year. In addition to computational data analysis, students will learn to do critical reading of the scientific iterature by reading high-profile research papers that generated groundbreaking or controversial results. [Applications] Recommended Course Background: Knowledge of the Unix operating system and programming expertise in a language such as Perl or Python.
Area: Engineering

EN.601.356. Seminar: Computer Integrated Surgery II. 1 Credit.
Students may receive credit for EN.601.456 or EN.601.356, but not both. Lecture only version of EN.601.456 (no project). Recommended Course Background: EN.601.455 or instructor permission required.
Prerequisite(s): EN.601.455 or instructor permission.;Students may receive credit for either EN.601.356 orEN.601.456, but not both.
Area: Engineering

EN.601.402. Digital Health and Biomedical Informatics. 1 Credit.
Advances in technology are driving a change in medicine, from personalized medicine to population health. Computers and information technology will be critical to this transition. We shall discuss some of the coming changes in terms of computer technology, including computer-based patient records, clinical practice guidelines, and region-wide health information exchanges. We will discuss the underlying technologies driving these developments - databases and warehouses, controlled vocabularies, and decision support.
Area: Engineering

EN.601.411. Computer Science Innovation & Entrepreneurship II. 3 Credits.
This course is the second half of a two-course sequence and is a continuation of course EN.660.410.01, CS Innovation and Entrepreneurship, offered by the Center for Leadership Education (CLE). In this sequel course the student groups, directed by CS faculty, will implement the business idea which was developed in the first course and will present the implementations and business plans to an outside panel made up of practitioners, industry representatives, and venture capitalists. [General]
Prerequisite(s): EN.660.410
Area: Engineering

EN.601.414. Computer Networks. 3 Credits.
Topics covered will include application layer protocols (e.g. HTTP, FTP, SMTP), transport layer protocols (UDP, TCP), network layer protocols (e.g. IP, ICMP), link layer protocols (e.g. Ethernet) and wireless protocols (e.g. IEEE 802.11). The course will also cover routing protocols such as link state and distance vector, multicast routing, and path vector protocols (e.g. BGP). The class will examine security issues such as firewalls and denial of service attacks. We will also study DNS, NAT, Web caching and CDNs, peer to peer, and protocol tunneling. Finally, we will explore security protocols (e.g. TLS, SSH, IPsec), as well as some basic cryptography necessary to understand these. Grading will be based on hands-on programming assignments, homeworks and two exams. [Systems]
Prerequisite(s): EN.601.226 AND EN.601.229 or permission.;Students may receive credit for only one of EN.600.344, EN.600.444, EN.601.414, EN.601.614.
Area: Engineering

EN.601.415. Databases. 3 Credits.
Similar material as EN.601.315 covered in more depth for advanced undergraduates. Introduction to database management systems and database design, focusing on the relational and object-oriented data models, query languages and query optimization, transaction processing, parallel and distributed databases, recovery and security issues, commercial systems and case studies, heterogeneous and multimedia databases, and data mining. [Systems] (www.cs.jhu.edu/~yarowsky/cs415.html)
Prerequisite(s): EN.600.226/EN.601.226;Students may receive credit for only one of EN.600.315, EN.600.415, EN.601.315, EN.601.415, EN.601.615.
Area: Engineering

EN.601.417. Distributed Systems. 3 Credits.
Graduate version of 601.317 Systems. Students may receive credit for 601.317 or 601.417 but not both. Recommended Course Background: EN.601.220, EN.601.226
Prerequisite(s): Students may receive credit for only one of 417/617; (EN.600.120 OR EN.601.220) AND (EN.600.226 OR EN.601.226)
Area: Engineering

EN.601.418. Operating Systems. 3 Credits.
Similar material as EN.601.318, covered in more depth. Intended for advanced undergraduate students. This course covers fundamental topics related to operating systems theory and practice. Topics include processor management, storage management, concurrency control, multi-programming and processing, device drivers, operating system components (e.g., file system, kernel), modeling and performance measurement, protection and security, and recent innovations in operating system structure. Course work includes the implementation of operating systems techniques and routines, and critical parts of a small but functional operating system.
Prerequisite(s): EN.600.120/EN.601.220 AND EN.600.226/EN.601.226 AND EN.600.233/EN.601.229;Students may receive credit for only one of EN.600.318, EN.600.418, EN.601.318, EN.601.418, EN.601.618.
Area: Engineering

EN.601.419. Cloud Computing. 3 Credits.
Clouds host a wide range of the applications that we rely on today. In this course, we study common cloud applications, traffic patterns that they generate, critical networking infrastructures that support them, and core networking and distributed systems concepts, algorithms, and technologies used inside clouds. We will also study how today's application demand is influencing the network's design, explore current practice, and how we can build future's networked infrastructure to better enable both efficient transfer of big data and low-latency requirements of real-time applications. The format of this course will be a mix of lectures, discussions, assignments, and a project designed to help students practice and apply the theories and techniques covered in the course. [Systems] Prerequisites: EN.601.226 or permission. Students can only receive credit for one of 601.419/619. Recommended: a course in operating systems, networks or systems programming.
Prerequisite(s): EN.601.226 (or EN.600.226) AND EN.601.414 or permission from the instructor.;Students may earn credit for EN.601.419 or EN.601.619, but not both.
Area: Engineering

EN.601.420. Parallel Computing for Data Science. 3 Credits.
This course studies parallelism in data science, drawing examples from data analytics, statistical programming, and machine learning. It focuses mostly on the Python programming ecosystem but will use C/C++ to accelerate Python and Java to explore shared-memory threading. It explores parallelism at all levels, including instruction level parallelism (pipelining and vectorization), shared-memory multicore, and distributed computing. Concepts from computer architecture and operating systems will be developed in support of parallelism, including Moore's law, the memory hierarchy, caching, processes/threads, and concurrency control. The course will cover modern data-parallel programming frameworks, including Dask, Spark, Hadoop!, and Ray. The course will not cover GPU deep-learning frameworks nor CUDA. The course is suitable for second-year undergraduate CS majors and graduate students from other science and engineering disciplines that have prior programming experience and familiarity with Python. [Systems]
Prerequisite(s): EN.601.226 AND EN.601.229;Students may receive credit for only one of EN.600.320, EN.600.420, EN.601.320, EN.601.420, EN.601.620.
Area: Engineering

EN.601.421. Object Oriented Software Engineering. 3 Credits.
This course covers object-oriented software construction methodologies and their application. The main component of the course is a large team project on a topic of your choosing. Course topics covered include object-oriented analysis and design, UML, design patterns, refactoring, program testing, code repositories, team programming, and code reviews. [Systems or Applications]
Prerequisite(s): EN.601.220 AND EN.601.226 AND (EN.601.280 OR EN.601.290);Students may receive credit for only one of EN.600.321, EN.600.421, EN.601.421, EN.601.621.
Area: Engineering

EN.601.422. Software Testing & Debugging. 3 Credits.
Studies show that testing can account for over 50% of software development costs. This course presents a comprehensive study of software testing, principles, methodologies, tools, and techniques. Topics include testing principles, coverage (graph coverage, logic coverage, input space partitioning, and syntax-based coverage), unit testing, higher-order testing (integration, system-level, acceptance), testing approaches (white-box, black-box, grey-box), regression testing, debugging, delta debugging, and several specific types of functional and non-functional testing as schedule/interest permits (GUI testing, usability testing, security testing, load/performance testing, A/B testing etc.). For practical topics, state- of-the-art tools/techniques will be studied and utilized. [Systems]
Prerequisite(s): EN.601.290 OR EN.601.421;Students can take EN.601.422 or EN.601.622, but not both.
Area: Engineering

EN.601.424. Reliable Software Systems. 3 Credits.
Reliability is an essential quality requirement for all artifacts operating in the real-world, ranging from bridges, cars to power grids. Software systems are no exception. In this computing age when software is transforming even traditional mission-critical artifacts, making sure the software we write is reliable becomes ever more important. This course exposes students to the principles and techniques in building reliable systems. We will study a set of systematic approaches to make software more robust. These include but are not limited to static analysis, testing framework, model checking, symbolic execution, fuzzing, and formal verification. In addition, we will cover the latest research in system reliability.
Prerequisite(s): EN.601.220 AND (EN.601.328 OR EN.601.428)
Area: Engineering

EN.601.426. Principles of Programming Languages. 3 Credits.
Functional, object-oriented, and other language features are studied independent of a particular programming language. Students become familiar with these features by implementing them. Most of the implementations are in the form of small language interpreters. Some type checkers and a small compiler will also be written. The total amount of code written will not be overly large, as the emphasis is on concepts. The ML programming language is the implementation language used. [Analysis] Prerequisites include EN.601.226. No Freshmen or Sophomores.
Prerequisite(s): EN.601.226
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.427. Principles of Programming Languages II. 3 Credits.
This course is designed as a follow-on to Principles of Programming languages. It will cover a wide array of fundamental topics in programming languages, including advanced functional programming, the theory of inductive definitions, advanced operational semantics, advanced type systems, program analysis, program verification, theorem provers and SAT solvers. [Analysis]
Prerequisite(s): EN.601.426
Area: Engineering

EN.601.428. Compilers & Interpreters. 3 Credits.
Introduction to compiler design, including lexical analysis, parsing, syntax-directed translation, symbol tables, runtime environments, and code generation and optimization. Students are required to write a compiler as a course project. [Systems]
Prerequisite(s): EN.600.120/EN.601.220 AND EN.600.226/EN.601.226 AND EN.600.233/EN.601.229
Area: Engineering

EN.601.429. Functional Programming in Software Engineering. 3 Credits.
How can we effectively use functional programming techniques to build real-world software? This course will primarily focus on using the OCaml programming language for this purpose. Topics covered include OCaml basics, modules, standard libraries, testing, quickcheck, build tools, functional data structures and efficiency analysis, monads, streams, and promises. Students will practice what they learn in lecture via functional programming assignments and a final project.
Prerequisite(s): EN.601.226 OR Instructor Permission
Area: Engineering

EN.601.430. Combinatorics & Graph Theory in Computer Science. 3 Credits.
This is a graduate level course studying the applications of combinatorics and graph theory in computer science. We will start with some basic combinatorial techniques such as counting and pigeon hole principle, and then move to advanced techniques such as the probabilistic method, spectral graph theory and additive combinatorics. We shall see their applications in various areas in computer science, such as proving lower bounds in computational models, randomized algorithms, coding theory and pseudorandomness. [Analysis]Recommended Course Background: probability theory and linear algebra
Prerequisite(s): EN.553.171 OR EN.553.172 OR EN.550.171; probability theory and linear algebra recommended.;Students may receive credit for only one of 430/630
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.433. Intro Algorithms. 3 Credits.
This course concentrates on the design of algorithms and the rigorous analysis of their efficiency. topics include the basic definitions of algorithmic complexity (worst case, average case); basic tools such as dynamic programming, sorting, searching, and selection; advanced data structures and their applications (such as union-find); graph algorithms and searching techniques such as minimum spanning trees, depth-first search, shortest paths, design of online algorithms and competitive analysis. [Analysis]
Prerequisite(s): EN.601.226 AND (EN.553.171 OR EN.553.172 OR EN.601.230 OR EN.601.231);Students may receive credit for only one of EN.600.363, EN.600.463, EN.601.433, EN.601.633.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.434. Randomized and Big Data Algorithms. 3 Credits.
The course emphasizes algorithmic design aspects, and how randomization can be a helpful tool. The topics covered include: tail inequalities, linear programming relaxation & randomized rounding, de-randomization, existence proofs, universal hashing, markov chains, metropolis and metropolis-hastings methods, mixing by coupling and by eigenvalues, counting problems, semi-definite programming and rounding, lower bound arguments, and applications of expanders. [Analysis] (www.cs.jhu.edu/~cs464) Recommended Course Background: Probability
Prerequisite(s): ((N.600.363 OR EN.600.463) OR (EN.601.433 OR EN.601.633)) AND (EN.550.310 OR EN.553.310 OR EN.553.311 OR EN.550.420 OR EN.550.620) or equivalent.;Students may receive credit for only one of EN.600.464, EN.600.664, EN.601.434, EN.601.634.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.435. Approximation Algorithms. 3 Credits.
This course provides an introduction to approximation algorithms. Topics include vertex cover, TSP, Steiner trees, cuts, greedy approach, linear and semi-definite programming, primal-dual method, and randomization. Additional topics will be covered as time permits. There will be a final project. Students may receive credit for EN.601.435 or EN.601.635, but not both. [Analysis]
Prerequisite(s): EN.600.363 OR EN.601.433 OR EN.601.633 OR permission.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.436. Algorithmic Game Theory. 3 Credits.
This course provides an introduction to algorithmic game theory: the study of games from the perspective of algorithms and theoretical computer science. There will be a particular focus on games that arise naturally from economic interactions involving computer systems (such as economic interactions between large-scale networks, online advertising markets, etc.), but there will also be broad coverage of games and mechanisms of all sorts. Topics covered will include a) complexity of computing equilibria and algorithms for doing so, b) (in)efficiency of equilibria, and c) algorithmic mechanism design. [Analysis]
Prerequisite(s): EN.600.363 OR EN.600.463 OR EN.601.433 OR EN.601.633
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.437. Federated Learning and Analytics. 3 Credits.
Federated Learning (FL) is an area of machine learning where data is distributed across multiple devices and training is performed without exchanging the data between devices. FL can be contrasted with classical machine learning settings when data is available in a central location. As such, FL faces additional challenges and limitations such as privacy and communication. For example, FL may deal with questions of learning from sensitive data on mobile devices while protecting privacy of individual users and dealing with low power and limited communication. As a result, FL requires knowledge of many interdisciplinary areas such as differential privacy, distributed optimization, sketching algorithms, compression and more. In this course students will learn basic concepts and algorithms for FL and federated analytics, and gain hands-on experience with new methods and techniques. Students will gain understanding in reasoning about possible trade-offs between privacy, accuracy and communication. [Analysis] ML: DL, linear algebra, probability
Prerequisite(s): EN.601.433/EN.601.633 AND (EN.601.464/EN.601.664 OR EN.601.475/EN.601.675);Students may only earn credit for EN.601.437 OR EN.601.637.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.440. Web Security. 3 Credits.
This course begins with reviewing basic knowledge of the World Wide Web, and then exploring the central defense concepts behind Web security, such as same-origin policy, cross-origin resource sharing, and browser sandboxing. It will cover the most popular Web vulnerabilities, such as cross-site scripting (XSS) and SQL injection, as well as how to attack and penetrate software with such vulnerabilities. Students will learn how to detect, respond, and recover from security incidents. Newly proposed research techniques will also be discussed. [Systems]
Prerequisite(s): (EN.601.226 OR EN.600.226) AND (EN.601.229 OR EN.600.233);Students may receive credit for only one of 340/440/640.
Area: Engineering

EN.601.441. Blockchains and Cryptocurrencies. 3 Credits.
This course will introduce students to cryptocurrencies and the main underlying technology of Blockchains. The course will start with the relevant background in cryptography and then proceed to cover the recent advances in the design and applications of blockchains. This course should primarily appeal to students who want to conduct research in this area or wish to build new applications on top of blockchains. It should also appeal to those who have a casual interest in this topic or are generally interested in cryptography. Students are expected to have mathematical maturity. [Analysis]
Prerequisite(s): EN.601.226 AND (EN.553.211 OR EN.553.310 OR EN.553.311 OR EN.560.348 OR EN.553.420);Students may recieve credit for only one of EN.600.451 OR EN.601.441 OR EN.601.641
Area: Engineering

EN.601.442. Modern Cryptography. 3 Credits.
Modern Cryptography includes seemingly paradoxical notions such as communicating privately without a shared secret, proving things without leaking knowledge, and computing on encrypted data. In this challenging but rewarding course we will start from the basics of private and public key cryptography and go all the way up to advanced notions such as zero-knowledge proofs, functional encryption and program obfuscation. The class will focus on rigorous proofs and require mathematical maturity. [Analysis]
Prerequisite(s): Students may receive credit for only one of EN.600.442, EN.601.442, EN.601.642.;(EN.601.230 OR EN.601.231) AND (EN.553.310 OR EN.553.311 OR EN.553.420 OR EN.553.421)
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.443. Security & Privacy in Computing. 3 Credits.
Lecture topics will include computer security, network security, basic cryptography, system design methodology, and privacy. There will be a heavy work load, including written homework, programming assignments, exams and a comprehensive final. The class will also include a semester-long project that will be done in teams and will include a presentation by each group to the class. [Applications] Recommended Course Background: A basic course in operating systems and networking, or permission of instructor.
Prerequisite(s): Students may receive credit for only one of EN.600.443, EN.601.443, EN.601.643.;(EN.600.318/EN.601.318 OR EN.600.418/EN.601.418) OR (EN.600.344 OR EN.600.444/EN.601.414) AND (EN.600.233 OR EN.601.229)
Area: Engineering

EN.601.444. Network Security. 3 Credits.
This course focuses on communication security in computer systems and networks. The course is intended to provide students with an introduction to the field of network security. The course covers network security services such as authentication and access control, integrity and confidentiality of data, firewalls and related technologies, Web security and privacy. Course work involves implementing various security techniques. A course project is required. [Systems]
Prerequisite(s): EN.600.120 AND EN.600.226 AND (EN.600.344 OR EN.600.444) or permission;Students may receive credit for only one of EN.600.424, EN.650.424, EN.601.444, EN.601.644.
Area: Engineering

EN.601.445. Practical Cryptographic Systems. 3 Credits.
This semester-long course will teach systems and cryptographic design principles by example: by studying and identifying flaws in widely-deployed cryptographic products and protocols. Our focus will be on the techniques used in practical security systems, the mistakes that lead to failure, and the approaches that might have avoided the problem. We will place a particular emphasis on the techniques of provable security and the feasibility of reverse-engineering undocumented cryptographic systems. [Systems]
Prerequisite(s): Students may receive credit for only one of EN.600.454, EN.601.445, EN.601.645.;EN.600.226/EN.601.226 AND EN.600.233/EN.601.229
Area: Engineering

EN.601.446. Sketching and Indexing for Sequences. 3 Credits.
Many of the world's largest and fastest-growing datasets are text, e.g. DNA sequencing data, web pages, logs and social media posts. Such datasets are useful only to the degree we can query, compare and analyze them. Here we discuss two powerful approaches in this area. We will cover sketching, which enables us to summarize very large texts in small structures that allow us to measure the sizes of sets and of their unions and intersections. This in turn allows us to measure similarity and find near neighbors. Second, we will discuss indexing --- succinct and compressed indexes in particular -- which enables us to efficiently search inside very long strings, especially in highly repetitive texts. [Analysis]
Prerequisite(s): EN.601.226
Area: Engineering

EN.601.447. Computational Genomics: Sequences. 3 Credits.
Your genome is the blueprint for the molecules in your body. It's also a string of letters (A, C, G and T) about 3 billion letters long. How does this string give rise to you? Your heart, your brain, your health? This, broadly speaking, is what genomics research is about. This course will familiarize you with a breadth of topics from the field of computational genomics. The emphasis is on current research problems, real-world genomics data, and efficient software implementations for analyzing data. Topics will include: string matching, sequence alignment and indexing, assembly, and sequence models. Course will involve significant programming projects. [Applications]
Prerequisite(s): EN.600.120/EN.601.220 AND EN.600.226/EN.601.226;Students may receive credit for only one of EN.600.439, EN.600.639, EN.601.447, EN.601.647.
Area: Engineering

EN.601.448. Computational Genomics: Data Analysis. 3 Credits.
Genomic data has the potential to reveal causes of disease, novel drug targets, and relationships among genes and pathways in our cells. However, identifying meaningful patterns from high-dimensional genomic data has required development of new computational tools. This course will cover current approaches in computational analysis of genomic data with a focus on statistical methods and machine learning.Topics will include disease association, prediction tasks, clustering and dimensionality reduction, data integration, and network reconstruction. There will be some programming and a project component. [Applications]Prerequisites: EN.601.226 or other programming experience, probability and statistics, linear algebra or calculus.
Prerequisite(s): Students may receive credit for only one of EN.600.438, EN.600.638, EN.601.448, EN.601.648.
Area: Engineering

EN.601.449. Computational Genomics: Applied Comparative Genomics. 3 Credits.
The goal of this course is to study the leading computational and quantitative approaches for comparing and analyzing genomes starting from raw sequencing data. The course will focus on human genomics and human medical applications, but the techniques will be broadly applicable across the tree of life. The topics will include genome assembly & comparative genomics, variant identification & analysis, gene expression & regulation, personal genome analysis, and cancer genomics. The grading will be based on assignments, a midterm exam, class presentations, and a significant class project. Prerequisites: knowledge of the Unix operating system and programming expertise in a language such as R or Python. [Applications]
Prerequisite(s): Students may receive credit for only one of EN.600.449, EN.600.649, EN.601.749.
Area: Engineering

EN.601.451. Introduction to Computational Immunogenomics. 3 Credits.
Immunology studies defensive mechanisms of living organisms against external threats. Computational immunogenomics is a new field of bioinformatics that develops and applies computational approaches to the study and interpretation of immunological data, seeking to answer questions about adaptive immune responses in humans and important animals. In this course, students will learn how to design, apply, and benchmark algorithms for solving immunogenomics problems. [Applications] Students may receive credit for only one of EN.601.451, EN.601.651.
Prerequisite(s): EN.601.220 AND EN.601.226;Students may receive credit for only one of EN.601.451 OR EN.601.651.
Area: Engineering

EN.601.452. Computational Biomedical Research. 3 Credits.
[Co-listed with AS.020.415] This course for advanced undergraduates includes classroom instruction in interdisciplinary research approaches and lab work on an independent research project in the lab of a Bloomberg Distinguished Professor and other distinguished faculty. Lectures will focus on cross-cutting techniques such as data visualization, statistical inference, and scientific computing. In addition to two 50-minute classes per week, students will commit to working approximately 3 hours per week in the lab of one of the professors. The student and professor will work together to schedule the research project. Students will present their work at a symposium at the end of the semester.
Area: Engineering

EN.601.454. Augmented Reality. 3 Credits.
Same as EN.601.654, for undergraduate students. This course introduces students to the field of Augmented Reality. It reviews its basic definitions, principles and applications. It then focuses on Medical Augmented Reality and its particular requirements. The course also discusses the main issues of calibration, tracking, multi-modal registration, advance visualization and display technologies. Homework in this course will relate to the mathematical methods used for calibration, tracking and visualization in medical augmented reality. [Applications]
Prerequisite(s): EN.601.220 AND EN.601.226 AND (AS.110.201 OR AS.110.212 OR EN.550.291);Students may receive credit for only one of EN.600.484, EN.600.684, EN.601.454, EN.601.654.
Area: Engineering

EN.601.455. Computer Integrated Surgery I. 4 Credits.
This course focuses on computer-based techniques, systems, and applications exploiting quantitative information from medical images and sensors to assist clinicians in all phases of treatment from diagnosis to preoperative planning, execution, and follow-up. It emphasizes the relationship between problem definition, computer-based technology, and clinical application and includes a number of guest lectures given by surgeons and other experts on requirements and opportunities in particular clinical areas. Recommended Course Background: EN.601.220, EN.601.457, EN.601.461, image processing.
Prerequisite(s): Students may receive credit for only one of EN.600.445, EN.600.645, EN.601.455, EN.601.655.;EN.600.226/EN.601.226 AND (AS.110.201 OR AS.110.212 OR EN.553.291) or permission of the instructor.
Area: Engineering

EN.601.456. Computer Integrated Surgery II. 3 Credits.
This weekly lecture/seminar course addresses similar material to EN.601.455, but covers selected topics in greater depth. In addition to material covered in lectures/seminars by the instructor and other faculty, students are expected to read and provide critical analysis/presentations of selected papers in recitation sessions. Students taking this course are required to undertake and report on a significant term project under the supervision of the instructor and clinical end users. Typically, this project is an extension of the term project from EN.601.455, although it does not have to be. Grades are based both on the project and on classroom recitations. Students who wish to use this course to satisfy the "Team" requirement should register for EN.601.496 instead. Students wishing to attend the weekly lectures as a 1-credit seminar should sign up for EN.601.356. [Applications, Oral]
Prerequisite(s): EN.601.455 or EN.601.655 or permisssion;Students may receive credit for only one of EN.600.446, EN.600.646, EN.601.456, EN.601.656.
Area: Engineering

EN.601.457. Computer Graphics. 3 Credits.
This course introduces computer graphics techniques and applications, including image processing, rendering, modeling and animation. [Applications]
Prerequisite(s): EN.600.120/601.220 AND EN.600.226/601.226, and linear algebra or permission of instructor.;Students may receive credit for only one of EN.600.357, EN.600.457, EN.601.457, EN.601.657.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.459. Computational Geometry. 3 Credits.
This course will provide an introduction to computational geometry. It will cover a number of topics in two- and three-dimensions, including polygon triangulations and partitions, convex hulls, Delaunay and Voronoi diagrams, arrangements, and spatial queries. Time-permitting, we will also look at kD-trees, general BSP-trees, and quadtrees. [Analysis]
Prerequisite(s): EN.600.120/EN.601.220 AND EN.600.226/EN.601.226 AND (EN.600.363 OR EN.600.463/EN.601.433 OR EN.601.633)
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.461. Computer Vision. 3 Credits.
This course provides an overview of fundamental methods in computer vision from a computational perspective. Methods studied include: camera systems and their modelling, computation of 3-D geometry from binocular stereo, motion, and photometric stereo, and objectrecognition, image segmentation, and activity analysis. Elements of machine vision and biological vision are also included.
Prerequisite(s): Students may receive credit for only one of EN.600.361, EN.600.461, EN.600.661, EN.601.461, EN.601.661.;(EN.553.310 OR EN.553.311 OR ((EN.553.420 OR EN.553.421) AND (EN.553.430 OR EN.553.431)) OR EN.560.348) AND (AS.110.201 OR AS.110.212 OR EN.553.291) AND (EN.500.112 OR EN.500.113 OR EN.500.114 OR EN.601.220 AS.250.205)
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.462. Introduction to Spatial Computing. 3 Credits.
his course will provide students with a rich understanding of immersive technology and spatial computing, including virtual, augmented, and mixed reality as the next wave of computing after personal and mobile computing, and belongs to the systems that can sense the space or are “spatially” aware. It also covers input systems and interaction modalities that have evolved to support human-computer interaction required for immersive environments. It will go through principles of design thinking including verification and validation, as a mindset for creating immersive experiences, and students will explore the practical implication of this subject in healthcare, industry, and society through the projects.
Prerequisite(s): EN.601.220 AND EN.601.226 AND (AS.110.201 OR AS.110.212 OR EN.553.291)
Area: Engineering

EN.601.463. Algorithms for Sensor-Based Robotics. 3 Credits.
This course surveys the development of robotic systems for navigating in an environment from an algorithmic perspective. It will cover basic kinematics, configuration space concepts, motion planning, and localization and mapping. It will describe these concepts in the context of the ROS software system, and will present examples relevant to mobile platforms, manipulation, robotics surgery, and human-machine systems. [Analysis]
Prerequisite(s): (AS.110.201 OR AS.110.212) AND AS.110.202 AND EN.601.226;Students may receive credit for only one of EN.600.336, EN.600.436, EN.600.636, EN.601.463, EN.601.663.;(EN.553.310 OR EN.553.311 OR EN.553.420)
Area: Engineering

EN.601.464. Artificial Intelligence. 3 Credits.
This course is recommended for scientists and engineers with a genuine curiosity about the fundamental obstacles in getting machines to perform tasks such as deduction, learning, planning and navigation. It covers methods for automated reasoning, automatic problem solversand planners, knowledge representation mechanisms, game playing, machine learning,and statistical pattern recognition. Strong programming skills are expected, as well asbasic familiarity with probability. Students intending to also take courses in machinelearning (e.g. 601.475/675, 601.476/676, 601.482/682) may find it beneficial to take thiscourse first, or concurrently. [Applications]
Prerequisite(s): Students may receive credit for only one of EN.600.335, EN.600.435, EN.601.464, EN.601.664.;EN.600.226/EN.601.226
Area: Engineering

EN.601.465. Natural Language Processing. 4 Credits.
This course is an in-depth overview of techniques for processing human language. How should linguistic structure and meaning be represented? What algorithms can recover them from text? And crucially, how can we build statistical models to choose among the many legal answers? The course covers methods for trees (parsing and semantic interpretation), sequences (finite-state transduction such as morphology), and words (sense and phrase induction), with applications to practical engineering tasks such as information retrieval and extraction, text classification, part-of-speech tagging, speech recognition and machine translation. There are a number of structured but challenging programming assignments. [Applications]
Prerequisite(s): Students may receive credit for only one of EN.600.465, EN.601.465, EN.601.665.;EN.600.226/EN.601.226
Area: Engineering

EN.601.466. Information Retrieval and Web Agents. 3 Credits.
An in-depth, hands-on study of current information retrieval techniques and their application to developing intelligent WWW agents. Topics include a comprehensive study of current document retrieval models, mail/news routing and filtering, document clustering, automatic indexing, query expansion, relevance feedback, user modeling, information visualization and usage pattern analysis. In addition, the course explores the range of additional language processing steps useful for template filling and information extraction from retrieved documents, focusing on recent, primarily statistical methods. The course concludes with a study of current issues in information retrieval and data mining on the World Wide Web. Topics include web robots, spiders, agents and search engines, exploring both their practical implementation and the economic and legal issues surrounding their use. Recommended Course Background: EN.601.226
Prerequisite(s): EN.600.226 OR EN.601.226
Area: Engineering

EN.601.467. Introduction to Human Language Technology. 3 Credits.
This course gives an overview of basic foundations and applications of human language technology, such as: morphological, syntactic, semantic, and pragmatic processing; machine learning; signal processing; speech recognition; speech synthesis; information retrieval; text classification; topic modelling; informationextraction; knowledge representation; machine translation; dialog systems; etc. [Applications]Pre-req: EN.601.226 Data Structures; knowledge of Python recommended.
Prerequisite(s): EN.601.226 OR EN.600.226
Area: Engineering

EN.601.468. Machine Translation. 3 Credits.
Google translate can instantly translate between any pair of over fifty human languages (for instance, from French to English). How does it do that? Why does it make the errors that it does? And how can you build something better? Modern translation systems learn to translate by reading millions of words of already translated text, and this course will show you how they work. The course covers a diverse set of fundamental building blocks from linguistics, machine learning, algorithms, data structures, and formal language theory, along with their application to a real and difficult problem in artificial intelligence.
Prerequisite(s): Students may receive credit for only one of EN.600.468, EN.601.468, EN.601.668.;EN.600.226/EN.601.226 and prob/stat.
Area: Engineering

EN.601.470. Artificial Agents. 3 Credits.
This course covers a number of topics explored in introductory AI, such as knowledge representation, reasoning, and natural language understanding. Unlike introductory AI, we will pursue these topics based on the transformer neural architecture. We will motivate the material through interacting with agents in games: how to build models that understand user commands, how to generate responses back to a user, and how to reason about a synthetic environment to determine a course of action. Assignments will include programming, presentations on readings, and written summaries of readings. [Applications]
Prerequisite(s): (EN.601.475 OR EN.601.675) OR (EN.601.482 OR EN.601.682) OR (EN.601.488 OR EN.601.688) OR (EN.601.486 OR EN.601.686)
Area: Engineering

EN.601.474. ML: Learning Theory. 3 Credits.
This is an undergraduate level course in machine learning. It will provide a formal and in-depth coverage of topics in statistical and computational learning theory. We will revisit popular machine learning algorithms and understand their performance in terms of the size of the data (sample complexity), memory needed (space complexity), as well as the overall runtime (computational or iteration complexity). We will cover topics including PAC learning, uniform convergence, VC dimension, Rademacher complexity, algorithmic stability, kernel methods, online learning and reinforcement learning, as well as introduce students to current topics in large-scale machine- learning and randomized projections. General focus will be on combining methodology with theoretical and computational foundations. [Analysis]
Prerequisite(s): AS.110.202 AND ((EN.553.420 AND EN.553.430) OR (EN.553.211 OR EN.553.310 OR EN.553.311) OR EN.560.348) AND (AS.110.201 OR AS.110.212 OR EN.553.291) AND (EN.500.112 OR EN.500.113 OR EN.500.114) OR (EN.601.220 OR AS.250.205 OR EN.580.200 OR EN.601.107)
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.475. Machine Learning. 3 Credits.
Machine learning is subfield of computer science and artificial intelligence, whose goal is to develop computational systems, methods, and algorithms that can learn from data to improve their performance. This course introduces the foundational concepts of modern Machine Learning, including core principles, popular algorithms and modeling platforms. This will include both supervised learning, which includes popular algorithms like SVMs, logistic regression, boosting and deep learning, as well as unsupervised learning frameworks, which include Expectation Maximization and graphical models. Homework assignments include a heavy programming components, requiring students to implement several machine learning algorithms in a common learning framework. Additionally, analytical homework questions will explore various machine learning concepts, building on the pre-requisites that include probability, linear algebra, multi-variate calculus and basic optimization. Students in the course will develop a learning system for a final project. [Analysis or Applications]
Prerequisite(s): Students may receive credit for only one of EN.600.475, EN.601.475, EN.601.675.;Linear Algebra, Probability, Statistics, Calc III, and Intro Computing/Programming - AS.110.202 AND (EN.553.211 OR EN.553.310 OR EN.553.311 OR ((EN.553.420 or EN.553.421) AND (EN.553.430 OR EN.553.431)) OR EN.560.348) AND (AS.110.201 OR AS.110.212 OR EN.553.291) AND (EN.500.112 OR EN.500.113 OR EN.500.114 OR (EN.601.220 OR EN.600.120) OR AS.250.205 OR EN.580.200 OR (EN.600.107 OR EN.601.107)).
Area: Engineering

EN.601.476. Machine Learning: Data to Models. 3 Credits.
How can robots localize themselves in an environment when navigating? Which factors predict whether patients are at greatest-risk for complications in the hospital? Can we reconstruct the brain's "connectome" from fMRI data? Many such big data questions can be answered using the paradigm of probabilistic models in machine learning. This is the second course on machine learning which focuses on probabilistic graphical models. You will learn about directed and undirected graphical models, inference methods, sampling, structure learning algorithms, latent variables, and temporal models. There will be regular assignments, which include theory and some programming. Students will analyze real data for their final project, applying methods discussed in class and writing up a report of their results. [Analysis or Applications] Students may receive credit for EN.600.476 or EN.600.676, but not both.
Prerequisite(s): Students may receive credit for only one of EN.600.476, EN.601.476, EN.601.676.;EN.600.475/EN.601.475 OR EN.600.675/EN.601.675 or equivalent.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.477. Causal Inference. 3 Credits.
"Big data" is not necessarily "high quality data." Systematically missing records, unobserved confounders, and selection effects present in many datasets make it harder than ever to answer scientifically meaningful questions. This course will teach mathematical tools to help you reason about causes, effects, and bias sources in data with confidence. We will use graphical causal models, and potential outcomes to formalize what causal effects mean, describe how to express these effects as functions of observed data, and use regression model techniques to estimate them. We will consider techniques for handling missing values, structure learning algorithms for inferring causal directionality from data, and connections between causal inference and reinforcement learning. [Analysis] Pre-requisites: familiarity with the R programming language, multivariate calculus, basics of linear algebra and probability.
Prerequisite(s): EN.601.475 OR (EN.553.211 OR EN.553.311 OR EN.553.420 OR EN.553.421) AND AS.110.202 or permission of instructor.;Students may receive credit for only one of EN.600.477, EN.600.677, EN.601.477, EN.601.677.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.481. Machine Learning: Optimization. 3 Credits.
Optimization is at the heart of machine learning. Most machine learning problems can be posed as optimization problems. However, unlike mathematical optimization where the focus is on efficient algorithms for finding solutions with a high degree of accuracy as measured by optimality conditions, optimization for machine learning focuses on algorithms that are efficient and generalize well. In this course, we will focus on optimization for problems that arise in machine learning, design and analysis of algorithms for solving these problems, and the interplay of optimization and machine learning. The coursework will include homework assignments and a final project focusing on applying optimization algorithms to real world machinelearning problems. [Analysis or Applications]
Prerequisite(s): EN.601.475 OR (EN.553.211 OR EN.553.310 OR EN.553.311 OR ((EN.553.420 OR EN.553.421) AND (EN.553.430 OR EN.553.431)) AND AS.110.201 AND AS.110.202);Students may receive credit for only one of EN.601.481/681.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.482. Machine Learning: Deep Learning. 4 Credits.
Deep learning (DL) has emerged as a powerful tool for solving data-intensive learning problems such as supervised learning for classification or regression, dimensionality reduction, and control. As such, it has a broad range of applications including speech and text understanding, computer vision, medical imaging, and perception-based robotics. The goal of this course is to introduce the basic concepts of deep learning (DL). The course will include a brief introduction to the basic theoretical and methodological underpinnings of machine learning, commonly used architectures for DL, DL optimization methods, DL programming systems, and specialized applications to computer vision, speech understanding, and robotics. Students will be expected to solve several DL problems on standardized data sets, and will be given the opportunity to pursue team projects on topics of their choice. [Applications]
Prerequisite(s): EN.601.226 AND (AS.110.201 OR AS.110.212 OR EN.553.291) AND (EN.553.310 OR EN.553.311 OR EN.553.420 OR EN.553.420); Python recommended.
Area: Engineering

EN.601.484. ML: Interpretable Machine Learning Design. 3 Credits.
There are considerable research thrusts that seek to increase the trustworthiness and perceived reliability of machine learning solutions. One such thrust, interpretable machine learning, attempts to reveal the working mechanisms of a machine learning system. However, other than on-task performance, interpretability is not a property of machine learning algorithms, but an affordance: a relationship between interpretable model and the target users in their context. Successful development of machine learning solutions that afford interpretation thus requires understanding of techniques beyond pure machine learning. In this course, we will first review the basics of machine learning and human-centered design. Then, during student team-delivered lectures, we will learn about contemporary techniques to introduce interpretability to machine learning models and discuss recent literature on the topic. In addition to hands-on homework assignments, students will work in groups to design, justify, implement, and test an interpretable machine learning algorithm for a problem of their choosing. Recommended background in (601.454/654, 601.290, 601.490/690 or 601.491/691) and 601.477/677, and coding in Python/PyTorch.
Prerequisite(s): (EN.601.476 OR EN.601.476) OR (EN.601.464 OR EN.601.664) OR (EN.601.482 OR EN.601.682)
Area: Engineering

EN.601.486. Machine Learning: Artificial Intelligence System Design & Development. 3 Credits.
The field of artificial intelligence (AI) has recently seen a substantial increase in popularity, largely fueled by the successes of training deep neural networks that achieve state-of-the-art performance in a large variety of problems. These successes are not limited to academic benchmarks but have started to impact our everyday lives in the form of products such as Google Lens, Amazon Alexa, and Tesla Autopilot. In order for such AI systems to succeed we must consider its impact on everyday life, its overall capabilities and performance, and the effectiveness of the human-AI interaction. The importance of harmonic interplay between all these components is dramatically highlighted by recent catastrophic events in road transport and aviation. In this project-based course you will work in teams of 3-5 students to 1) Identify a need with high-impact implications on everyday life; 2) Conceptualize and design an AI system targeting this need, and 3) Develop the AI system by refining a demo-able prototype based on feedback received during course presentations. Required course background: (EN.601.475/675 or EN.601.464/664 or EN.601.482/682) and Python programming. Recommended: 601.290 or 601.454/654 or 601.490/690 or 601.491/691 (experience with human computer interface design).
Prerequisite(s): (EN.601.475 OR EN.601.675) OR (EN.601.464 OR EN.601.664) OR (EN.601.482 OR EN.601.682)
Area: Engineering

EN.601.490. Introduction to Human-Computer Interaction. 3 Credits.
This course is designed to introduce undergraduate and graduate students to design techniques and practices in human-computer interaction (HCI), the study of interactions between humans and computing systems. Students will learn design techniques and evaluation methods, as well as current practices and exploratory approaches, in HCI through lectures, readings, and assignments. Students will practice various design techniques and evaluation methods through hands-on projects focusing on different computing technologies and application domains. This course is intended for undergraduate and graduate students in Computer Science/Cognitive Science/Psychology. Interested students from different disciplines should contact the instructor before enrolling in this course. [Applications]Recommended Background: Basic programming skills.
Prerequisite(s): Students can receive credit for either EN.601.490 or EN.601.690, but not both.
Area: Engineering

EN.601.491. Human-Robot Interaction. 3 Credits.
This course is designed to introduce advanced students to research methods and topics in human-robot interaction (HRI), an emerging research area focusing on the design and evaluation of interactions between humans and robotic technologies. Students will (1) learn design principles for building and research methods of evaluating interactive robot systems through lectures, readings, and assignments, (2) read and discuss relevant literature to gain sufficient knowledge of various research topics in HRI, and (3) work on a substantial project that integrates the principles, methods, and knowledge learned in this course. [Applications]
Prerequisite(s): EN.601.220/EN.600.120 AND EN.601.226/EN.600.226
Area: Engineering

EN.601.496. Computer Integrated Surgery II - Teams. 3 Credits.
This weekly lecture/seminar course addresses similar material to 600.455, but covers selected topics in greater depth. In addition to material covered in lectures/seminars by the instructor and other faculty, students are expected to read and provide critical analysis/presentations of selected papers in recitation sessions. Students taking this course are required to undertake and report on a significant term project in teams of at least 3 students, under the supervision of the instructor and clinical end users. Typically, this project is an extension of the term project from 600.455, although it does not have to be. Grades are based both on the project and on classroom recitations. Students who prefer to do individual projects must register for EN.601.456 instead. [Applications, Oral]
Prerequisite(s): EN.601.455 or permission;Students may receive credit for only one of EN.601.456, EN.601.496, OR EN.601.656
Area: Engineering

EN.601.501. Computer Science Workshop. 1 - 3 Credits.
An applications-oriented, computer science project done under the supervision and with the sponsorship of a faculty member in the Department of Computer Science. Computer Science Workshop provides a student with an opportunity to apply theory and concepts of computer science to a significant project of mutual interest to the student and a Computer Science faculty member. Permission to enroll in CSW is granted by the faculty sponsor after his/her approval of a project proposal from the student. Interested students are advised to consult with Computer Science faculty members before preparing a Computer Science Workshop project proposal.
Prerequisite(s): You must request Independent Academic Work using the Independent Academic Work form found in Student Self-Service: Registration > Online Forms.

EN.601.503. Independent Study. 1 - 3 Credits.
Individual guided study for undergraduate students under the direction of a faculty member in the department. The program of study, including the credit to be assigned, must be worked out in advance between the student and the faculty member involved. Permission required.
Prerequisite(s): You must request Independent Academic Work using the Independent Academic Work form found in Student Self-Service: Registration > Online Forms.

EN.601.507. Undergraduate Research. 1 - 3 Credits.
Individual research for undergraduates under the direction of a faculty member in the department. The program of research, including the credit to be assigned, must be worked out in advance between the student and the faculty member involved. Permission required.
Prerequisite(s): You must request Independent Academic Work using the Independent Academic Work form found in Student Self-Service: Registration > Online Forms.

EN.601.509. Computer Science Internship. 1 Credit.
Individual work in the field with a learning component, supervised by a faculty member in the department. The program of study and credit assigned must be worked out in advance between the student and the faculty member involved. Students may not receive credit for work that they are paid to do. As a rule of thumb, 40 hours of work is equivalent to one credit. Permission required.
Prerequisite(s): You must request Independent Academic Work using the Independent Academic Work form found in Student Self-Service: Registration > Online Forms.

EN.601.517. Group Undergraduate Research. 1 - 3 Credits.
Independent research for undergraduates under the direction of a faculty member in the department. This course has a weekly research group meeting that students are expected to attend. The program of research, including the credit to be assigned, must be worked out in advance between the student and the faculty member involved.
Prerequisite(s): You must request Independent Academic Work using the Independent Academic Work form found in Student Self-Service: Registration > Online Forms.

EN.601.519. Senior Honors Thesis. 3 Credits.
The student will undertake a substantial independent research project under the supervision of a faculty member, potentially leading to the notation "Departmental Honors with Thesis" on the final transcript. Students are expected to enroll in both semesters of this course during their senior year. Project proposals must be submitted and accepted in the preceding spring semester (junior year) before registration. Students will present their work publicly before April 1st of senior year. They will also submit a first draft of their project report (thesis document) at that time. Faculty will meet to decide if the thesis will be accepted for honors. Computer science majors only. Students should have a 3.5 GPA in computer science courses at the end of their junior year and permission of faculty sponsor.
Prerequisite(s): You must request Independent Academic Work using the Independent Academic Work form found in Student Self-Service: Registration > Online Forms.

EN.601.520. Senior Honors Thesis. 1 - 3 Credits.
For computer science majors only, a continuation of EN.601.519.
Recommended Course Background: EN.601.519
Prerequisite(s): You must request Independent Academic Work using the Independent Academic Work form found in Student Self-Service: Registration > Online Forms.

EN.601.556. Senior Thesis In CIS. 3 Credits.
Prerequisite(s): EN.600.445 or permission of instructor.;You must request Independent Academic Work using the Independent Academic Work form found in Student Self-Service: Registration > Online Forms.

EN.601.611. Computer Science Innovation & Entrepreneurship II. 3 Credits.
This course is the second half of a two-course sequence and is a continuation of course EN.660.410.01, CS Innovation and Entrepreneurship, offered by the Center for Leadership Education (CLE). In this sequel course the student groups, directed by CS faculty, will implement the business idea which was developed in the first course and will present the implementations and business plans to an outside panel made up of practitioners, industry representatives, and venture capitalists. [General]
Prerequisite(s): EN.660.410
Area: Engineering

EN.601.614. Computer Networks. 3 Credits.
Topics covered will include applications layer protocols (e.g. HTTP, FTP, SMTP), transport layer protocols (UDP, TCP), network layer protocols (e.g. IP, ICMP), link layer protocols (e.g. Ethernet) and wireless protocols (e.g. IEEE 802.11). The course will also cover routing protocols such as link state and distance vector, multicast routing, and path vector protocols (e.g. BGP). The class will examine security issues such as firewalls and denial of service attacks. We will also study DNS, Web caching and CDNS, peer to peer, and protocol tunneling. Finally, we will explore security protocols (e.g. TLS, SSH, IPsec), as well as some basic cryptography necessary to understand these. Grading will be based on hands-pn programming assignments, homework and two exams. [Systems]Required course background: C/C++ programming and data structures, or permission.
Prerequisite(s): Students can only receive credit for EN.601.414 or EN.601.614, but not both.
Area: Engineering

EN.601.615. Databases. 3 Credits.
Same material as 601.415, for graduate students. Introduction to database management systems and database design, focusing on the relational and object-oriented data models, query languages and query optimization, transaction processing, parallel and distributed databases, recovery and security issues, commercial systems and case studies, heterogeneous and multimedia databases, and data mining. [Systems] (www.cs.jhu.edu/~yarowsky/cs415.html) Required course background: Data Structures
Prerequisite(s): Students may receive credit for only one of EN.600.315, EN.600.415, EN.601.315, EN.601.415, EN.601.615.
Area: Engineering

EN.601.617. Distributed Systems. 3 Credits.
Graduate version of 601.317 Systems. Students may receive credit for 601.317 or 601.417 but not both. Recommended Course Background: EN.601.220, EN.601.226
Prerequisite(s): Students may receive credit for only one of 417/617
Area: Engineering

EN.601.618. Operating Systems. 3 Credits.
Same material as 601.418, for graduate students. This course covers fundamental topics related to operating systems theory and practice. Topics include processor management, storage management, concurrency control, multi-programming and processing, device drivers, operating system components (e.g., file system, kernel), modeling and performance measurement, protection and security, and recent innovations in operating system structure. Course work includes the implementation of operating systems techniques and routines, and critical parts of a small but functional operating system. [Systems] Required course background: Data Structures & Computer System Fundamentals
Prerequisite(s): Students may receive credit for only one of EN.600.318, EN.600.418, EN.601.318, EN.601.418, EN.601.618.
Area: Engineering

EN.601.619. Cloud Computing. 3 Credits.
Clouds host a wide range of the applications that we rely on today. In this course, we study common cloud applications, traffic patterns that they generate, critical networking infrastructures that support them, and core networking and distributed systems concepts, algorithms, and technologies used inside clouds. We will also study how today's application demand is influencing the network's design, explore current practice, and how we can build future's networked infrastructure to better enable both efficient transfer of big data and low-latency requirements of real-time applications. The format of this course will be a mix of lectures, discussions, assignments, and a project designed to help students practice and apply the theories and techniques covered in the course. [Systems] Prerequisites: EN.601.226 or permission. Students can only receive credit for one of 601.419/619. Recommended: a course in operating systems, networks or systems programming.
Prerequisite(s): Students may earn credit for EN.601.419 or EN.601.619, but not both.
Area: Engineering

EN.601.620. Parallel Computing for Data Science. 3 Credits.
This course studies parallelism in data science, drawing examples from data analytics, statistical programming, and machine learning. It focuses mostly on the Python programming ecosystem but will use C/C++ to accelerate Python and Java to explore shared-memory threading. It explores parallelism at all levels, including instruction level parallelism (pipelining and vectorization), shared-memory multicore, and distributed computing. Concepts from computer architecture and operating systems will be developed in support of parallelism, including Moore's law, the memory hierarchy, caching, processes/threads, and concurrency control. The course will cover modern data-parallel programming frameworks, including Dask, Spark, Hadoop!, and Ray. The course will not cover GPU deep-learning frameworks nor CUDA. The course is suitable for second-year undergraduate CS majors and graduate students from other science and engineering disciplines that have prior programming experience. Required course background: Data Structures, Computer System Fundamentals, and familiarity with Python. [Systems]
Prerequisite(s): Students may receive credit for only one of EN.601.320, EN.601.420, OR EN.601.620.
Area: Engineering

EN.601.621. Object Oriented Software Engineering. 3 Credits.
Same material as EN.601.421, for graduate students.This course covers object-oriented software construction methodologies and their application. The main component of the course is a large team project on a topic of your choosing. Course topics covered include object-oriented analysis and design, UML, design patterns, refactoring, program testing, code repositories, team programming, and code reviews. [Systems or Applications] Required course background: intermediate programming, data structures, and experience in mobile or web app development. Students may receive credit for only one of 601.421/621.
Prerequisite(s): Students may receive credit for only one of EN.600.321, EN.600.421, EN.601.421, EN.601.621.
Area: Engineering

EN.601.622. Software Testing & Debugging. 3 Credits.
Studies show that testing can account for over 50% of software development costs. This course presents a comprehensive study of software testing, principles, methodologies, tools, and techniques. Topics include testing principles, coverage (graph coverage, logic coverage, input space partitioning, and syntax-based coverage), unit testing, higher-order testing (integration, system-level, acceptance), testing approaches (white-box, black-box, grey-box), regression testing, debugging, delta debugging, and several specific types of functional and non-functional testing as schedule/interest permits (GUI testing, usability testing, security testing, load/performance testing, A/B testing etc.). For practical topics, state- of-the-art tools/techniques will be studied and utilized. [Systems]
Prerequisite(s): EN.601.290 OR EN.601.421 OR EN.601.621;Students can only take EN.601.422 or EN.601.622, but not both.
Area: Engineering

EN.601.624. Reliable Software Systems. 3 Credits.
Reliability is an essential quality requirement for all artifacts operating in the real-world, ranging from bridges, cars to power grids. Software systems are no exception. In this computing age when software is transforming even traditional mission-critical artifacts, making sure the software we write is reliable becomes ever more important. This course exposes students to the principles and techniques in building reliable systems. We will study a set of systematic approaches to make software more robust. These include but are not limited to static analysis, testing framework, model checking, symbolic execution, fuzzing, and formal verification. In addition, we will cover the latest research in system reliability. Recommended course background: EN.601.220 AND EN.601.628.
Prerequisite(s): Students may receive credit for EN.601.424 OR EN.601.624, but not both.

EN.601.626. Principles of Programming Languages. 3 Credits.
Same material as EN.601.426, for graduate students. Functional, object-oriented, and other language features are studied independent of a particular programming language. Students become familiar with these features by implementing them. Most of the implementations are in the form of small language interpreters. Some type checkers and a small compiler will also be written. The total amount of code written will not be overly large, as the emphasis is on concepts. The ML programming language is the implementation language used. [Analysis]Required course background: EN.601.226.
Prerequisite(s): Students may only receive credit for EN.601.426 or EN.601.626, but not both
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.627. Principles of Programming Languages II. 3 Credits.
This course is designed as a follow-on to Principles of Programming Languages. It will cover a wide array of fundamental topics in programming languages, including advanced functional programming, the theory of inductive definitions, advanced operational semantics, advanced type systems, program analysis, program verification, theorem provers and SAT solvers.[Analysis]
Prerequisite(s): EN.601.426 OR EN.601.626

EN.601.628. Compilers & Interpreters. 3 Credits.
Introduction to compiler design, including lexical analysis, parsing, syntax-directed translation, symbol tables, runtime environments, and code generation and optimization. Students are required to write a compiler as a course project. [Systems] Required course background: intermediate programming, data structures and computer system fundamentalsRecommended background: automata and computation theory
Prerequisite(s): Students may receive credit for only one of EN.601.428 or 601.628.
Area: Engineering

EN.601.629. Functional Programming in Software Engineering. 3 Credits.
How can we effectively use functional programming techniques to build real-world software? This course will primarily focus on using the OCaml programming language for this purpose. Topics covered include OCaml basics, modules, standard libraries, testing, quickcheck, build tools, functional data structures and efficiency analysis, monads, streams, and promises. Students will practice what they learn in lecture via functional programming assignments and a final project. Required course background in data structures (601.226)
Area: Engineering

EN.601.630. Combinatorics & Graph Theory in Computer Science. 3 Credits.
This is a graduate level course studying the applications of combinatorics and graph theory in computer science. We will start with some basic combinatorial techniques such as counting and pigeon hole principle, and then move to advanced techniques such as the probabilistic method, spectral graph theory and additive combinatorics. We shall see their applications in various areas in computer science, such as proving lower bounds in computational models, randomized algorithms, coding theory and pseudorandomness. [Analysis]Recommended Course Background: probability theory and linear algebra
Prerequisite(s): Students may receive credit for only one of 430/630
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.631. Theory of Computation. 3 Credits.
This is a graduate-level course studying the theoretical foundations of computer science. Topics covered will be models of computation from automata to Turing machines, computability, complexity theory, randomized algorithms, inapproximability, interactive proof systems and probabilistically checkable proofs. Students may not take both EN.601.231 and EN.601.631, unless one is for an undergrad degree and the other for grad. [Analysis]Recommended Course Background: EN.553.171 or istructor permission.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.633. Intro Algorithms. 3 Credits.
Same material as EN.601.433, for graduate students.This course concentrates on the design of algorithms and the rigorous analysis of their efficiency. topics include the basic definitions of algorithmic complexity (worst case, average case); basic tools such as dynamic programming, sorting, searching, and selection; advanced data structures and their applications (such as union-find); graph algorithms and searching techniques such as minimum spanning trees, depth-first search, shortest paths, design of online algorithms and competitive analysis. [Analysis] Required background: Data Structures and Discrete Math or Automata/Computation Theory
Prerequisite(s): Students may receive credit for only one of EN.600.363, EN.600.463, EN.601.433, EN.601.633.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.634. Randomized and Big Data Algorithms. 3 Credits.
Same material as 601.434, for graduate students. The course emphasizes algorithmic design aspects, and how randomization can be a helpful tool. The topics covered include: tail inequalities, linear programming relaxation & randomized rounding, de-randomization, existence proofs, universal hashing, markov chains, metropolis and metropolis-hastings methods, mixing by coupling and by eigenvalues, counting problems, semi-definite programming and rounding, lower bound arguments, and applications of expanders. [Analysis] (www.cs.jhu.edu/~cs464) Required course background: EN.600.363 or EN.601.433 or EN.601.633.
Prerequisite(s): Students may receive credit for only one of EN.600.464, EN.600.664, EN.601.434, EN.601.634.
Area: Engineering

EN.601.635. Approximation Algorithms. 3 Credits.
Graduate version of EN.601.435. Recommended Background: EN.601.633 or equivalent. Students may receive credit for EN.601.435 or EN.601.635, but not both.

EN.601.636. Algorithmic Game Theory. 3 Credits.
Same material as EN.601.436, for graduate students. This course provides an introduction to algorithmic game theory: the study of games from the perspective of algorithms and theoretical computer science. There will be a particular focus on games that arise naturally from economic interactions involving computer systems (such as economic interactions between large-scale networks, online advertising markets, etc.), but there will also be broad coverage of games and mechanisms of all sorts. Topics covered will include a) complexity of computing equilibria and algorithms for doing so, b) (in)efficiency of equilibria, and c) algorithmic mechanism design. [Analysis]
Prerequisite(s): Students may receive credit for EN.601.436 or EN.601.636, but not both.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.637. Federated Learning and Analytics. 3 Credits.
Federated Learning (FL) is an area of machine learning where data is distributed across multiple devices and training is performed without exchanging the data between devices. FL can be contrasted with classical machine learning settings when data is available in a central location. As such, FL faces additional challenges and limitations such as privacy and communication. For example, FL may deal with questions of learning from sensitive data on mobile devices while protecting privacy of individual users and dealing with low power and limited communication. As a result, FL requires knowledge of many interdisciplinary areas such as differential privacy, distributed optimization, sketching algorithms, compression and more. In this course students will learn basic concepts and algorithms for FL and federated analytics, and gain hands-on experience with new methods and techniques. Students will gain understanding in reasoning about possible trade-offs between privacy, accuracy and communication. [Analysis] Required:: 433/633 (Algo), 475/675 (ML) or 482/682 (ML: DL), linear algebra, probability
Prerequisite(s): Students may receive credit for only one of 601.437 or 601.637

EN.601.640. Web Security. 3 Credits.
This course begins with reviewing basic knowledge of the World Wide Web, and then exploring the central defense concepts behind Web security, such as same-origin policy, cross-origin resource sharing, and browser sandboxing. It will cover the most popular Web vulnerabilities, such as cross-site scripting (XSS) and SQL injection, as well as how to attack and penetrate software with such vulnerabilities. Students will learn how to detect, respond, and recover from security incidents. Newly proposed research techniques will also be discussed. [Systems]Required background: data structures and computer system fundamentals.
Prerequisite(s): Students may receive credit for only one of 601.640/440/640

EN.601.641. Blockchains and Cryptocurrencies. 3 Credits.
Same as EN.601.441, for graduate students. This course will introduce students to cryptocurrencies and the main underlying technology of Blockchains. The course will start with the relevant background in cryptography and then proceed to cover the recent advances in the design and applications of blockchains. This course should primarily appeal to students who want to conduct research in this area or wish to build new applications on top of blockchains. It should also appeal to those who have a casual interest in this topic or are generally interested in cryptography. Students are expected to have mathematical maturity. Recommended Course Background: EN.601.226 AND (EN.553.310 OR EN.553.420) [Analysis]
Prerequisite(s): Students may receive credit for only one of EN.600.451 OR EN.601.441 OR EN.601.641
Area: Engineering

EN.601.642. Modern Cryptography. 3 Credits.
Same material as 601.442, for graduate students. Modern Cryptography includes seemingly paradoxical notions such as communicating privately without a shared secret, proving things without leaking knowledge, and computing on encrypted data. In this challenging but rewarding course we will start from the basics of private and public key cryptography and go all the way up to advanced notions such as zero-knowledge proofs, functional encryption and program obfuscation. The class will focus on rigorous proofs and require mathematical maturity. [Analysis] Required course background: Probability & Automata/Computation Theory
Prerequisite(s): Students may receive credit for only one of EN.601.442 OR EN.601.642.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.643. Security & Privacy in Computing. 3 Credits.
Same material as 601.443, for graduate students. Lecture topics will include computer security, network security, basic cryptography, system design methodology, and privacy. There will be a heavy work load, including written homework, programming assignments, exams and a comprehensive final. The class will also include a semester-long project that will be done in teams and will include a presentation by each group to the class. [Applications] Required Course Background: A basic course in operating systems and networking, or permission of instructor.
Prerequisite(s): Students may receive credit for only one of EN.600.443, EN.601.443, EN.601.643.
Area: Engineering

EN.601.644. Network Security. 3 Credits.
Same material as 601.444, for graduate students. This course focuses on communication security in computer systems and networks. The course is intended to provide students with an introduction to the field of network security. The course covers network security services such as authentication and access control, integrity and confidentiality of data, firewalls and related technologies, Web security and privacy. Course work involves implementing various security techniques. A course project is required. [Systems]Recommended. Course Background: EN.601.220, EN.601.226 or equivalent
Prerequisite(s): Students may receive credit for only one of EN.600.454, EN.650.454, EN.601.445, EN.601.645.
Area: Engineering

EN.601.645. Practical Cryptographic Systems. 3 Credits.
Same material as 601.445, for graduate students. This semester-long course will teach systems and cryptographic design principles by example: by studying and identifying flaws in widely-deployed cryptographic products and protocols. Our focus will be on the techniques used in practical security systems, the mistakes that lead to failure, and the approaches that might have avoided the problem.We will place a particular emphasis on the techniques of provable security and the feasibility of reverse-engineering undocumented cryptographic systems. [Systems]
Prerequisite(s): Students may receive credit for EN.600.454/EN.601.445 or EN.601.645, but not both.
Area: Engineering

EN.601.646. Sketching and Indexing for Sequences. 3 Credits.
Many of the world's largest and fastest-growing datasets are text, e.g. DNA sequencing data, web pages, logs and social media posts. Such datasets are useful only to the degree we can query, compare and analyze them. Here we discuss two powerful approaches in this area. We will cover sketching, which enables us to summarize very large texts in small structures that allow us to measure the sizes of sets and of their unions and intersections. This in turn allows us to measure similarity and find near neighbors. Second, we will discuss indexing --- succinct and compressed indexes in particular -- which enables us to efficiently search inside very long strings, especially in highly repetitive texts. [Analysis]
Prerequisite(s): Students may receive credit for EN.601.446 or EN.601.646, but not both.

EN.601.647. Computational Genomics: Sequences. 3 Credits.
Same material as 601.447, for graduate students. Your genome is the blueprint for the molecules in your body. It's also a string of letters (A, C, G and T) about 3 billion letters long. How does this string give rise to you? Your heart, your brain, your health? This, broadly speaking, is what genomics research is about. This course will familiarize you with a breadth of topics from the field of computational genomics. The emphasis is on current research problems, real-world genomics data, and efficient software implementations for analyzing data. Topics will include: string matching, sequence alignment and indexing, assembly, and sequence models. Course will involve significant programming projects. [Applications]Required course background: Intermediate programming (C/C++) and Data Structures
Prerequisite(s): Students may receive credit for only one EN.601.447/647/747
Area: Engineering

EN.601.648. Computational Genomics: Data Analysis. 3 Credits.
Same material as EN.601.448, for graduate students. Genomic data has the potential to reveal causes of disease, novel drug targets, and relationships among genes and pathways in our cells. However, identifying meaningful patterns from high-dimensional genomic data has required development of new computational tools. This course will cover current approaches in computational analysis of genomic data with a focus on statistical methods and machine learning. Topics will include disease association, prediction tasks, clustering and dimensionality reduction, data integration, and network reconstruction. There will be some programming and a project component. [Applications]Recommended Course Background: EN.600.226 or other programming experience, probability and statistics, linear algebra or calculus.
Area: Engineering

EN.601.649. Computational Genomics: Applied Comparative Genomics. 3 Credits.
The goal of this course is to study the leading computational and quantitative approaches for comparing and analyzing genomes starting from raw sequencing data. The course will focus on human genomics and human medical applications, but the techniques will be broadly applicable across the tree of life. The topics will include genome assembly & comparative genomics, variant identification & analysis, gene expression & regulation, personal genome analysis, and cancer genomics. The grading will be based on assignments, a midterm exam, class presentations, and a significant class project. Prerequisites: knowledge of the Unix operating system and programming expertise in a language such as R or Python. [Applications]

EN.601.651. Introduction to Computational Immunogenomics. 3 Credits.
Immunology studies defensive mechanisms of living organisms against external threats. Computational immunogenomics is a new field of bioinformatics that develops and applies computational approaches to the study and interpretation of immunological data, seeking to answer questions about adaptive immune responses in humans and important animals. In this course, students will learn how to design, apply, and benchmark algorithms for solving immunogenomics problems. [Applications] Students may receive credit for only one of EN.601.451, EN.601.651.
Prerequisite(s): Students may receive credit for only one of EN.601.451 OR EN.601.651.

EN.601.654. Augmented Reality. 3 Credits.
This course introduces students to the field of Augmented Reality. It reviews its basic definitions, principles and applications. It then focuses on Medical Augmented Reality and its particular requirements. The course also discusses the main issues of calibration, tracking, multi-modal registration, advance visualization and display technologies. Homework in this course will relate to the mathematical methods used for calibration, tracking and visualization in medical augmented reality. Students may also be asked to read papers and implement various techniques within group projects. Recommended Course Background: EN.601.220, EN.601.226, and AS.110.201. [Applications]
Prerequisite(s): Students may receive credit for only one of EN.600.484, EN.600.684, EN.601.454, EN.601.654.

EN.601.655. Computer Integrated Surgery I. 4 Credits.
Same material as 601.455, for graduate students. This course focuses on computer-based techniques, systems, and applications exploiting quantitative information from medical images and sensors to assist clinicians in all phases of treatment from diagnosis to preoperative planning, execution, and follow-up. It emphasizes the relationship between problem definition, computer-based technology, and clinical application and includes a number of guest lectures given by surgeons and other experts on requirements and opportunities in particular clinical areas. [Applications] Required Course Background: data structures and linear algebra or permission.Recommended Course Background: intermediate programming in C/C++, EN.601.457, EN.601.461, image processing.
Prerequisite(s): Students may receive credit for only one of EN.601.455 or EN.601.655.
Area: Engineering

EN.601.656. Computer Integrated Surgery II. 3 Credits.
Same material as EN.601.456, for graduate students. This weekly lecture/seminar course addresses similar material to EN.601.655, but covers selected topics in greater depth. In addition to material covered in lectures/seminars by the instructor and other faculty, students are expected to read and provide critical analysis/presentations of selected papers in recitation sessions. Students taking this course are required to undertake and report on a significant term project under the supervision of the instructor and clinical end users. Typically, this project is an extension of the term project from EN.601.655, although it does not have to be. Grades are based both on the project and on classroom recitations. Students wishing to attend the weekly lectures as a 1-credit seminar should sign up for EN.601.356. [Applications]
Prerequisite(s): EN.600.445/EN.601.455 OR EN.600.645/EN.601.655 OR permission of the instructor.;Students may receive credit for only one of EN.600.446, EN.600.646, EN.601.456, EN.601.656.

EN.601.657. Computer Graphics. 3 Credits.
Same material as 601.457, for graduate students. This course introduces computer graphics techniques and applications, including image processing, rendering, modeling and animation. [Applications]Permission of instructor is required for students not satisfying a pre-requisite. No Audits.Required course background: EN.601.220 (C++), EN.601.226, linear algebra.
Prerequisite(s): Students may receive credit for only one of EN.601.457 OR EN.601.657.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.659. Computational Geometry. 3 Credits.
This course will provide an introduction to computational geometry. It will cover a number of topics in two- and three-dimensions, including polygon triangulations and partitions, convex hulls, Delaunay and Voronoi diagrams, arrangements, and spatial queries. Time-permitting, we will also look at kD-trees, general BSP-trees, and quadtrees. [Analysis] Recommended Course Background: EN.601.220 AND EN.601.226 AND (EN.600.363 OR EN.601.433).
Prerequisite(s): Students may earn credit for EN.601.459 or EN.601.659, but not both.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.661. Computer Vision. 3 Credits.
This course provides an overview of fundamental methods in computer vision from a computational perspective. Methods studied include: camera systems and their modelling, computation of 3¬D geometry from binocular stereo, motion, and photometric stereo, and object recognition, image segmentation, and activity analysis. Elements of machine learning and deep learning are also included. [Applications]Required course background: Intro to Programming, Linear Algebra & prob/stats
Prerequisite(s): Students may receive credit for only one of EN.601.461, EN.601.661, OR EN.601.761.
Area: Engineering

EN.601.662. Introduction to Spatial Computing. 3 Credits.
This course will provide students with a rich understanding of immersive technology and spatial computing, including virtual, augmented, and mixed reality as the next wave of computing after personal and mobile computing, and belongs to the systems that can sense the space or are “spatially” aware. It also covers input systems and interaction modalities that have evolved to support human-computer interaction required for immersive environments. It will go through principles of design thinking including verification and validation, as a mindset for creating immersive experiences, and students will explore the practical implication of this subject in healthcare, industry, and society through the projects.
Area: Engineering

EN.601.663. Algorithms for Sensor-Based Robotics. 3 Credits.
Same material as EN.601.463, for graduate students. This course surveys the development of robotic systems for navigating in an environment from an algorithmic perspective. It will cover basic kinematics, configuration space concepts, motion planning, and localization and mapping. It will describe these concepts in the context of the ROS software system, and will present examples relevant to mobile platforms, manipulation, robotics surgery, and human-machine systems. [Analysis] Required course background: Data Structures, Linear Algebra & prob/ stats
Prerequisite(s): Students may receive credit for only one of 601.463/663/763
Area: Engineering

EN.601.664. Artificial Intelligence. 3 Credits.
Same material as EN.601.464, for graduate students. This course is recommended for students, scientists, and engineers with a genuine curiosity about the fundamental obstacles in getting machines to perform tasks such as deduction, learning, planning and navigation. It covers methods for automated reasoning, automatic problem solvers and planners, knowledge representation mechanisms, game playing, machine learning, and statistical pattern recognition. Strong programming skills are expected, as well as basic familiarity with probability. Students intending to also take courses in machine learning (e.g. 601.475/675, 601.476/676, 601.482/682) may find it beneficial to take this course first, or concurrently.Prereq: Data Structures ; Recommended: linear algebra, prob/stat.
Prerequisite(s): Students may receive credit for only one of EN.601.464 OR EN.601.664.
Area: Engineering

EN.601.665. Natural Language Processing. 3 Credits.
Same material as 601.465, for graduate students. This course is an in-depth overview of techniques for processing human language. How should linguistic structure and meaning be represented? What algorithms can recover them from text? And crucially, how can we build statistical models to choose among the many legal answers? The course covers methods for trees (parsing and semantic interpretation), sequences (finite-state transduction such as morphology), and words (sense and phrase induction), with applications to practical engineering tasks such as information retrieval and extraction, text classification, part-of-speech tagging, speech recognition and machine translation. There are a number of structured but challenging programming assignments. [Applications]Prerequisite: Data Structures and basic familiarity with Python, partial derivatives, matrix multiplication and probabilities.
Prerequisite(s): Students may receive credit for only one of EN.601.465 OR EN.601.665.
Area: Engineering

EN.601.666. Information Retrieval and Web Agents. 3 Credits.
Same material as EN.601.466, for graduate students. An in-depth, hands-on study of current information retrieval techniques and their application to developing intelligent WWW agents. Topics include a comprehensive study of current document retrieval models, mail/news routing and filtering, document clustering, automatic indexing, query expansion, relevance feedback, user modeling, information visualization and usage pattern analysis. In addition, the course explores the range of additional language processing steps useful for template filling and information extraction from retrieved documents, focusing on recent, primarily statistical methods. The course concludes with a study of current issues in information retrieval and data mining on the World Wide Web. Topics include web robots, spiders, agents and search engines, exploring both their practical implementation and the economic and legal issues surrounding their use. [Applications]
Area: Engineering

EN.601.667. Introduction to Human Language Technology. 3 Credits.
This course gives an overview of basic foundations and applications of human language technology, such as: morphological, syntactic, semantic, and pragmatic processing; machine learning; signal processing; speech recognition; speech synthesis; information retrieval; text classification; topic modelling; informationextraction; knowledge representation; machine translation; dialog systems; etc. [Applications]Required Background: EN.601.226 Data Structures; knowledge of Python recommended.
Prerequisite(s): Students may receive credit for only one of 601.467/667

EN.601.668. Machine Translation. 3 Credits.
Same material as 601.468, for graduate students. Google translate can instantly translate between any pair of over fifty human languages (for instance, from French to English). How does it do that? Why does it make the errors that it does? And how can you build something better? Modern translation systems learn to translate by reading millions of words of already translated text, and this course will show you how they work. The course covers a diverse set of fundamental building blocks from linguistics, machine learning, algorithms, data structures, and formal language theory, along with their application to a real and difficult problem in artificial intelligence. [Applications] Required course background: Data Structures and prob/stats
Prerequisite(s): Students may receive credit for only one of EN.601.468 OR EN.601.668.
Area: Engineering

EN.601.670. Artificial Agents. 3 Credits.
This course covers a number of topics explored in introductory AI, such as knowledge representation, reasoning, and natural language understanding. Unlike introductory AI, we will pursue these topics based on the transformer neural architecture. We will motivate the material through interacting with agents in games: how to build models that understand user commands, how to generate responses back to a user, and how to reason about a synthetic environment to determine a course of action. Assignments will include programming, presentations on readings, and written summaries of readings. [Applications]

EN.601.674. ML: Learning Theory. 3 Credits.
This is a graduate level course in machine learning. It will provide a formal and in-depth coverage of topics in statistical and computational learning theory. We will revisit popular machine learning algorithms and understand their performance in terms of the size of the data (sample complexity), memory needed (space complexity), as well as the overall runtime (computational or iteration complexity). We will cover topics including PAC learning, uniform convergence, VC dimension, Rademacher complexity, algorithmic stability, kernel methods, online learning and reinforcement learning, as well as introduce students to current topics in large-scale machine- learning and randomized projections. General focus will be on combining methodology with theoretical and computational foundations. [Analysis]

EN.601.675. Machine Learning. 3 Credits.
Same material as 601.475, for graduate students. Machine learning is subfield of computer science and artificial intelligence, whose goal is to develop computational systems, methods, and algorithms that can learn from data to improve their performance. This course introduces the foundational concepts of modern Machine Learning, including core principles, popular algorithms and modeling platforms. This will include both supervised learning, which includes popular algorithms like SVMs, logistic regression, boosting and deep learning, as well as unsupervised learning frameworks, which include Expectation Maximization and graphical models. Homework assignments include a heavy programming components, requiring students to implement several machine learning algorithms in a common learning framework. Additionally, analytical homework questions will explore various machine learning concepts, building on the pre-requisites that include probability, linear algebra, multi-variate calculus and basic optimization. Students in the course will develop a learning system for a final project. [Applications or Analysis] Required course background: multivariable calculus, probability, linear algebra, intro to computing
Prerequisite(s): Students may receive credit for only one of EN.601.475 OR EN.601.675.
Area: Engineering

EN.601.676. Machine Learning: Data to Models. 3 Credits.
Same material as EN.601.476, for graduate students. How can robots localize themselves in an environment when navigating? Which factors predict whether patients are at greatest-risk for complications in the hospital? Can we reconstruct the brain's "connectome" from fMRI data? Many such big data questions can be answered using the paradigm of probabilistic models in machine learning. This is the second course on machine learning which focuses on probabilistic graphical models. You will learn about directed and undirected graphical models, inference methods, sampling, structure learning algorithms, latent variables, and temporal models. There will be regular assignments, which include theory and some programming. Students will analyze real data for their final project, applying methods discussed in class and writing up a report of their results. [Analysis or Applications] Recommended Background: EN.600.475 or EN.601.675 or equivalent. Students may receive credit for EN.600.476 or EN.600.676, but not both.
Prerequisite(s): Students may receive credit for only one of EN.600.476, EN.601.476, EN.601.676.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.677. Causal Inference. 3 Credits.
"Big data" is not necessarily "high quality data." Systematically missing records, unobserved confounders, and selection effects present in many datasets make it harder than ever to answer scientifically meaningful questions. This course will teach mathematical tools to help you reason about causes, effects, and bias sources in data with confidence. We will use graphical causal models, and potential outcomes to formalize what causal effects mean, describe how to express these effects as functions of observed data, and use regression model techniques to estimate them. We will consider techniques for handling missing values, structure learning algorithms for inferring causal directionality from data, and connections between causal inference and reinforcement learning. [Analysis] Pre-requisites: familiarity with the R programming language, multivariate calculus, basics of linear algebra and probability.
Prerequisite(s): Students may receive credit for only one of EN.601.477 OR EN.601.677.
Area: Engineering, Quantitative and Mathematical Sciences

EN.601.681. Machine Learning: Optimization. 3 Credits.
Same material as EN.601.481, for graduate students. Optimization is at the heart of machine learning. Most machine learning problems can be posed as optimization problems. However, unlike mathematical optimization where the focus is on efficient algorithms for finding solutions with a high degree of accuracy as measured by optimality conditions, optimization for machine learning focuses on algorithms that are efficient and generalize well. In this course, we will focus on optimization for problems that arise in machine learning, design and analysis of algorithms for solving these problems, and the interplay of optimization and machine learning. The coursework will include homework assignments and a final project focusing on applying optimization algorithms to real world machine learning problems. [Analysis or Applications]Required Course Background: EN 601.475/675 Machine Learning or all of the following:1. Linear algebra (vector spaces, normed vectors, inner product spaces, singular value decomposition2. Probability and Statistics (random variables, probability distributions, expectation, mean, variance, covariance, conditional probability, Bayes rule3. Introductory machine learning (classification, regression, empirical risk minimization, regularization)4. Multivariate calculus (partial derivative, gradient, Jacobian, Hessian, critical points
Prerequisite(s): Students may receive credit for only one of EN.601.481/681

EN.601.682. Machine Learning: Deep Learning. 4 Credits.
Deep learning (DL) has emerged as a powerful tool for solving data-intensive learning problems such as supervised learning for classification or regression, dimensionality reduction, and control. As such, it has a broad range of applications including speech and text understanding, computer vision, medical imaging, and perception-based robotics. The goal of this course is to introduce the basic concepts of deep learning (DL). The course will include a brief introduction to the basic theoretical and methodological underpinnings of machine learning, commonly used architectures for DL, DL optimization methods, DL programming systems, and specialized applications to computer vision, speech understanding, and robotics. Students will be expected to solve several DL problems on standardized data sets, and will be given the opportunity to pursue team projects on topics of their choice. [Applications] Recommended Course Background: (AS.110.201 or AS.110.212 or EN.553.291) and (EN.553.310 EN.553.311 or EN.553.420); numerical optimization recommended.
Prerequisite(s): Students may receive credit for EN.601.482 or EN.601.682, but not both.
Area: Engineering

EN.601.684. ML: Interpretable Machine Learning Design. 3 Credits.
There are considerable research thrusts that seek to increase the trustworthiness and perceived reliability of machine learning solutions. One such thrust, interpretable machine learning, attempts to reveal the working mechanisms of a machine learning system. However, other than on-task performance, interpretability is not a property of machine learning algorithms, but an affordance: a relationship between interpretable model and the target users in their context. Successful development of machine learning solutions that afford interpretation thus requires understanding of techniques beyond pure machine learning. In this course, we will first review the basics of machine learning and human-centered design. Then, during student team-delivered lectures, we will learn about contemporary techniques to introduce interpretability to machine learning models and discuss recent literature on the topic. In addition to hands-on homework assignments, students will work in groups to design, justify, implement, and test an interpretable machine learning algorithm for a problem of their choosing. Required course background: 601.475/675 or 601.464/664 or 601.482/682; coding in Python/PyTorch. Recommended (601.454/654, 601.290, 601.490/690 or 601.491/691) and 601.477/677.

EN.601.686. Machine Learning: Artificial Intelligence System Design & Development. 3 Credits.
The field of artificial intelligence (AI) has recently seen a substantial increase in popularity, largely fueled by the successes of training deep neural networks that achieve state-of-the-art performance in a large variety of problems. These successes are not limited to academic benchmarks but have started to impact our everyday lives in the form of products such as Google Lens, Amazon Alexa, and Tesla Autopilot. In order for such AI systems to succeed we must consider its impact on everyday life, its overall capabilities and performance, and the effectiveness of the human-AI interaction. The importance of harmonic interplay between all these components is dramatically highlighted by recent catastrophic events in road transport and aviation. In this project-based course you will work in teams of 3-5 students to 1) Identify a need with high-impact implications on everyday life; 2) Conceptualize and design an AI system targeting this need, and 3) Develop the AI system by refining a demo-able prototype based on feedback received during course presentations.Required course background: (EN.601.475/675 or EN.601.464/664 or EN.601.482/682) and Python programming. Recommended: 601.290 or 601.454/654 or 601.490/690 or 601.491/691 (experience with human computer interface design).
Prerequisite(s): (EN.601.475 OR EN.601.675) OR (EN.601.464 OR EN.601.664) OR (EN.601.482 OR EN.601.682)
Area: Engineering

EN.601.690. Introduction to Human-Computer Interaction. 3 Credits.
This course is designed to introduce undergraduate and graduate students to design techniques and practices in human-computer interaction (HCI), the study of interactions between humans and computing systems. Students will learn design techniques and evaluation methods, as well as current practices and exploratory approaches, in HCI through lectures, readings, and assignments. Students will practice various design techniques and evaluation methods through hands-on projects focusing on different computing technologies and application domains. This course is intended for undergraduate and graduate students in Computer Science/Cognitive Science/Psychology. Interested students from different disciplines should contact the instructor before enrolling in this course. [Applications]Recommended Background: Basic programming skills.
Prerequisite(s): Students can receive credit for either EN.601.490 or EN.601.690, but not both.
Area: Engineering

EN.601.691. Human-Robot Interaction. 3 Credits.
This course is designed to introduce graduate students to research methods and topics in human-robot interaction (HRI), an emerging research area focusing on the design and evaluation of interactions between humans and robotic technologies. Students will (1) learn design principles for building and research methods of evaluating interactive robot systems through lectures, readings, and assignments, (2) read and discuss relevant literature to gain sufficient knowledgeof various research topics in HRI, and (3) work on a substantial project that integrates the principles, methods, and knowledge learned in this course. [Applications] Required course background: EN.601.220 and EN.601.226.
Prerequisite(s): Students may receive credit for EN.601.491 or EN.601.691.

EN.601.713. Future Networks. 3 Credits.
This will be a graduate-level networking course. New applications such as ones for metaverse require networking and computing to be imbedded together. This feature is already beginning to be implemented in 5G and 6G networks; 6G will also allow networks to be used as sensors. These advances are enabled by new technologies such as mobile edge computing, software-defined networking (SDN), network slicing, digital twins, and named-data networking (NDN). This course will start with introductory lectures on these topics. Students will be asked to study new papers and do course projects. These activities should result in longer term research projects. Required Course Background: A course in computer networks (e.g., EN.601.414/614 Computer Network Fundamentals or the equivalent), or permission of the instructor. [Systems]

EN.601.714. Advanced Computer Networks. 3 Credits.
This is a graduate-level course on computer networks. It provides a comprehensive overview on advanced topics in network protocols and networked systems. The course will cover both classic papers on Internet protocols and recent research results. It will examine a wide range of topics, e.g., routing, congestion control, network architectures, datacenter networks, network virtualization, software-defined networking, and programmable networks, with an emphasize on core networking concepts and principles. The course will include lectures, paper discussions, programming assignments and a research project. Recommended Course Background: One undergraduate course in computer networks (e.g., EN.601.414/614 Computer Network Fundamentals or the equivalent), or permission of the instructor. The course assignments and projects assume students to be comfortable with programming.

EN.601.717. Advanced Distributed Systems & Networks. 3 Credits.
The course explores the state of the art in distributed systems, networks and Internet research and practice, trying to see what it would take to push the envelop a step further. The course is conducted as a discussion group, where the professor and students brainstorm and pick interesting semester-long projects with high potential future impact. Example areas include robust scalable infrastructure (distributed datacenters, cloud networking, scada systems), real-time performance (remote surgery, trading systems), hybrid networks (mesh networks, 3-4G/Wifi/Bluetooth). Students should feel free to bring their own topics of interest and ideas. Recommended Course Background: a systems course (distributed systems, operating systems, computer networks, parallel programming) or permission of instructor.

EN.601.718. Advanced Operating Systems. 3 Credits.
Students will study advanced operating system topics and be exposed to recent developments in operating systems research. This course involves readings on classic and new papers. Topics include virtual memory management, synchronization and communication, file systems, protection and security, operating system structure and extension techniques, fault tolerance, and history and experience of systems programming. [Systems]
Prerequisite(s): EN.600.318 OR EN.600.418 OR EN.601.318 OR EN.601.418 OR EN.601.618

EN.601.723. Advanced Topics in Data-Intensive Computing. 3 Credits.
The advent of cloud computing has lead to an explosion of storage system and data analysis software, including NoSQL databases, bulk-synchronous processing, graph computing engines, and stream processing. This course will explore scale-out software architectures for data-processing tasks. It will examine the algorithms and data-structures that underlie scalable systems and look at how hardware and networking trends influence the design and deployment of cloud computing. Recommended Course Background: EN.601.320/420 or permission of instructor. [Systems]
Prerequisite(s): EN.600.320 OR EN.600.420 OR EN.601.620

EN.601.740. Language-based Security. 3 Credits.
This course will introduce Language-based Security, an emerging field in cyber security that leverages techniques from compilers and program analysis for security-related problems. Topics include but are not limited to: Control-flow and data-flow graphs, Program slicing, Code property graph (CPG), and Control-flow integrity. Students are expected to read new and classic papers in this area and discuss them in class. Recommended backgrounds are Operating Systems and preferably Compilers.

EN.601.741. Advanced Topics in Secure and Censorship-Resistant Communications. 3 Credits.
Topics will vary from year to year, but will focus on applied cryptography and communications, focused on the development of secure and uncensorable communication mechanisms for communities at risk. This course will include topics such as: communication protocol design and analysis, blockchain-based protocols, anonymous communication, cryptographic backdoors, and other topics. Emphasis in this course is on understanding how cryptographic issues impact real systems, while maintaining an appreciation for grounding the work in fundamental science. The course will consist of in-class workshops and interactive discussions. There will be programming assignments and a course project with real world impact. Students will also be expected to read assigned papers and to present at least one research paper and lead a discussion on it. [Systems]
Prerequisite(s): ((EN.601.441 OR EN.601.641) OR (EN.601.442 OR EN.601.642) OR (EN.601.445 OR EN.601.645))

EN.601.742. Advanced Topics in Cryptography. 3 Credits.
This course will focus on advanced cryptographic topics with an emphasis on open research problems and student presentations.

EN.601.743. Advanced Topics in Computer Security. 3 Credits.
Topics will vary from year to year, but will focus mainly on network perimeter protection, host-level protection, authentication technologies, intellectual property protection, formal analysis techniques, intrusion detection and similarly advanced subjects. Emphasis in this course is on understanding how security issues impact real systems, while maintaining an appreciation for grounding the work in fundamental science. Students will study and present various advanced research papers to the class. There will be homework assignments and a course project. A college level security or crypto course at Hopkins or any other school is required.

EN.601.745. Advanced Topics in Applied Cryptography. 3 Credits.
This reading and project based course will explore the latest research in the area of applied cryptography and cryptographic engineering. Topics covered will include zero knowledge, efficient multiparty computation, cryptocurrencies, and trusted computing hardware. Readings will be drawn from the latest applied cryptography and security conferences. The course will include both reading, critical analysis, presentations and a course programming project. [Analysis or Applications]
Prerequisite(s): EN.600.454 OR EN.601.445 OR EN.601.645 OR EN.600.442 OR EN.601.442 OR EN.601.642

EN.601.749. Advanced Computational Genomics: Applied Comparative Genomics. 3 Credits.
The goal of this course is to study the leading computational and quantitative approaches for comparing and analyzing genomes starting from raw sequencing data. The course will focus on human genomics and human medical applications, but the techniques will be broadly applicable across the tree of life. The topics will include genome assembly & comparative genomics, variant identification & analysis, gene expression & regulation, personal genome analysis, and cancer genomics. The grading will be based on assignments, a midterm & final exam, class presentations, and a significant class project. [Applications] Expected course background: familiarity with UNIX scripting and/or programming.

EN.601.760. FFT in Graphics & Vision. 3 Credits.
In this course, we will study the Fourier Transform from the perspective of representation theory. We will begin by considering the standard transform defined by the commutative group of rotations in 2D and translations in two- and three-dimensions, and will proceed to the Fourier Transform of the non-commutative group of 3D rotations. Subjects covered will include correlation of images, shape matching, computation of invariances, and symmetry detection. Recommended Course Background: AS.110.201 and comfort with mathematical derivations.

EN.601.765. Machine Learning: Linguistic & Sequence Modeling. 3 Credits.
This course surveys formal ingredients that are used to build structured models of character and word sequences. We will unpack recent deep learning architectures that consider various kinds of latent structure, and see how they draw on earlier work in structured prediction, dimensionality reduction, Bayesian nonparametrics, multi-task learning, etc. We will also examine a range of strategies used for inference and learning in these models. Students will beexpected to read recent papers and carry out a research project. [Applications or Analysis]
Prerequisite(s): EN.600.465/EN.601.465 or EN.601.665

EN.601.769. Events Semantics in Theory and Practice. 3 Credits.
This course explores selected topics in the nature of event representations from the perspective of cognitive science, computer science, linguistics, and philosophy. These fields have developed a rich array of scientific theories about the representation of events, and how humans make inferences about them -- we investigate how (and if) such theories could be applied to current research topics and tasks in computational semantics such as inference from text, automated summarization, veridicality assessment, and so on. In addition to classic articles dealing with formal semantic theories, the course considers available machine-readable corpora, ontologies, and related resources that bear on event structure, such as WordNet, PropBank, FrameNet, etc.. The course is aimed to marry theory with practice: students with either a computational or linguistic background are encouraged to participate. [Applications]

EN.601.771. Self-Supervised Statistical Models: Opportunities, Challenges and Risks. 3 Credits.
The rise of massive self-supervised (pre-trained) models has transformed various data-driven fields such as natural language processing, computer vision, robotics, and medical imaging. This advanced graduate course aims to provide a holistic view of the issues related to these models: We will start with the history of how we got here, and then delve into the latest success stories. We will then focus on the implications of these technologies: social harms, security risks, legal issues, and environmental impacts. The class ends with reflections on the future implications of this trajectory.Required Course Background: knowledge equivalent to EN.601.675 ML, EN.601.664 AI, EN.601.465 NLP; also linear algebra and statistics.

EN.601.778. Advanced Topics in Causal Inference. 3 Credits.
This course will cover advanced topics on all areas of causal inference, including learning causal effects, path-specific effects, and optimal policies from data featuring biases induced by missing data, confounders, selection, and measurement error, techniques for generalizing findings to different populations, complex probabilistic models relevant for causal inference applications, learning causal structure from data, and inference under interference and network effects. The course will feature a final project which would involve either an applied data analysis problem (with a causal inference flavor), a literature review, or theoretical work. [Analysis] Recommended Background: EN.600.477/677 or EN.601.477/677

EN.601.779. Machine Learning: Advanced Topics. 3 Credits.
This course will focus on recent advances in machine learning. Topics will vary from year to year. The course will be project focused and involve presenting and discussing recent research papers.

EN.601.780. Unsupervised Learning: From Big Data to Low-Dimensional Representations. 3 Credits.
In the era of data deluge, the development of methods for discovering structure in high-dimensional data is becoming increasingly important. This course will cover state-of-the-art methods from algebraic geometry, sparse and low-rank representations, and statistical learning for modeling and clustering high-dimensional data. The first part of the course will cover methods for modeling data with a single low-dimensional subspace, such as PCA, Robust PCA, Kernel PCA, and manifold learning techniques. The second part of the course will cover methods for modeling data with multiple subspaces, such as algebraic, statistical, sparse and low-rank subspace clustering techniques. The third part of the course will cover applications of these methods in image processing, computer vision, and biomedical imaging. Requistes include Linear Algebra, Optimization, and prior exposure to Machine I.

EN.601.783. Vision as Bayesian Inference. 3 Credits.
This is an advanced course on computer vision from a probabilistic and machine learning perspective. It covers techniques such as linear and non-linear filtering, geometry, energy function methods, markov random fields, conditional random fields, graphical models, probabilistic grammars, and deep neural networks. These are illustrated on a set of vision problems ranging from image segmentation, semantic segmentation, depth estimation, object recognition, object parsing, scene parsing, action recognition, and text captioning. [Analysis orApplications] Required course background: calculus, linear algebra (AS.110.201 or equiv.), probability and statistics (AS.553.311 or equiv.), and the ability to program in Python and C++. Background in computer vision (EN.601.461/661) and machine learning (EN.601.475) suggested but not required.

EN.601.787. Advanced Machine Learning: Machine Learning for Trustworthy AI. 3 Credits.
This course teaches advanced machine learning methods for the design, implementation, and deployment of trustworthy AI systems. The topics we will cover include but are not limit to different types of robust learning methods, fair learning methods, safe learning methods, and research frontiers in transparency, interpretability, privacy, sustainability, AI safety and ethics. Students will learn the state-of-the-art methods in lectures, understand the recent advances by critiquing research articles, and apply/innovate new machine learning methods in an application. There will be homework assignments and a course project.
Area: Engineering

EN.601.801. Computer Science Seminar. 1 Credit.
Required for all full-time CS PhD students. Recommended for MSE students.

EN.601.803. Masters Research. 3 - 10 Credits.
Permission required. Independent research for masters or pre-dissertation PhD students.

EN.601.805. Graduate Independent Study. 1 - 3 Credits.
Permission required. Individual study in an area of mutual interest to a graduate student and a faculty member in the department.

EN.601.807. Teaching Practicum. 1 Credit.
PhD students will gain valuable teaching experience, working closely with their assigned faculty supervisor. Successful completion of this course fulfills the PhD teaching requirement.(grad students) Permission req'd.

EN.601.808. Selected Topics in CS Education. 1 Credit.
This course will explore current issues and research in computer science education. Topics will be drawn from literature, news items, and participant experience. Current faculty and students with interests in academic careers are encouraged to attend.

EN.601.809. PhD Research. 3 - 20 Credits.

EN.601.810. Diversity and Inclusion in Computer Science and Engineering. 1 Credit.
This reading seminar will focus on the question of diversity and inclusion in Computer Science (in particular) and engineering (in general). We aim to study the ways in which the curriculum, environment, and structure of computer science within academia perpetuates biases alienating female and minoritized students, and to explore possible approaches for diversifying our field. The seminar will meet on a weekly basis, readings will be assigned, and students will be expected to participate in the discussion.

EN.601.814. Selected Topics in Computer Networks. 1 Credit.
In this course, we will read, discuss and present classic papers and current research in computer networks. The topic coverage will vary each semester.
Area: Engineering

EN.601.817. Selected Topics in Systems Research. 1 Credit.
This course covers latest advances in the research of computer systems including operating systems, distributed system, mobile and cloud computing. Students will read and discuss recent research papers in top systems conferences. Each week, one student will present the paper and lead the discussion for the week. The focus topics covered in the papers vary semester to semester. Example topics include fault-tolerance, reliability, verification, energy efficiency, and virtualization.

EN.601.819. Selected Topics in Cloud Computing and Networked Systems. 1 Credit.
Participants will read and discuss seminal and recent foundational research on cloud and networked systems.

EN.601.826. Selected Topics in Programming Languages. 1 Credit.
This seminar course covers recent developments in the foundations of programming language design and implementation. Topics covered include type theory, process algebra, higher-order program analysis, and constraint systems. Students will be expected to present papers orally.

EN.601.831. CS Theory Seminar. 1 Credit.
Seminar series in theoretical computer science. Topics include algorithms, complexity theory, and related areas of TCS. Speakers will be a mix of internal and external researchers, mostly presenting recently published research papers.

EN.601.833. Seminar in Algorithms. 1 Credit.
This course will explore algorithms and theoretical computer science with a focus on algorithms for massive data. Examples of topics include streaming algorithms, approximation algorithms, online algorithms. Students will be encouraged to select a paper and lead a discussion. External speakers will be invited to present current work as well. This course is a good opportunity for motivated students to learn modern algorithmic methods.Recommended Course Background: EN.601.433 or equivalent.
Area: Engineering

EN.601.845. Selected Topics in Applied Cryptography. 1 Credit.
In this course students will read, discuss and present current research papers in applied cryptography. Topic coverage will vary each semester. Instructor approval required.

EN.601.849. Selected Topics in Computational Immunogenomics. 1 Credit.
Immunology studies defensive mechanisms of living organisms against external threats. Computational immunogenomics is a new branch of bioinformatics that develops and applies computational approaches to the study and interpretation of immunological data, seeking to answer questions about human adaptive immune responses to various pathogens, including but not limited to flu, HIV, and SARS-CoV-2. In this course, students will attend lectures and present immunogenomics papers in a journal club format.

EN.601.856. Seminar: Medical Image Analysis. 1 Credit.
This weekly seminar will focus on research issues in medical image analysis, including imagesegmentation, registration, statistical modeling, and applications. It will also include selected topicsrelating to medical image acquisition, especially where they relate to analysis. The purpose of thecourse is to provide the participants with a thorough background in current research in these areas, as well as to promote greater awareness and interaction between multiple research groups withinthe University. The format of the course is informal. Students will read selected papers. All students will be assumed to have read these papers by the time the paper is scheduled for discussion. But individual students will be assigned on a rotating basis to lead the discussion on particular papers or sections of papers. Co-listed with En.520.746.

EN.601.857. Selected Topics in Computer Graphics. 1 Credit.
In this course we will review current research in computer graphics. We will meet for an hour once a week and one of the participants will lead the discussion for the week.

EN.601.864. Selected Topics in Multilingual Natural Language Processing. 1 Credit.
This is a weekly reading group focused on Natural Language Processing (NLP) outside of English. Whereas methods have gotten very strong in recent years on English NLP tasks, many methods fail on other languages due to both linguistic differences as well as lack of available annotated resources. This course will focus on Cross-Language Information Retrieval, Cross-Lingual Information Extraction, Multilingual Semantics, Massively Multilingual Language Modeling, and other non-English NLP sub-fields. Students will be expected to read, discuss, and present papers. Required course background: EN.601.465/665.

EN.601.865. Selected Topics in Natural Language Processing. 1 Credit.
A reading group exploring important current research in the field and potentially relevant material from related fields. Enrolled students are expected to present papers and lead discussion.
Prerequisite(s): EN.601.465 OR EN.601.665.

EN.601.866. Selected Topics in Computational Semantics. 1 Credit.
This weekly reading group will review current research and survey articles on the topics of computational semantics, statistical machine translation, and natural language generation. Enrolled students will present papers and lead discussions.

EN.601.868. Selected Topics in Machine Translation. 1 Credit.
Students in this course will review, present, and discuss current research in machine translation. Permission of instructor.

EN.601.875. Selected Topics in Machine Learning. 1 Credit.
This seminar is recommended for all students interested in data intensive computing research areas (e.g., machine learning, computer vision, natural language processing, speech, computational social science). The meeting format is participatory. Papers that discuss best practices and the state-of-the-art across application areas of machine learning and data intensive computing will be read. Student volunteers lead individual meetings. Faculty and external speakers present from time-to-time. Recommended Course Background: machine learning or permission of the instructor.

Cross Listed Courses

Biomedical Engineering

EN.580.488. Foundations of Computational Biology and Bioinformatics. 3 Credits.
This course is designed to give students a foundation in the basics of statistical and algorithmic approaches developed in computational biology/bioinformatics over the past 30 years, while emphasizing the need to extend these approaches to emerging problems in the field. Topics covered include probabilistic modeling applied to biological sequence analysis, supervised machine learning, interpretation of genetic variants, cancer genomics bioinformatic workflows and computational immuno-oncology. Attending the lab section "Annotate Your Genome" is required.
Prerequisite(s): EN.601.220
Area: Engineering, Natural Sciences

EN.580.688. Foundations of Computational Biology and Bioinformatics. 3 Credits.
This course will introduce probabilistic modeling and information theory applied to biological sequence analysis, focusing on statistical models of protein families, alignment algorithms, and models of evolution. Topics will include probability theory, score matrices, hidden Markov models, maximum likelihood, expectation maximization and dynamic programming algorithms. Homework assignments will require programming in Python. Recommended Course Background: Math through linear algebra and differential equations, EN.580.221 or equivalent, EN.601.226 or equivalent.

EN.580.709. Sparse Representations in Computer Vision and Machine Learning. 3 Credits.
Sparse and redundant representations constitutes a fascinating area of research in signal and image processing. This is a relatively young field that has been taking form for the last 15 years or so, with contributions from harmonic analysis, numerical algorithms and machine learning, and has been vastly applied to myriad of problems in computer vision and other domains. This course will focus on sparsity as a model for general data, generalizing many different other constructions or priors. This idea - that signals can be represented with just "a few" coefficients - leads to a long series of beautiful (and surprisingly, solvable) theoretical and numerical problems, and many applications that can benefit directly from the new developed theory. In this course we will survey this field, starting with the theoretical foundations, and systematically covering the knowledge that has been gathered in the past years. This course will touch on theory, numerical algorithms, and applications in image processing and machine learning. Recommended course background: Linear Algebra, Signals and Systems, Numerical Analysis.

EN.580.743. Advanced Topics in Genomic Data Analysis. 3 Credits.
Genomic data is becoming available in large quantities, but understanding how genetics contributes to human disease and other traits remains a major challenge. Machine learning and statistical approaches allow us to automatically analyze and combine genomic data, build predictive models, and identify genetic elements important to disease and cellular processes. This course will cover current uses of statistical methods and machine learning in diverse genomic applications including new genomic technologies. Students will present and discuss current literature. Topics include personal genomics, integrating diverse genomic data types, new technologies such as single cell sequencing and CRISPR, and other topics guided by student interest. The course will include a project component with the opportunity to explore publicly available genomic data. Recommended Course Background: coursework in data science or machine learning.

EN.580.745. Mathematics of Deep Learning. 1.5 Credits.
The past few years have seen a dramatic increase in the performance of recognition systems thanks to the introduction of deep networks for representation learning. However, the mathematical reasons for this success remain elusive. For example, a key issue is that the training problem is nonconvex, hence optimization algorithms are not guaranteed to return a global minima. Another key issue is that while the size of deep networks is very large relative to the number of training examples, deep networks appear to generalize very well to unseen examples and new tasks. This course will overview recent work on the theory of deep learning that aims to understand the interplay between architecture design, regularization, generalization, and optimality properties of deep networks. Recommended background: machine learning (EN.601.475), deep learning (EN.520.438 or EN.601.482), graduate-level matrix analysis and linear algebra (EN.553.792) and graduate-level optimization (EN.553.762).

Center for Leadership Education

EN.660.345. Multidisciplinary Engineering Design 1. 3 Credits.
This course number was formally EN.500.308. Students will work on teams with colleagues from different engineering disciplines to tackle a challenge for a clinical, community, or industry project partner. Through practicing a creative, human-centered design process, teams will understand the essential need behind the problem, prototype solutions, and test and refine their prototypes. In addition to project work, students will learn healthy team dynamics and how to collaborate among different working styles.Students will work on teams with colleagues from different engineering disciplines to tackle a challenge for a clinical, community, or industry project partner. Through practicing a creative, human-centered design process, teams will understand the essential need behind the problem, prototype solutions, and test and refine their prototypes. In addition to project work, students will learn healthy team dynamics and how to collaborate among different working styles. Students may choose to move their projects forward towards implementation in Multidisciplinary Engineering Design 2 in spring 2023.
Area: Engineering

Electrical & Computer Engineering

EN.520.216. Introduction To VLSI. 3 Credits.
This course teaches the basics of switch-level digital CMOS VLSI design. This includes creating digital gates using MOS transistors as switches, laying out a design using CAD tools, and checking the design for conformance to the Scalable CMOS design rules.Recommended: EN.520.213.
Prerequisite(s): (AS.171.101 AND AS.171.102) OR (AS.171.101 AND AS.171.108) OR (AS.171.102 AND AS.171.107) OR (AS.171.107 AND AS.171.108)
Area: Engineering

EN.520.349. Microprocessor Lab I. 3 Credits.
This course introduces the student to the programming of microprocessors at the machine level. 68HC08, 8051, and eZ8 microcontrollers are programmed in assembly language for embedded control purposes. The architecture, instruction set, and simple input/output operations are covered for each family. Upon completion, students can use these flash-based chips as elements in other project courses. Recommended Course Background: EN.520.142 or equivalent.The lab is open 24/7 and students can still take the class if they are unable to meet during lab time.
Prerequisite(s): Students must have completed Lab Safety training prior to registering for this class. To access the tutorial, login to myLearning and enter 458083 in the Search box to locate the appropriate module.
Area: Engineering

EN.520.412. Machine Learning for Signal Processing. 3 Credits.
This course will focus on the use of machine learning theory and algorithms to model, classify and retrieve information from different kinds of real world complex signals such as audio, speech, image and video.
Prerequisite(s): (AS.110.201 AND EN.553.310 AND EN.520.344) OR (AS.110.201 AND EN.553.311 AND EN.520.344) OR (AS.110.201 AND EN.553.420 AND EN.520.344) OR (AS.110.201 AND EN.553.421 AND EN.520.344);Students can only take EN.520.412 OR EN.520.612, not both.
Area: Engineering

EN.520.424. FPGA Synthesis Lab. 3 Credits.
An advanced laboratory course in the application of FPGA technology to information processing, using VHDL synthesis methods for hardware development. The student will use commercial CAD software for VHDL simulation and synthesis, and implement their systems in programmable XILINX 20,000 gate FPGA devices. The lab will consist of a series of digital projects demonstrating VHDL design and synthesis methodology, building up to final projects at least the size of an 8-bit RISC computer. Projects will encompass such things as system clocking, flip-flop registers, state-machine control, and arithmetic. The students will learn VHDL methods as they proceed through the lab projects, and prior experience with VHDL is not a prerequisite.
Prerequisite(s): Students must have completed Lab Safety training prior to registering for this class. To access the tutorial, login to myLearning and enter 458083 in the Search box to locate the appropriate module.
Area: Engineering, Quantitative and Mathematical Sciences

EN.520.432. Medical Imaging Systems. 3 Credits.
This course provides students with an introduction to the physics, instrumentation, and signal processing methods used in general radiography, X-ray computed tomography, ultrasound imaging, magnetic resonance imaging, and nuclear medicine. The primary focus is on the methods required to reconstruct images within each modality from a signals and systems perspective, with emphasis on the resolution, contrast, and signal-to-noise ratio of the resulting images. Students will additionally engage in hands-on activities to reconstruct medical images from raw data.
Prerequisite(s): EN.520.214 OR EN.580.222 OR (EN.580.243 AND EN.580.246)
Area: Engineering

EN.520.433. Medical Image Analysis. 3 Credits.
This course covers the principles and algorithms used in the processing and analysis of medical images. Topics include, interpolation, registration, enhancement, feature extraction, classification, segmentation, quantification, shape analysis, motion estimation, and visualization. Analysis of both anatomical and functional images will be studied and images from the most common medical imaging modalities will be used. Projects and assignments will provide students experience working with actual medical imaging data.
Prerequisite(s): EN.550.310 OR EN.550.311 OR EN.560.348
Area: Engineering

EN.520.447. Information Theory. 3 Credits.
This course will address some basic scientific questions about systems that store or communicate information. Mathematical models will be developed for (1) the process of error-free data compression leading to the notion of entropy, (2) data (e.g. image) compression with slightly degraded reproduction leading to rate-distortion theory and (3) error-free communication of information over noisy channels leading to the notion of channel capacity. It will be shown how these quantitative measures of information have fundamental connections with statistical physics (thermodynamics), computer science (string complexity), economics (optimal portfolios), probability theory (large deviations), and statistics (Fisher information, hypothesis testing).
Prerequisite(s): EN.553.310 OR EN.553.420 OR EN.553.311;Students can earn credit for either EN.520.447 or EN.520.647, but not both.
Area: Engineering, Quantitative and Mathematical Sciences

EN.520.448. Electronics Design Lab. 3 Credits.
An advanced laboratory course in which teams of students design, build, test and document application specific information processing microsystems. Semester long projects range from sensors/actuators, mixed signal electronics, embedded microcomputers, algorithms and robotics systems design. Demonstration and documentation of projects are important aspects of the evaluation process.Recommended: EN.600.333, EN.600.334, EN.520.214, EN.520.216, EN.520.349, EN.520.372, EN.520.490 or EN.520.491.
Prerequisite(s): (EN.520.240 OR EN.520.340 OR EN.520.230 OR EN.520.213) AND AS.110.108 AND AS.110.109 AND ((171.101 AND 171.102) OR (171.101 AND 171.108) OR (171.102 AND 171.107) OR (171.107 AND 171.108)) AND EN.520.142.;Students must have completed Lab Safety training prior to registering for this class. To access the tutorial, login to myLearning and enter 458083 in the Search box to locate the appropriate module.

EN.520.450. Advanced Micro-Processor Lab. 3 Credits.
This course covers the usage of common microcontroller peripherals. Interrupt handling, timer operations, serial communication, digital to analog and analog to digital conversions, and flash ROM programming are done on the 68HC08, 8051, and eZ8 microcontrollers. Upon completion, students can use these flash-based chips as elements in other project courses. Recommended Course Background: EN.520.349
Prerequisite(s): Students must have completed Lab Safety training prior to registering for this class. To access the tutorial, login to myLearning and enter 458083 in the Search box to locate the appropriate module.

EN.520.491. CAD Design of Digital VLSI Systems I (Juniors/Seniors). 3 Credits.
Juniors and Seniors Only.
Prerequisite(s): Student may take EN.520.491 or EN.520.691, but not both.;AS.110.109 AND (AS.171.102 OR AS.171.104 OR AS.171.108) AND EN.520.142 AND EN.520.142 AND (EN.520.230 OR (EN.520.213 AND EN.520.345 OR EN.520.216))
Area: Engineering

EN.520.612. Machine Learning for Signal Processing. 3 Credits.
This course will focus on the use of machine learning theory and algorithms to model, classify and retrieve information from different kinds of real world complex signals such as audio, speech, image and video. Recommended Course Background: AS.110.201, EN.553.310, and EN.520.435.
Prerequisite(s): Students must have completed Lab Safety training prior to registering for this class. To access the tutorial, login to myLearning and enter 458083 in the Search box to locate the appropriate module.;Credit may only be earned for EN.520.412 or EN.520.612.
Area: Engineering

EN.520.637. Foundations of Reinforcement Learning. 3 Credits.
The course will provide a rigorous treatment of reinforcement learning by building on the mathematical foundations laid by optimal control, dynamic programming, and machine learning. Topics include model-based methods such as deterministic and stochastic dynamic programming, LQR and LQG control, as well as model-free methods that are broadly identified as Reinforcement Learning. In particular, we will cover on and off-policy tabular methods such as Monte Carlo, Temporal Differences, n-step bootstrapping, as well as approximate solution methods, including on- and off-policy approximation, policy gradient methods, including Deep Q-Learning. The course has a final project where students are expected to formulate and solve a problem based on the techniques learned in class.
Area: Engineering

EN.520.666. Information Extraction. 3 Credits.
Introduction to statistical methods of speech recognition (automatic transcription of speech) and understanding. The course is a natural continuation of EN.601.465 but is independent of it. Topics include elementary probability theory, hidden Markov models, and n-gram models using maximum likelihood, Bayesian and discriminative methods, and deep learning techniques for acoustic and language modeling.Recommended Course Background: EN.550.310 AND EN.600.120 or equivalent, expertise in Matlab or Python programming.

EN.520.807. Current Topics in Language and Speech Processing. 1 Credit.
This biweekly seminar will cover a broad range of current research topics in human language technology, including automatic speech recognition, natural language processing and machine translation. The Tuesday seminars will feature distinguished invited speakers, while the Friday seminars will be given by participating students. A minimum of 75% attendance and active participation will be required to earn a passing grade. Grading will be S/U.

General Engineering

EN.500.112. Gateway Computing: JAVA. 3 Credits.
This course introduces fundamental programming concepts and techniques, and is intended for all who plan to develop computational artifacts or intelligently deploy computational tools in their studies and careers. Topics covered include the design and implementation of algorithms using variables, control structures, arrays, functions, files, testing, debugging, and structured program design. Elements of object-oriented programming. algorithmic efficiency and data visualization are also introduced. Students deploy programming to develop working solutions that address problems in engineering, science and other areas of contemporary interest that vary from section to section. Course homework involves significant programming. Attendance and participation in class sessions are expected.
Prerequisite(s): Students may not have earned credit in courses: EN.500.113 OR EN.500.114 OR EN.510.202 OR EN.530.112 OR EN.580.200 OR EN.601.107 OR EN.500.132 OR EN.500.133 OR EN.500.134.
Area: Engineering

EN.500.132. Bootcamp: Java. 1 Credit.
This on-line course provides students who have already achieved a basic understanding of programming and computational thinking in one programming language with an opportunity to apply these skills in another programming language. Students will be expected to complete projects to demonstrate proficiency in the new language. Satisfactory/unsatisfactory only.
Prerequisite(s): Not open to students who have completed EN.601.107, EN.600.107, or EN.500.112;Students must have completed: EN.500.113 OR EN.500.114 OR EN.510.202 OR EN.580.200 OR EN.530.112 OR EN.520.123 OR EN.601.220
Area: Engineering

EN.500.133. Bootcamp: Python. 1 Credit.
This on-line course provides students who have already achieved a basic understanding of programming and computational thinking in one programming language with an opportunity to apply these skills in another programming language. Students will be expected to complete projects to demonstrate proficiency in the new language. Satisfactory/unsatisfactory only
Prerequisite(s): Not open to students who have completed EN.500.113 or EN.580.200;Students must have completed: EN.500.112 OR EN.500.114 OR EN.601.107 OR EN.510.202 OR EN.530.112 OR EN.520.123 OR EN.601.220
Area: Engineering

EN.500.134. Bootcamp: MATLAB. 1 Credit.
This on-line course provides students who have already achieved a basic understanding of programming and computational thinking in one programming language with an opportunity to apply these skills in another programming language. Students will be expected to complete projects to demonstrate proficiency in the new language. Satisfactory/unsatisfactory only.
Prerequisite(s): Not open to students who have completed EN.500.114 OR EN.580.200.;Students must have completed: EN.500.112 OR EN.500.113 OR EN.601.107 OR EN.510.202 OR EN.530.112 OR EN.520.123 OR EN.601.220
Area: Engineering

Information Security Institute

EN.650.624. Network Security. 3 Credits.
This course focuses on communication security in computer systems and networks. The course is intended to provide students with an introduction to the field of network security. The course covers network security services such as authentication and access control, integrity and confidentiality of data, firewalls and related technologies, Web security and privacy. Course work involves implementing various security techniques. A course project is required. Course Background: EN.601.220, EN.601.226, EN.601.418 or equivalent.
Prerequisite(s): Students may only earn credit for one of the following courses: EN.650.624 OR EN.601.444 OR EN.601.644
Area: Engineering

EN.650.631. Ethical Hacking. 3 Credits.
Cyber security affects every facet of industry and our government, and thus is now a threat to National Security. This course is designed to introduce students to the skills needed to defend computer network infrastructure by exposing them to the hands-on identification and exploitation of vulnerabilities in servers (i.e., Windows and Linux), wireless networks, websites, and cryptologic systems. These skills will be tested by having teams of students develop and participate in instructor lead capture-the-flag competitions. Also included are advanced topics such as shell coding, IDA Pro analysis, fuzzing, and writing or exploiting network-based applications or techniques such as web servers, spoofing, and denial of service.
Area: Engineering

EN.650.654. Computer Intrusion Detection. 3 Credits.
Intrusion detection supports the on-line monitoring of computer system activities and the detection of attempts to compromise normal services. This course starts with an overview of intrusion detection tasks and activities. Detailed discussion introduces a traditional classification of intrusion detection models, applications in host-centered and distributed environments, and various intrusion detection techniques ranging from statistical analysis to biological computing. This course serves as a comprehensive introduction of recent research efforts in intrusion detection and the challenges facing modern intrusion detection systems. Students will also be able to pursue in-depth study of special topics of interest in course projects.
Area: Engineering, Natural Sciences

EN.650.656. Computer Forensics. 3 Credits.
This course introduces the student to the field of applied Computer Forensics as practiced by corporate security and law enforcement personnel. The emphasis is on "dead box" (powered off) data extraction and analysis with open-source tools. Topics covered include legal and regulatory issues, forensic imaging and data acquisition from a "dead" system, computer file systems (FAT/NTFS) and data recovery, Windows Registry and configuration records, Windows log analysis and operating system artifacts, memory dump analysis (RAM), software artifacts, computer network forensics, introductory mobile device forensics, case reporting and documentation, end-to-end computer forensic examinations, peer review, and testifying in court.
Area: Engineering

EN.650.660. Software Vulnerability Analysis. 3 Credits.
Competent execution of security assessments on modern software systems requires extensive knowledge in numerous technical domains and comprehensive understanding of security risks. This course provides necessary background knowledge and examines relevant theories for software vulnerabilities and exploits in detail. Key topics include historical vulnerabilities, their corresponding exploits, and associated risk mitigations. Fundamental tools and techniques for performing security assessments (e.g., software reverse engineering, static analysis, and dynamic analysis) are covered extensively. The format of this course includes lectures and assignments where students learn how to develop exploits to well-known historical vulnerabilities in a controlled environment. Students will complete and demonstrate a project as part of the course.
Area: Engineering

EN.650.663. Cloud Computing Security. 3 Credits.
Cloud computing promises significant cost savings via economies of scale that typically are not achievable by a single organization. This course examines cloud computing in detail and introduces the security concerns associated with cloud computing. Key topics include service models for cloud computing, virtualization, storage, management, and data processing. Fundamental security principles are introduced and applied to cloud computing environments. The format of this course includes lectures and hands-on assignments. Students will complete a project and present it as part of the course.
Area: Engineering, Natural Sciences

EN.650.757. Advanced Computer Forensics. 3 Credits.
This course will analyze advanced topics and state of the art issues in the field of digital forensics. The course will be run in a research seminar format and students will be given both basic and applied research projects in such areas as: intrusion analysis, network forensics, memory forensics, mobile devices, and other emerging issues.

Mechanical Engineering

EN.530.707. Robot System Programming. 4 Credits.

This course seeks to introduce students to open-source software tools that are available today for building complex experimental and fieldable robotic systems. The course is grouped into sections, each of which building on the previous in increasing complexity and specificity: tools and frameworks supporting robotics research, robotics-specific software frameworks, integrating complete robotic systems, and culminates with an independent project of the student's own design using small mobile robots or other robots in the lab. Students will need to provide a computer (with at least a few GB of memory and a several tens of GB of disc space) running Ubuntu (https://www.ubuntu.com or one of its variants such as Xubuntu) and ROS (http://ros.org/). Students should have an understanding of intermediate programming in C/C++ (including data structures and object-oriented programming). Familiarity with Linux programming. Familiarity with software version control systems such as Git, and linear algebra. Students should see the course homepage https://dscl.lcsr.jhu.edu/home/courses/me530707-2019 for more information and to get started with the course. Required Course Prerequisite/Corequisite: EN.530.646 and EN.601.436/663. No audit option.

Robotics

EN.620.745. Seminar in Computational Sensing and Robotics. 1 Credit.

Seminar series in robotics. Topics include: Medical robotics, including computer-integrated surgical systems and image-guided intervention. Sensor based robotics, including computer vision and biomedical image analysis. Algorithmic robotics, robot control and machine learning. Autonomous robotics for monitoring, exploration and manipulation with applications in home, environmental (land, sea, space), and defense areas. Biorobotics and neuromechanics, including devices, algorithms and approaches to robotics inspired by principles in biomechanics and neuroscience. Human-machine systems, including haptic and visual feedback, human perception, cognition and decision making, and human-machine collaborative systems.Cross-listed Mechanical Engineering, Computer Science, Electrical and Computer Engineering, and Biomedical Engineering.

For current faculty and contact information go to http://cs.jhu.edu/faculty/