Computer Science

The field of computer science is exceedingly exciting, challenging, and pervasive. The availability of relatively inexpensive high performance computing capabilities, ubiquitous high speed and wireless networking, and mobile computing have together created a technology-driven restructuring of the way society and most professions now operate. Information, and its associated processing and transport, is the commodity upon which corporations are built and lives are improved. At the center of this revolution, making it happen, are those who study computer science.

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 worth 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 or health informatics, gaming/entertainment, and business computing, 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. Whether the ultimate goal is a mainstream career in computer science or a desire to combine expertise in computer science with another area, 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 catalog 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, computer vision and image processing, computer graphics, geometric modeling, programming languages, natural language and speech processing, information retrieval, cryptography and information security, secure and robust systems, storage systems, high-performance and scientific computing, networks and distributed systems, stream processing, parallel and distributed databases, robotics, computer-integrated surgical systems, and wireless and sensor systems.

Additionally, interdisciplinary research centers in the university have heavy involvement by Computer Science faculty: the Information Security Institute (ISI), the Center for Computer-Integrated Surgical Systems and Technology (CISST), the Laboratory for Computational Sensing and Robotics (LCSR), the Center for Language and Speech Processing (CLSP), and the Institute for Data Intensive Engineering and Science (IDIES). An important component of the educational process in the department is the opportunity for student participation in the research programs of the faculty, and all faculty members have research laboratories in which individual projects are available for undergraduate and graduate students. Original research in close association with individual faculty members is emphasized at the graduate level.

There are several closely related programs which involve significant coursework and faculty involvement from the Department of Computer Science. A minor in Computer Integrated Surgery is administered by the Engineering Research Center for Computer Integrated Surgical Systems and Technology in the Laboratory for Computational Sensing and Robotics (LCSR). The LCSR also offers a minor in robotics. Details of these programs may be found in this catalog section following the minor in computer science. Undergraduates with a strong interest in system design and performance may elect to pursue a bachelor degree in computer engineering. This field of study includes course work in computer science, as well as electrical and computer engineering. Although jointly administered by both departments, specific goals and requirements of the computer engineering degrees may be found in the catalog section pertaining to the Department of Electrical and Computer Engineering only. Lastly, the Master of Science in Security Informatics (MSSI) is a specialized graduate program offered through the Information Security Institute (ISI) in the WSE. 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 MSSI online at www.jhuisi.jhu.edu or in the ISI section of this catalog.

For additional information regarding the academic programs available, and the facilities provided, please consult the sections which follow, or the departmental website www.cs.jhu.edu or the department office, 224 New Engineering Bldg, Baltimore, MD 21218-2694.

The Faculty

Yanif Ahmad, Assistant Professor: data management, stream processing, declarative languages, parallel and distributed databases.

Yair Amir, Professor: systems, distributed algorithms, secure distributed systems, overlay networks, wireless backbones, replication, survivability.

Giuseppe Ateniese, Associate Professor: applied cryptography, network security, and secure e-commerce.

Baruch Awerbuch, Professor: wireless networks, algorithmic theory of communication networks, online and distributed computing.

Vladimir Braverman, Assistant Professor: algorithms, massive data sets, data streams, and database systems.

Randal Burns, Associate Professor: storage systems, high performance and scientific computing, and database federations.

Jason M. Eisner, Associate Professor: computational linguistics (syntax and phonology), natural language processing, statistical machine learning, programming language design.

Peter Fröhlich, Senior Lecturer: programming languages, software engineering, systems software.

Gregory D. Hager, Professor (Chair): vision, robotics, human-machine systems, computer-assisted surgery.

Susan Hohenberger, Assistant Professor: cryptography, computer security, algorithms, and complexity theory.

Michael Kazhdan, Assistant Professor: computer graphics, 3D shape analysis, 3D shape matching.

S. Rao Kosaraju, Edward J. Schaefer Professor in Engineering: design of algorithms, parallel computation, pattern matching, computational geometry.

Gerald M. Masson, Professor (Director, Information Security Institute): computer engineering, fault-tolerant computing, computer communications and networking.

Aviel Rubin, Professor (Technical Director, Information Security Institute): system and networking security, computer privacy, applied cryptography.

Joanne Selinski, Senior Lecturer and Director of Undergraduate Studies: CS education, graph theory.

Scott F. Smith, Professor: programming languages, type systems, security in language design, component programming languages.

Russell H. Taylor, Professor (Director, CISST ERC): medical robotics, computer-integrated interventional medicine, medical image analysis, human-machine systems.

Andreas Terzis, Associate Professor: P2P, overlay and sensor networks, resilient internet infrastructure, NP-based architectures.

David Yarowsky, Professor: natural language and speech processing, information retrieval, machine translation, and machine learning.

Joint Appointments

Joel Bader, Associate Professor (Biomedical Engineering): bioinformatics and computational biology.

Emad Boctor, Assistant Professor (Radiology-Medical Imaging Physics): image-guided intervention, ultrasound imaging, elasticity, and thermal imaging.

Gregory Chirikjian, Professor (Mechanical Engineering): robotics, kinematics, dynamics, control, motion planning.

Noah Cowan, Assistant Professor (Mechanical Engineering): sensor-based control of locomotion and manipulation, and biologically inspired robotics.

Ralph Etienne-Cummings, Professor (Electrical and Computer Engineering): mixed-signal VLSI, computational sensors, robotics, neuromorphic engineering.

James Fill, Professor (Applied Mathematics and Statistics): probability, stochastic processes, random structures, and algorithms.

Rachel Karchin, Assistant Professor (Biomedical Engineering): computational molecular biology, bioinformatics, genetic variation.

Sanjeev Khudanpur, Associate Professor (Electrical and Computer Engineering): information theory, statistical language modeling for speech recognition and machine translation.

Han Liu, Assistant Professor (Biostatistics): statistical machine learning, high dimensional nonparametric learning and massive-data analysis, multiple hypothesis testing, time series analysis, genomics, proteomics, cognitive neuroscience.

Michael I. Miller, Professor (Biomedical Engineering): image understanding, computer vision, medical imaging, computational anatomy.

Carey Priebe, Professor (Applied Mathematics and Statistics): computational statistics, kernel and mixture estimates, statistical pattern recognition, and statistical image analysis.

Jerry L. Prince, William B. Kouwenhoven Professor (Electrical and Computer Engineering) (Associate Director for Research, CISST ERC): image processing, computer vision, medical imaging.

Jeff Siewerdsen, Associate Professor (Biomedical Engineering): imaging physics, diagnostic radiology, image-guided interventions.

Alexander Szalay, Professor (Physics and Astronomy): theoretical astrophysics, galaxy formation.

Rene Vidal, Associate Professor (Biomedical Engineering): computer vision, machine learning, robotics, and control.

Louis Whitcomb, Professor (Mechanical Engineering): dynamics and control of mechanical systems.

Raimond L. Winslow, Professor (Biomedical Engineering): modeling of biological systems, nonlinear systems theory, grid computing and data management, biomedical ontologies.

Adjunct, Research, and Visiting Faculty

Amihood Amir, Research Professor: algorithms design and analysis, multidimensional pattern matching, knowledge discovery algorithms, real time systems algorithms, computational molecular biology.

Mitra Basu, Visiting Professor: computational biology, pattern recognition, neural networks, artificial intelligence.

Philippe Burlina, Assistant Research Professor: computer vision, visual analysis and communications, multi-modality image exploitation, enterprise software systems for content and e-process management.

Chris Callison-Burch, Assistant Research Professor: statistical natural language processing, machine translation, paraphrasing, evaluation of human language technologies.

Kenneth Church, Research Professor: natural language processing, speech, data mining.

Robert Cole, Assistant Research Professor: data networking, performance modeling, internet protocol design and mobile ad-hoc networks (MANETS).

Bharat Doshi, Research Professor: optical and wireless networking technologies, internet protocols and architectures, speech technologies and signal processing, and network design and analysis algorithms and tools.

Mark Dredze, Assistant Research Professor: machine learning, natural language processing, intelligent user interfaces, intelligent email.

Gabor Fichtinger, Associate Research Professor: applied surgical robotics, surgical CAD/CAM systems, percutaneous therapies, stereotactic radiosurgery.

Matthew Green, Assistant Research Professor: applied cryptography, cryptographic protocol design, analysis of practical security systems, privacy-preserving storage and identification technologies.

John Linwood Griffin, Assistant Research Professor: data protection in information storage systems and networks, computer virtualization and performance.

Ragib Hasan, Assistant Research Scientist: computer security, secure provenance, trust management, storage systems.

Peter Kazanzides, Associate Research Professor: medical robotics, computer-assisted surgery, real-time systems.

Rajesh Kumar, Associate Research Professor: applications of robotics and vision in medicine and surgery.

Michael Lavine, Assistant Research Professor: computer forensics, information assurance and security, critical infrastructure protection.

Adam Lopez, Assistant Research Scientist: natural language processing, machine learning, language and automata theory, algorithms.

James Mayfield, Associate Research Professor: information retrieval, cross-language retrieval, information extraction, natural language processing.

Amitabh Mishra, Assistant Research Professor: wireless cellular, ad hoc and sensor networks, dynamic spectrum access networks, telecommunications.

Fabian Monrose, Associate Research Professor: computer and network security, biometrics and user authentication.

Christine Piatko, Assistant Research Professor: computational geometry, information visualization, information retrieval.

John W. Sheppard, Associate Research Professor: artificial intelligence, machine learning, data mining.

Sam Small, Assistant Research Scientist: systems and network security.

Veselin Stoyanov, Assistant Research Scientist: sentiment analysis, corefrence resolution, information extraction, natural language processing, machine learning, artificial intelligence, cognitive science, linguistics.

Jonathan Trostle, Assistant Research Professor: network and operating system security, cryptography, network security management.

Ben Van Durme, Assistant Research Professor: artificial intelligence, natural language processing (computational semantics), and streaming algorithms.

I-Jeng Wang, Assistant Research Professor: wireless networking, Bayesian networks, probabilistic models.

Theresa Wilson, Assistant Research Scientist: artificial intelligence, computational linguistics, natural language processing, subjectivity and sentiment analysis in text and speech.

Lawrence B. Wolff, Research Professor: computer vision, multi-sensor image fusion, augmented reality, biometrics.

Qinqing Zhang, Assistant Research Professor: wireless communications and networking, Mobile Ad-hoc networks, cellular system and network technologies, multimedia applications and QoS, Internet protocol and algorithm design, performance analysis.

Part-Time Lecturers

Sheela Kosaraju: computer ethics.

Harold Lehmann: medical informatics.

Facilities

The general department computing facilities include over 70 workstations and servers; a large undergraduate laboratory comprised of 24 Linux workstations, 12 Windows stations, and a separate collaboration room allowing students to work in a team-based environment; a Masters' Students Office consisting of 16 Linux workstations and a collaboration area; assigned locations and computers for Ph.D. students; multiple high-speed networked laser printers, as well as a networked color copier; remotely accessible Linux and Unix computer servers available to both graduate and undergraduate students.

Focused research laboratories have significant resources that provide greater specialization, including isolated networks of PCs for security studies, sensor and wireless computing testbeds, 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.

Undergraduate Programs

(See also General Requirements for Departmental Majors)

The objectives of our bachelor degree programs are to train computer scientists who will be able to:

- Successfully engage in professional practice in the computing sciences or apply computer science tools and techniques to another field of interest.
- Pursue advanced study in the computing sciences.
- Behave in a professional and ethical manner.
- Work successfully in both independent and team environments.

A successful major program of study leads to either the bachelor of science in computer science (B.S.) or the bachelor of arts in computer science (B.A.). Students should decide which degree program to complete by about their junior year. Both degree programs require specific courses and/or credits in several key areas: computer science, math, basic science, humanities and social sciences. However, there is much flexibility in how these requirements are fulfilled. Undergraduate majors may choose to pursue a broad selection of computer science and distributional courses, or to pursue a specific concentration within the field. Current concentrations reflect departmental and school strengths: information security, natural language processing, robotics, software engineering, and video game design. Further information on these concentrations may be found in the computer science undergraduate advising manual.

All undergraduate students majoring or minoring in computer science must have a faculty advisor in the department. They will be assigned an advisor as entering freshmen or upon deciding on the major/minor. Every major must follow a program approved by his/her faculty advisor.

The department also offers a minor in computer science, and tangentially, a minor in computer integrated surgery and a minor in robotics. Some students majoring in computer science may be eligible for a concurrent bachelor's/master's degree program. Requirements for these programs are included here as well. Additional details regarding undergraduate programs can be found in the department's undergraduate advising manual or on the website at www.cs.jhu.edu.

Requirements for the B.S. Degree

The bachelor of science degree in computer science degree program is accredited by the Computing Accreditation Commission of ABET,

www.abet.org. It provides for the acquisition of the following knowledge base and skill set:

- An ability to apply knowledge of computing and mathematics appropriate to the discipline.
- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
- An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
- An ability to function effectively on teams to accomplish a common goal.
- An understanding of professional, ethical, legal, security, and social issues and responsibilities.
- An ability to communicate effectively with a range of audiences.
- An ability to analyze the local and global impact of computing on individuals, organizations, and society.
- Recognition of the need for and an ability to engage in continuing professional development.
- An ability to use current techniques, skills, and tools necessary for computing practice.
- An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
- An ability to apply design and development principles in the construction of software systems of varying complexity.

To meet the course credit requirements for the B.S. in computer science, the student must complete a minimum of 126 credits. The basic requirements for the B.S. degree are as follows: 42 credits in computer science, 22 credits in mathematics, 16 credits in basic sciences, 18 credits in humanities/social sciences, and 2 writing intensive courses. Details and course recommendations of these distributional requirements are below. These requirements add up to 98 credits and fulfill general university distribution requirements. Of the remaining 28 elective credits that students choose freely, at least 12 of them must be in the humanities, social science, arts, or other disciplines that serve to broaden the student's background. These 12 credits may not be CS, ECE or math courses. Freshman majors are also expected to take 600.105 M&Ms (optional for transfers into the major). This is a one credit S/U course that may only be counted as an elective.

Except for electives, courses should not be taken on a satisfactory/unsatisfactory basis. By university policy, no more than 18 D or D+ credits can be counted toward the total credit requirements for a degree.

Computer Science (42 credits):

- The following foundational courses in computer science must be included in a student's program:
  600.104 Computer Ethics
  600.107 Intro to Programming in Java
  (or AP credit)
  600.120 Intermediate Programming
  600.226 Data Structures
  600.271 Automata and Computation Theory
  600.333 Computer System Fundamentals
- At least 16 credit hours, not including 600.333/433, must be at the 300-level or above. At least one course in each classification area of Analysis, Applications, and Systems must be chosen. An exhaustive list of the area classifications for each of our courses may be found on the department's website.
- Students must take at least one of the following courses which contain oral communication components: 600.255, 600.321/421, 600.355, 600.392, 600.446, 600.520.
- With the advisor's explicit permission regarding course selections, up to 6 of the 42 required credits may be taken in the Department of Electrical and Computer Engineering or the Information Security Institute.
- No more than 6 credits of independent study (including 600.491-492 Computer Science Workshop I, II) and no more than 3 credits of short courses can be counted toward this requirement. However, B.S. students doing the Senior Honors Thesis (600.519-520) may use an additional three credits of independent work toward their CS requirements, for a total of nine credits.
- No courses with grades below C- or with satisfactory/unsatisfactory grades can be used to fulfill this requirement unless they are not offered for a grade.

Mathematics (22 credits):

- The following courses must be included:
  110.108 Calculus I
  110.109 Calculus II
  550.171 Discrete Mathematics
- The remaining courses must be 200-level or above, chosen from Mathematics (110.xxx) or Applied Math and Statistics (550.xxx), and must include coverage of both probability and statistics. Some highly recommended math electives are Probability & Statistics, Calculus III, Linear

Algebra, and Differential Equations. Note that students will need at least six courses to fulfill the credit requirement.

Basic Sciences (16 credits):
At least two semesters of physics or two semesters of chemistry, with the associated laboratories, must be included. The remaining courses must be chosen in accordance with the list posted on the department's website, which includes most courses in Physics, Chemistry, Biology, Biophysics, Earth & Planetary Sciences, and some 'N' designated courses in Neuroscience & Cognitive Science, but not all.

Humanities/Social Sciences (18 credits):
Six courses in the humanities and social sciences must be taken, with each course at least 3 credits. These courses must have either an 'H' or 'S' area designator on them; however, foreign language courses may also be used to satisfy this requirement.

Writing Requirement:
Students are required to fulfill the university's requirement of two writing intensive (W) courses, each at least 3 credits. Students must receive at least a C- grade or better in these writing courses. These courses may overlap other requirements. At least one of the following options must be chosen:

060.113 Expository Writing
220.105/106 Intro to Fiction and Poetry Writing
661.110 Technical Communication

Requirements for the B.A. Degree
To meet the course credit requirements for the B.A. in computer science, the student must complete a minimum of 120 credits. The basic requirements for the B.A. degree are: 30 credits in computer science, 18 credits in mathematics, 12 credits in basic sciences, 18 credits in humanities/social sciences, 6 credits in foreign languages, and 4 writing intensive courses. Details and course recommendations of these distributional requirements are below. These requirements add up to 84 credits and fulfill general university distribution requirements. The remaining 36 credits are electives, to be chosen by the student with the guidance and approval of his/her advisor. Freshman majors are also expected to take 600.105 M&Ms (optional for transfers into the major). This is a one credit S/U course that may only be counted as an elective.

Except for electives, courses should not be taken on a satisfactory/unsatisfactory basis. By university policy, no more than 18 D or D+ credits can be counted toward the total credit requirements for a degree.

Computer Science (30 credits):

- The following foundational courses in computer science must be included in a student's program:
  600.107 Intro to Programming in Java (or AP credit)
  600.120 Intermediate Programming
  600.226 Data Structures
  600.271 Automata and Computation Theory
  600.333 Computer System Fundamentals
- At least 15 credit hours must be at the 300-level or above. With the advisor's permission, up to 6 of the 30 required credits may be taken in the Department of Electrical and Computer Engineering.
- No more than 3 credits of short courses or 3 credits of independent study (including 600.491-492) may be applied toward this requirement. However, B.A. students doing the Senior Honors Thesis (600.519-520) may use an additional 3 credits of independent work toward their CS requirements, for a total of 6 credits.
- No courses with grades below C- or with satisfactory/unsatisfactory grades may be used to fulfill this requirement unless they are not offered for a grade.

Mathematics (18 credits):

- The following courses must be included:
  110.108 Calculus I
  110.109 Calculus II
  550.171 Discrete Mathematics
- The remaining courses may be chosen from Mathematics (110.xxx) or Applied Math and Statistics (550.xxx). At least one course must be 200-level or above. Highly recommended: Calculus III, Linear Algebra, Differential Equations, Probability/Statistics. Note that students will need at least five courses to fulfill the credit requirement.

Basic Sciences (12 credits):
At least two semesters of physics or chemistry or a combination of both, with the associated laboratories, must be included. The remaining courses must be chosen in accordance with the list posted on the department website, which includes most courses in Physics, Chemistry, Biology, Biophysics, Earth & Planetary Sciences, and some 'N' designated courses in Neuroscience & Cognitive Science, but not all.

Humanities/Social Sciences (18 credits):
Six courses in the Humanities/Social Sciences must be taken, with each course at least 3 credits. At least two 3-credit courses at the 300-level or above are required. As befits a B.A. degree, students have ample flexibility to choose courses that broaden the

scope of their study, in consultation with their advisors. A subset of the courses selected to satisfy this requirement should demonstrate coherence within an area. Any course with (H) or (S) area designators may fulfill these distributional requirements.

Foreign Languages (6 credits):
Two courses in a foreign language, with a total of at least 6 credits are required. These foreign language credits are in addition to the 18 required humanities/social sciences credits.

Writing Requirement:
All B.A. candidates in computer science are required to fulfill the university's requirement of four writing intensive (W) courses, each at least 3 credits. Students must receive at least a C- grade in these courses. Highly recommended, at least one of:

060.113 Expository Writing
220.105/106 Intro to Fiction and Poetry Writing
661.110 Technical Communication

Minor in Computer Science

To satisfy the course credit requirements for a minor in computer science, a student must take a minimum of seven courses, with a total of at least 22 credits, earning at least a C- in each course. These must include four core courses, to provide the student with a foundation, and three upper-level courses (300-level and above), to allow the student to pursue an advanced topic in depth.

Core Courses (4):

- 600.107 Intro to Programming in Java (or AP credit)
- 600.120 Intermediate Programming
- 600.226 Data Structures
- 600.271 Automata and Computation Theory

With the approval of a faculty member in the Department of Computer Science, serving as a computer science minor advisor, substitutions for these core courses are possible.

Upper-Level Courses (3):
These courses should be chosen to form a cohesive minor and must be accepted by the computer science minor advisor. It is strongly recommended that students choose all three courses from within one of the three research areas of analysis, applications, and systems. Each upper-level course description in this catalog includes its area for reference. In addition, a current listing of courses grouped by area is provided on the departmental website.

Short courses cannot be used toward the minor requirements. All courses must be taken for a grade, not S/U.

Students whose primary major is in the Whiting School may use the same courses to satisfy the requirements of the primary major and also those of a computer science minor. Students who plan to fulfill requirements for a minor must go to the Department of Computer Science director of undergraduate studies to declare the minor and be advised on course selections, and inform the Office of Academic Advising by the end of their junior year.

Minor in Computer Integrated Surgery

The Department of Computer Science offers a minor in Computer Integrated Surgery (CIS) for full-time, undergraduate students at Johns Hopkins. The minor is particularly well suited for students interested in computer integrated surgery issues who are majoring in a variety of disciplines including biomedical engineering, computer science, computer engineering, electrical engineering, and mechanical engineering. The minor provides formal recognition of the depth and strength of a student's knowledge of the concepts fundamental to CIS beyond the minimal requirements of his/her major.

The Computer Science Department of the Whiting School of Engineering is responsible for the minor in computer integrated surgery. In order to minor in CIS, a student will require a minor advisor from the Engineering Research Center in Computer Integrated Surgical Systems and Technology (CISST ERC) in the Laboratory for Computational Sensing and Robotics. Current faculty members available as advisors include Professors Russell Taylor (CS), Greg Hager (CS), Jerry Prince (ECE), Ralph Etienne-Cummings (ECE), Louis Whitcomb (ME), Noah Cowan (ME), Peter Kazanzides (CS), Rajesh Kumar (CS), Iulian Iordachita (ME), and Emad Boctor (Radiology).

To satisfy the requirements for the minor in CIS, a student must have a fundamental background in computer programming and computer science, sufficient mathematical background, and also take a minimum of six courses (with a total of at least 18 credits, earning at least a C- in each course) directly related to concepts relevant to CIS. These six CIS courses must include three core courses, which provide the student with the fundamental basis for CIS, and three upper-level courses (300 level or above) to allow the student to pursue an advanced CIS topic in depth.

Required Fundamental Computer Science Courses

600.120 Intermediate Programming
600.226 Data Structures
or
Equivalent experience determined by your advisor

Required Fundamental Mathematics Courses

110.106 or 110.108 Calculus I
110.107 or 110.109 Calculus II
110.202 or 110.211-212 Calculus III
550.291, 110.201, or 110.211-12 Linear Algebra

Required Fundamental Computer Integrated Surgery Courses

- 600.445 Computer Integrated Surgery I
- A design course in CIS. Either Computer Integrated Surgery II (600.446) or a design course in biomedical engineering, electrical and computer engineering, or mechanical engineering with substantial CIS content approved by the student's faculty advisor in the CIS minor.
- One course in imaging, chosen from the following:

  600.361 Computer Vision (undergraduate version)
  600.461 Computer Vision (graduate version)
  520.414 Image Processing and Analysis I
  520.432/580.472 Medical Imaging Systems
  or
- One course in robotics, chosen from the following:

  530.420 Robotic Sensors and Actuators
  530.421 Mechatronics
  530.646 Introduction to Robotics
- Three advanced specialty courses chosen in consultation with the student's faculty advisor in the CIS minor which define a topic relevant to CIS (such as CIS instrumentation, CIS imaging, or the mechanics of CIS). Note that these courses must be chosen together with the other three required CIS courses (600.445, the CIS design course and the CIS imaging course) to include at least one biomedical course and must be selected from the following courses or equivalent courses with significant CIS content, as determined by the CIS advisor:

  580.470 Biomedical Sensors
  600.461 Computer Vision
  530.651 Haptic Systems for Teleoperation and Virtual Reality
  520.414 Image Processing and Analysis I
  530.646 Introduction to Robotics
  580.450 Mechanics of Living Tissues
  530.421 Mechatronics
  580.472 Medical Imaging Systems
  580.471 Principles of the Design of Biomedical Instrumentation
  530.420 Robot Sensors and Actuator

Please visit **www.lcsr.jhu.edu/Education/Undergraduate/CISminor** for current course listing.

Minor in Robotics

The field of robotics integrates sensing, information processing, and movement to accomplish specific tasks in the physical world. As such, it encompasses several topics, including mechanics and dynamics, kinematics, sensing, signal processing, control systems, planning, and artificial intelligence. Applications of these concepts appear in many areas including medicine, manufacturing, space exploration, disaster recovery, ordinance disposal, deep-sea navigation, home care, and home automation.

The faculty of the Laboratory for Computational Sensing and Robotics (LCSR), in collaboration with the academic departments and centers of the Whiting School of Engineering, offers a robotics minor in order to provide a structure in which undergraduate students at Johns Hopkins University can advance their knowledge in robotics while receiving recognition on their transcript for this pursuit. The minor is not "owned" by any one department, but rather it is managed by the LCSR itself. Any student from any department within the university can work toward the minor.

Robotics is fundamentally integrative and multidisciplinary. Therefore, any candidate for the robotics minor must cover a set of core skills that cut across these disciplines, as well as obtain advanced supplementary skills.

Core skills include the following:

- Robot kinematics and dynamics (R)
- Systems theory, signal processing, control (S)
- Computation and sensing (C)

Supplementary advanced skills may be obtained in the following areas:

- Specialized applications, such as space, medicine, underwater, or haptics
- Advanced kinematics and dynamics
- Advanced systems theory
- Advanced computation, such as AI, machine learning, motion planning
- Advanced sensing such as computer vision

The full minor course listing (provided on the robotics minor website) specifies which courses fill these requirements. Note that ALL core areas must be filled, but that ANY advanced/supplementary courses can be chosen from the list. This allows students to strike a balance between breadth and depth.

Requirements

An undergraduate qualifies for the minor provided he or she has taken at least 18 credits (at the 300 level or above, with a C- or above) from an approved list of courses (provided on the robotics minor website), with the following requirements and restrictions:

- Between 2 and 4 courses chosen to cover the three core skills (see table below).
- At least 2 courses chosen from advanced supplementary skills (see table).
- At least 3 credits of the 18 must be a laboratory course (at least 15 hours of laboratory time that includes working with physical hardware and/or real data);
- At most 3 credits of the 18 can be an independent research or individual study with a faculty member on the list of approved faculty advisors;
- At least 2 courses must be primarily listed in a department other than the student's home department (it is acceptable if such a course is cross-listed in the student's home department).

Please visit the robotics minor website at **https://lcsr.jhu.edu/Robotics_Minor** for more details including a check-out sheet, a list of available advisors, sample curricula and the full minor course listing.

Short Courses

The Department of Computer Science offers 1-credit short courses covering a variety of topics in computer science and engineering. The purpose of the short courses is to expose students to topics of current interest in the field of computer science and engineering. Short courses are taught not only by faculty and graduate students in the Department of Computer Science and visiting faculty from other universities, but by individuals from local government or industry who have demonstrable expertise in a given area and are practicing the application of computer science theory and concepts.

Students should be aware that short course offerings are likely to change from year to year, depending on instructor commitments: there is no guarantee that the same course will be available at a later time. Students interested in getting details about a particular short course can contact the instructor through the departmental office. No more than 3 credits of short courses may be applied toward the computer science course credit requirement for the B.S. or the B.A. degrees.

Double Majors with Computer Science

It is possible for students to pursue a double major program in which one of the majors is computer science. The computer science requirements are flexible enough to allow for combination with most majors in the Whiting School of Engineering and the Krieger School of Arts and Sciences. Whether computer science is your primary or secondary major, you will be assigned a faculty advisor in the department. In order to declare a first or second major in computer science, students should see the Director of Undergraduate Studies before the start of senior year. Those students must also inform the Office of Academic Affairs of the Whiting School of Engineering and the Registrar of their double major status. Subject to restrictions set by the department offering a second major, students whose primary major is in the Whiting School may use courses to satisfy both the requirements of the student's primary major and those of a double major.

Concurrent Bachelor's/Master's Program

As early as the end of their sophomore year, qualified students may apply for admission to a concurrent 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 concurrent 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 600.3xx course as an undergraduate and the corresponding 600.4xx course for the MSE. Upon admission to the program students will be assigned 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 below, or ask in the departmental office for the document that lists those requirements.

Graduate Programs

Every graduate student in the Department of Computer Science must follow a program approved by a faculty advisor in the department. The advisor assigned to a student may change, subject to the acceptance of the new advisor.

Requirements for the M.S.E. Degree

The master of science in engineering (M.S.E.) is a full- or part-time day program offered by the Department of Computer Science. Most students complete the program in three full-time semesters. Two consecutive semesters of residence as a full-time graduate student are required. Those interested in part-time evening study should refer to the Engineering Programs for Professionals at **www.epp.jhu.edu**.

Entering students are expected to have completed a program of study equivalent to that required by the B.S. in computer science. Applicants from other disciplines are required to have course work (or equivalent experience) in intermediate programming (C++ and Java), data structures, and automata theory. Upon admission to the master of science in engineering program, a student is assigned a graduate advisor from the Department of Computer Science who must approve the courses to be applied to the M.S.E. degree.

The Department of Computer Science classifies its courses into three sub-areas: Analysis, Applications, and Systems. All M.S.E. candidates must complete at least two graduate courses (6 credit hours, 400-level and above) from each of these three areas. Each upper-level course description in this catalog includes its area for reference. A course in multiple areas may only be counted toward one requirement. A current listing of courses grouped by area is provided on the departmental website. While this listing includes a few highly relevant courses outside the Department of Computer Science, only one such course may be applied toward the area requirements. M.S.E. students must also complete an additional two elective graduate courses (chosen from any CS area or from closely related departments such as Electrical and Computer Engineering, Cognitive Science, Mathematics, or Applied Mathematics and Statistics) for a total of eight graduate courses. *The course work program must be approved by the student's faculty advisor and the department.*

In addition to the eight courses, a student must elect one of the following options in order to fulfill the degree requirements:

- Two additional graduate courses in Computer Science, approved as above.
- A supervised research project including an approved project report that will be made publicly available.
- An original, faculty-approved master's essay, submitted for binding to the Milton S. Eisenhower Library.

By satisfying the Ph.D. qualifying course requirements and the first qualifying project, a student will also satisfy the M.S.E. degree requirements (unless more than two course requirements have been satisfied using courses transferred from other institutions). Please refer to the Ph.D. program information for details.

All M.S.E. degree candidates are encouraged to regularly attend the department seminars. You may enroll in Computer Science Seminar 600.601-602; however, these courses may not be counted toward the degree course requirements.

Course Requirement Details

- All courses counted toward the M.S.E. degree requirement must be 400-level or above. At most, two courses with grades less than B- may be counted toward the course work requirements. No courses with grades less than C- may be counted.
- The overall grade point average of the courses counted toward the course work requirements must be a 3.0 or higher (B average).
- At most, two independent study courses (including 600.491-492 Computer Science Workshop I and II) can be counted toward the course requirements.
- Other than independent study courses and 600.464/664, no courses with grades of S can be counted toward the course work requirement. Courses with grades of S will not be included in the grade point average calculation.
- One of the courses required for the M.S.E. degree, but only one, can be replaced by 3 credits from comparable short courses.
- A majority of the courses counted toward the degree must be taught in the Department of Computer Science.
- At most, two courses can be transferred from graduate programs of other institutions to be counted toward the degree requirements. Such transfer courses must be approved by the student's faculty advisor and the department. It is the obligation of the student to provide all necessary data to the Department of Computer Science regarding the course(s) for which transfer credit is being requested.

- A grade of D or F results in probation; a second D or F is cause for being dropped from the program.

Tuition Support

Students studying for an M.S.E. degree may be eligible for partial tuition support after their first two semesters in the Department of Computer Science. There are also course assistant positions (paid by the hour) available for qualified students who are seeking financial support. Those interested must apply at the start of each semester for specific courses in need.

Requirements for the Ph.D. Degree

The goal of the Doctor of Philosophy (Ph.D.) program in the Department of Computer Science is to prepare first-rate scholars in the analysis, systems, and applications areas of computer science. Successful graduates may assume significant positions in academia, research institutes, industry, or government laboratories.

Applications for admission to the Ph.D. program in Computer Science are reviewed by a faculty committee. Although the specific criteria are not rigid, all students admitted will exhibit exceptional intellectual achievements and promise. Applicants must submit letters of recommendation, GRE scores, and (for foreign applicants) TOEFL scores.

In keeping with Hopkins' traditions, program requirements are flexible, as described below.

University Residency

Two consecutive semesters of residence as a full-time graduate student are required.

Seminar Attendance

All Ph.D. degree candidates are required to enroll and maintain satisfactory attendance in Computer Science Seminar 600.601-602 each semester for the duration of their enrollment in the program. Although seminar attendance is required, the seminar may not be counted toward the qualifying course requirement

Qualifying Course Requirements

The Department of Computer Science classifies its courses into three research areas: analysis, applications, and systems. All Ph.D. candidates must complete at least two graduate courses (400-level and above) from each of these three areas. Each upper-level course description in this catalog includes its area for reference. A courses in multiple areas may only be counted toward one requirement. A current listing of courses grouped by area is provided on the departmental website. While this listing includes a few highly relevant courses outside the Department of Computer Science, only one such course may be applied toward the area requirements. Ph.D. students must also complete an additional two elective graduate courses (chosen from any CS area or from closely related departments such as Electrical and Computer Engineering, Cognitive Science, Mathematics, or Applied Mathematics and Statistics) for a total of eight graduate courses. *The course work program must be approved by the student's faculty advisor and the department.* The overall grade point average for these eight courses must be at least equivalent to a B+. No course with a grade of less than C- may be counted toward this Ph.D. qualifying course requirement. Other than independent study courses and 600.464/664, no courses with grades of P can be counted toward the course work requirement. Courses with grades of P will not be included in the grade point average calculation. One of the courses required for the degree, but only one, may be replaced by 3 credits from comparable short courses. With approval of the student's faculty advisor, up to two courses can be transferred from graduate programs of other institutions; more than two such courses can be transferred with approval of the department. It is the obligation of the student to provide all necessary data to the Department of Computer Science regarding the course(s) for which transfer credit is being requested. Students are expected to complete the course requirements by the end of their second year as a Ph.D. candidate.

Qualifying Project Requirements

A Ph.D. student must complete two projects, each under the supervision and with the written agreement of a different faculty member in the Department of Computer Science. Upon conclusion of each project, the student must write a "Project Report" describing the project in detail. This report will be a public document and will be kept on file in the department office. The supervising faculty member must approve the project report. Departmental approval of a given project will be determined collectively by the faculty of the Department of Computer Science following the spring semester of each academic year. A factor taken into account in the departmental review of a project is the stated willingness of each supervising faculty member to enter the initial stages of a Ph.D. research advisor/advisee relationship with the student. Students are expected to complete the qualifying projects by the end of their second year as a Ph.D. candidate.

Upon completion of the Ph.D. qualifying course requirements and the first qualifying project, students are ordinarily eligible to receive a master of science in engineering degree. The degree will be awarded upon student request.

Graduate Board Oral Examination (GBO)
This examination is a university requirement, to be taken within one year of passing the Ph.D. qualifying requirements. The oral exam is administered by a panel consisting of the research sponsor, two faculty members from the Department of Computer Science, and two from outside the department. The exam seeks to establish the student's readiness to conduct original research in the area of his or her "Preliminary Research Proposal," which should be distributed to the examiners in advance and presented by the student at the start of the exam.

Part-Time Ph.D.
Two consecutive semesters of residence as a full-time graduate student are required by the university. Part-time students must pass both the Ph.D. qualifying requirements and the Graduate Board oral exam within four years of being admitted to the program. Attempting to obtain a Ph.D. is a major commitment and involves close coordination with a faculty advisor in the department. Part-time students must be able to establish and maintain these close links.

Departmental Seminar
Ph.D. students must give an official departmental seminar on their research area. This is to be done after the GBO and prior to the dissertation defense, or as part of the dissertation defense.

Dissertation and Defense
Ph.D. students must write a dissertation consisting of original research in their chosen area. They must deliver a public presentation of the dissertation before a dissertation committee consisting of the faculty advisor, a second faculty member in the Department of Computer Science (who must have a primary tenure-track appointment in the Department if the advisor does not), and one or more other members with Ph.D. degrees. In conformity with University requirements, the members of the dissertation committee must submit a referee's letter to the Graduate Board recommending that the dissertation be accepted. Completed dissertations will be bound and submitted to the Milton S. Eisenhower Library.

Student Progress Review
Ph.D. students will be reviewed annually by the department faculty and notified by their advisors as to their standing in the program. Beginning in the third year of graduate study, this annual review is conducted primarily by the dissertation committee. The committee may establish milestones such as a written thesis proposal. While the membership of the committee may change, in general it should be chosen by the student, in consultation with the advisor and subject to the consent of the committee members.

Financial Aid

Financial aid is available for candidates of high promise. Fellowships provide a student with a stipend plus tuition. Teaching assistantships normally consist of tuition plus a stipend commensurate with the teaching or grading duties assigned. Research assistantships are available on sponsored research projects directed by members of the faculty. Students determined to have significant deficiency in spoken English may be required to take one or more semesters of English as a Second Language in order to qualify for employment as a teaching or research assistant.

Undergraduate Courses

600.102 (E) CS Foundations
This course is an introduction to computer science for majors and non-majors. Students are exposed to the discipline through vignettes of logic and algebra, computer systems and networks, algorithms, programming languages, computation theory, and selected applications. CS majors can only take this course in their first year of CS coursework. Prerequisite: 600.101 or equivalent knowledge. [General] (last offered Spring 2008)
Fröhlich 4 credits

600.103 (E) Fundamentals of Practical Computing
Intended audience: students majoring in science, engineering or medicine. This course will provide a sampling of the theory behind and practical use of a broad spectrum of computational tools and technologies. We will start with Scratch (a programming language for kids) and show how many of the same concepts show up in Web Programming (HTML & Javascript). There will be a taste of algorithms, databases (SQL), Unix, statistics packages (R), data mining and visualization tools (graphviz), natural language processing, web search, interpreted languages (Python & LISP), compiled languages (C), and more. Students should come away with a few tools and concepts that will prove useful in their major, as well as the confidence that they can search the web to find what they need, when they need it, just-in-time.
Church 3 credits

600.104 (HE) Computer Ethics—Theory and Practice

Students will examine a variety of issues regarding various 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 general issues related to various ethical frameworks and move to topics specifically related to computers. The topics will include privacy issues, computer crime, intellectual property law—specifically copyright and patent issues, globalization, and ethical responsibilities for computer science professionals. Short course.

Kosaraju 1 credit

600.105 M&Ms: Freshman Experience

This course is required for all freshman computer science majors. Transfers into the major and minors may enroll by permission only. Students will attend four 3-week blocks of meetings with different computer science professors, focused on a central theme. Active participation is required. Satisfactory/Unsatisfactory only. [General]

Selinski 1 credit fall

600.106 Pre-Programming: Algorithmic Thinking

This course is intended for novice programmers, to be taken before or in conjunction with 600.107 or 600.109. The purpose is to provide students with the abstraction and logical thinking tools necessary for writing computer programs. It will introduce students to fundamental concepts and algorithms common to many programming languages. Students will primarily do paper solutions. Short course. Satisfactory/Unsatisfactory only. [General]

Staff 1 credit

600.107 (E) Introduction to Programming in Java

This course introduces fundamental structured and object-oriented programming concepts and techniques, using Java, and is intended for all who plan to use computer programming in their studies and careers. Topics covered include variables, arithmetic operators, control structures, arrays, functions, recursion, dynamic memory allocation, files, class usage and class writing. Program design and testing are also covered, in addition to more advanced object-oriented concepts including inheritance and exceptions as time permits. First-time programmers are strongly advised to take 600.108 concurrently. Prerequisite: familiarity with computers. [General]

Selinski 3 credits fall, spring, summer

600.108 (E) Introduction to Programming Lab

This course is intended for novice programmers, and must be taken in conjunction with 600.107. The purpose of this course is to give first-time programmers extra hands-on practice with guided supervision. Students will work in pairs each week to develop working programs, with checkpoints for each development phase. Prerequisite: familiarity with computers. Corequisite: 600.107. Satisfactory/ Unsatisfactory only. [General]

Selinski 1 credit fall, spring

600.111 (E) Python Scripting

For non-majors, this is an introductory "learning by doing" course focused on the quick prototyping of computational solutions to problems from a variety of disciplines. After an introduction to the UNIX and IDLE environments we briefly cover the basics of programming in Python. We then spend the rest of the semester surveying a variety of powerful Python libraries, frameworks, and tools. We use these building blocks to create, for example, systems for image and sound processing, data analysis and visualization, event-based simulation, or database-driven web applications. There will be several sizeable team-based programming projects. [Note: This course may not be used for the CS major or minor requirements, except perhaps as a substitute for 600.107.] Prerequisite: none.

Froehlich 3 credits

600.120 (E) Intermediate Programming

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: 600.107 or AP CS or equivalent. [General]

Staff 4 credits fall, spring

600.133 (E) Embedded Systems Fundamentals: Programming the Hardware-Software Interface

This hands-on course will be a gentle introduction to the field of embedded computing; computer systems that interact with the real world, making possible cell phones, MP3 players, flash drives, Wii games, and many more. Students will be programming different projects in modern microcontrollers, witnessing the effects immediately. The course will address the hardware-software interface of computer systems, setting a good foundation to understand the physical layer of computer applications and networking. The main topics are computer architecture fundamentals, assembly language, interfacing peripherals, programming device drivers, working with sensors, and data acquisition. Each student will be provided with the material to perform experiments individually, as well as with learning handouts. The course will be intensive but fun.

Staff 2 credits intersession

600.145 (E) Introduction to Computer-Integrated Surgery

This course will give an introduction to the concepts and major elements of computer-integrated surgery (CIS) through clinical applications. Students will learn to ask questions and look for answers the way clinical engineers build and analyze CIS systems. Major topics will include medical imaging, image processing, surgical planning, surgical robotics, robot navigation, systems integration, and clinical validation. Optional visits to CIS laboratories

and clinical experiments will also be offered. No computer programming will be necessary to complete the assignments. Prerequisites: pre-calc required; knowledge of linear algebra helpful. Short course. [General]
Taylor 1 credit intersession

600.146 (E) Introduction to Medical Imaging
This intersession class will provide an introduction to the principles of medical imaging. X-ray, CT and ultrasound imaging will be covered. The course will offer an introduction to the principles, instrumentation and applications of each modality. The class will be a mixture of lectures, class discussions and imaging demos using medical imaging resources at the Computational Sciences and Engineering Building. Assignments will test theoretical knowledge and also practical applications. Basic Matlab knowledge and pre-calculus math are recommended. Note: Students should not expect an in depth analysis of medical imaging systems. This class is not intended as a substitute for Medical Imaging courses offered during fall and spring terms.
Staff 1 credit intersession

600.147 (E) Computer Applications in Radiation Therapy
The course consists of three parts: the physics of radiation therapy (one week), the computer in delivery systems—hardware (one week) and the computer in treatment planning systems—software (one week). The materials are generally high level topics in this area and they are not detailed mathematics or physics. It is for students to understand how important the role of software and hardware is in this particular field of medicine.
Staff 1 credit intersession

600.161 (E) Exploring Vision in the Real World
The course will focus on real world applications of computer vision and image processing, primarily in the areas of medicine and sports. The goal is to introduce students to computer vision concepts and explain how they are the building blocks for interesting and practical applications. One such example is the use of stereo vision to enhance micro surgery. Another example is the use of computer vision to create realistic 3D real-time video fly-through in sporting events, such as the NFL Super Bowl. The class will consist of lectures and class discussions. Students will be evaluated on weekly assignments and participation in discussions. This course is designed to introduce computer vision to interested students with or without a computer science background. An engineering background is recommended but not required. Matlab programming will be part of homeworks. Pre-calculus level math and some linear algebra is required.
Staff 1 credit intersession

600.202 (E) Introduction to Public Health and Biomedical Informatics
Information technology should radically change the practice of medicine, the research of health science, and the assurance of public health. In this course, we review the core technologies of informatics and how those technologies ought to be considered, used, and evaluated, using examples from Johns Hopkins, from developing countries, and from around the world. Topics covered include basic technology, data, information, knowledge, standards and interoperability, software engineering frameworks, electronic patient records, biosurveillance, and clinical research systems. This course should be of interest to those aiming toward the biosciences, computers, the information sciences, and cognate social sciences.
Lehmann 3 credits summer

600.211 (E) UNIX Systems Programming
This course covers a variety of topics in UNIX programming, including process control, signal handling, daemon processes, and interprocess communication. Participants must be familiar with using the UNIX environment and be fluent in the C programming language. Prerequisite: 600.120. [General]
Fröhlich 3 credits

600.226 (E,Q) Data Structures
This course covers the design and implementation of data structures including collections, sequences, trees, and graphs. Other topics include sorting, searching, and hashing. Course work involves both written homework and Java programming assignments. An overview of Java will be provided. Prerequisite: AP CS, 600.107 or equivalent. [General]
Staff 3 credits fall, spring

600.245 (E) Foundations of Computer Integrated Surgery
This course will give an introduction to the concepts and major elements of computer-integrated surgery (CIS) through clinical applications. Major topics will include medical imaging, image processing, surgical planning, surgical robotics, robot navigation, systems integration, and clinical validation. The class includes a human cadaver lab module to perform minimally invasive spine surgery with the use of novel technologies discussed in class. Grades will be calculated based on participation in class and three homework assignments. No computer programming will be necessary or required to complete the assignments, but bonus offered for demonstrating programming skills. Prerequsite: pre-calculus; recommended: linear algebra and vector calculus.
Kumar 3 credits summer

600.250 (E) User Interfaces and Mobile Applications
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 tech-

niques and tools. Prerequisite: 600.120 and 600.226. [General]
Selinski 3 credits

600.255 (E) Introduction to Video Game Design
A broad survey course in video game design (as opposed to mathematical game theory), covering artistic, technical, as well as sociological aspects of video games. Students will learn about the history of video games, archetypal game styles, computer graphics and programming, user interface and interaction design, graphical design, spatial and object design, character animation, basic game physics, plot and character development, as well as psychological and sociological impact of games. Students will design and implement an experimental video game in interdisciplinary teams of 3-4 students as part of a semester-long project. Prerequisite: sophomores and above, permission of instructor, technical students should have taken at least one (preferably two or more) programming-related courses; artistic students should have taken at least one (preferably two or more) multimedia-related courses; corequisite: 600.256.
Froehlich 3 credits

600.256 Introduction to Video Game Design Lab
A lab course in support of 600.255: Introduction to Video Game Design covering a variety of multi-media techniques and applications from image processing, through sound design, to 3D modeling and animation. See 600.255: Introduction to Video Game Design for details about enrolling. Corequisite: 600.255.
Froehlich 1 credit

600.271 (E,Q) Automata and Computation Theory
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. Students may not take both 600.271 and 600.471, unless one is for an undergrad degree and the other for grad. Prerequisite: none. [General]
Kosaraju 3 credits spring, summer

600.306 (E) Introduction to Speech
This course will introduce students to speech from an interdisciplinary perspective including computer science, electrical engineering, linguistics, and psychology. Topics such as pitch will be discussed from a variety of perspectives including signal processing (estimating fundamental frequency), perception, linguistics, and computational linguistics. Vowels will be described from multiple perspectives ranging from distinctive features in linguistics to formants in signal processing. Students will become familiar with a variety of topics ranging from spectrogram reading to using XML to program phones and Python (NLTK) to find interesting patterns in text corpora. To reach a diverse interdisciplinary audience, no background experience is required. Short course.
Church 1 credit

600.315 (E) Databases
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. Course work includes significant practical implementation experience. Prerequisite: 600.226. [Systems]
Yarowsky 3 credits fall

600.316 (E) Transaction Processing Systems
This course covers the design and implementation of transaction processing and database systems. Topics include transaction semantics, write-ahead logging, memory management, checkpoints, concurrency control, replication, restart recovery, and distributed commit protocols. The course employs examples of advanced database applications to develop this material. Examples include Internet databases, TP monitors, multidatabases, and federated databases. Course work includes a project. Prerequisites: 600.315/415 or equivalent, 600.120. [Systems] [Last offered 2009.]
Burns 3 credits

600.318 (E) Operating Systems
This course covers the 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. Prerequisites: 600.120, 600.226, and 600.333; 600.211 recommended. [Systems]
Staff 4 credits spring

600.319 (E) Storage Systems
Storage systems is one of the fastest growing and most interesting research areas in computer science. Storage systems often dominate the performance of computer systems as a whole. Also, they are responsible for the safe-keeping of an organization's most valuable assets—information! The course will cover the design and implementation of storage systems and the architecture and characteristics of the components on which storage systems are built. Topics will range from the device level up to distributed systems concepts. This will include disk drive hardware and firmware, file system and database structures, mirroring and RAID, disk array controllers, local storage interconnects, storage area networks, capacity planning and configuration, distributed file systems and network-attached storage, backup/restore and disaster recovery, and security for storage. Prerequisites: 600.226 and 600.333/433. [Systems] [Last offered 2008.]
Burns 3 credits

600.320 (E) Parallel Programming
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" (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 three weeks and each section will involve a programming project. The course is also suitable for second-year undergraduate CS majors and undergraduate and graduate students from other science and engineering disciplines that have prior programming experience. Prerequisites: 600.120 or equiv. [Systems]
Burns 3 credits

600.321 (E) Object-Oriented Software Engineering
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. Prerequisites: 600.226 and 600.120. [Systems or Applications]
Smith 3 credits fall

600.324 (E) Network Security
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. Prerequisites: 600.226, 600.344/444 or permission; 600.120 (or equivalent) recommended. [Systems]
Staff 3 credits

600.325 (E) Declarative Methods
Suppose you could simply write down a description of your problem, and let the computer figure out how to solve it. What notation could you use? What strategy should the computer then use? In this survey class, you'll learn to recognize when your problem is an instance of satisfiability, constraint programming, logic programming, dynamic programming, or mathematical programming (e.g., integer linear programming). For each of these related paradigms, you'll learn to reformulate hard problems in the required notation and apply off-the-shelf software that can solve any problem in that notation—including NP-complete problems and many of the problems you'll see in other courses and in the real world. You'll also gain some understanding of the general-purpose algorithms that power the software. Prerequisites: 600.226, Calc II. [Analysis]
Eisner 3 credits spring

600.328 (E) Compilers and Interpreters
Introduction to compiler design, including lexical analysis, parsing, syntax-directed translation, symbol tables, run-time environments, and code generation and optimization. Students are required to write a compiler as a course project. Prerequisite: 600.120 and 600.226. [Systems]
Fröhlich 3 credits

600.333 (E) Computer System Fundamentals
CSF addresses the design and performance of the principal operational components of a reduced-instruction-set computing system (RISC) which supports the efficient execution of widely used instruction sets. Arithmetic and logic units, memory hierarchy designs, state-machine controllers, and other related hardware and firmware components are studied, and the qualities of their combined processing capabilities are assessed by means of execution times associated with a range of benchmark programs. Assembly language programming projects, homework problems, and exams are employed to assess a student's fundamental understanding of the tradeoffs resulting from an assortment of variations in digital system design decisions that ultimately characterize the performance of the computing system architecture that is developed. Prerequisite: 600.107 or equiv. [Systems]
Masson 3 credits fall, summer

600.334 (E) Laboratory for Computer System Fundamentals
This course is a hands-on laboratory supplement to computer system fundamentals (600.333). Corequisite: 600.333.
Masson 1 credit fall

600.335 (E) Artificial Intelligence
Artificial Intelligence (AI) is introduced by studying automated reasoning, automatic problem solvers and planners, knowledge representation mechanisms, game playing, machine learning, and statistical pattern recognition. The class is recommended for all scientists and engineers with a genuine curiosity about the fundamental obstacles to getting machines to perform tasks such as deduction, learning, planning, and navigation. Strong programming skills and a good grasp of the English language are expected; students will be asked to complete both programming assignments and writing assignments. The course will include a brief introduction to scientific writing and experimental design, including assignments to apply these concepts. Prerequisite: 600.226, 550.171; linear algebra, prob/stat recommended. [Applications]
Staff 3 credits spring

600.336 (E) Algorithms for Sensor-Based Robotics
This is an introductory course presenting a series of algorithms related to the representation and use of geometric models acquired from sensor data. Course topics include: basic sensing and estimation techniques, geometric model representations, and motion planning algorithms. The course will also discuss applications in diverse areas such as mobile systems, robot manipulation, and medicine. Prerequisite: 600.226, calculus, prob/stat. [Analysis]
Hager 3 credits

600.337 (E) Distributed Systems
This course teaches how to design and implement protocols that enable processes to exchange information, cooperate, and coordinate efficiently in a consistent manner over a computer network. Topics include communication protocols, group communication, distributed databases, distributed operating systems, and security. The course gives hands-on experience as well as some theoretical background. Prerequisites: 600.120 and 600.226. [Systems]
Amir 3 credits

600.341 (E, Q) Basics of Applied Cryptography
This course is an introduction to algorithms, cryptography and network security, meant to give students a good foundation for upper-level courses in the area. Students will learn how to implement a simple cryptographic library in C. Prerequisites: 600.120 and 600.226. [Analysis]
Ateniese 3 credits

600.344 (E) Computer Network Fundamentals
This course considers intersystem communication issues. Topics include layered network architectures; the OSI model; bandwidth, data rates, modems, multiplexing, error detection/correction; switching; queuing models, circuit switching, packet switching; performance analysis of protocols, local area networks; and congestion control. Prerequisite: 600.333 or general knowledge of computer architecture. [Systems]
Terzis 3 credits spring

600.355 (E) Video Game Design Project
An intensive capstone design project experience in video game development. Students will work in groups of 4-8 on developing a complete video game of publishable quality. Teams will (hopefully) include programmers, visual artists, composers, and writers. Students will be mentored by experts from industry and academia. Aside from the project itself, project management and communication skills will be emphasized. Enrollment is limited to ensure parity between the various disciplines. Prerequisite: 600.255/256 or permission; junior or senior standing recommended. [General]
Fröhlich 3 credits

600.357 (E,Q) Computer Graphics
This course introduces computer graphics techniques and applications, including image processing, rendering, modeling and animation. Prerequisites: 600.120, 600.226, and linear algebra; or permission of instructor. [Applications]
Kazhdan 3 credits spring

600.361 (E,Q) Computer Vision
This course gives an overview of fundamental methods in computer vision from a computational perspective. Methods include computation of 3-D geometric constraints from binocular stereo, motion, texture, shape-from-shading, and photometric stereo. Edge detection and color perception are studied as well. Elements of machine vision and biological vision are also included. Prerequisites: 600.226. [Applications]
Hager 3 credits fall

600.363 (E,Q) Introduction to Algorithms
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. Prerequisite: 600.226. [Analysis]
Braverman 3 credits spring

600.392 (E) Senior Design Project
This course will give senior CS majors an intensive capstone design project experience. Students will work in groups with real world customers to develop a working system. Project design, management and communication skills will be emphasized. Software development methodologies may also be presented. Prerequisites: 600.120, 600.226; 600.321 recommended. [General]
Fröhlich 3 credits

Advanced Undergraduate/ Graduate Courses

600.402 (E) Medical Informatics
Key decision makers in government and industry and across the world believe that health information technology is crucial to improving health and safety and cutting costs, and are investing billions of dollars over the next few years to test that belief. In this course, you will learn to understand this new context and to figure out what role you might play in it. Prerequisite: none. Short course.
Lehmann 1 credit

600.405 (E, Q) Applications of Probabilistic Graphical Models in Language and Speech Processing
Probabilistic graphical models (PGMs) combine ideas from statistics and computer science into a unifying framework for modeling complex real-world phenomena. PGMs are now widespread in language and speech processing. PGMs are well suited to handle the inherent challenges of linguistic problems: complex and structured relationships, a large number of relevant attributes, and large volumes of data. This short course will provide students with advanced training in several specific applications of graphical models that are important in natural language processing. After reviewing the essentials of directed and undirected graphical models, we will discuss complex CRFs, approximate inference including variational and MCMC methods, Bayesian models and non-

parametric Bayesian models including Chinese Restaurant Processes. Students will also gain practical experience by solving problems using existing PGM software. Prerequisite: 600.465. Short course.
Staff 1 credit

600.406 (E) Developing Photo and Video Applications for Online Social Networks
How many hours do you spend on facebook a day? This experimental course will teach you how to create and launch web 2.0 applications. The class provides an introduction to the field of computer vision, giving you the tools to detect and track objects in the environment. Class topics include social network interfaces—primarily facebook application interface (API), image processing, face detection, virtual environment and rendering methods. Students will work in small teams to conceptualize, develop, distribute, and market new applications to facebook users. Course is appropriate for students interested in computer vision, entrepreneurship, or human-computer interaction. Prerequisite: 600.107 and 600.120. Short course.
Staff 2 credits

600.407 (E) General Purpose Computation on the GPU
Programmable graphics hardware not only provides a way to perform advanced real-time 3D rendering, but also a platform for highly parallel numerical computing. Over the past 5 years, the General Purpose Graphics Processor Unit (GPGPU) community has grown around performing non-graphics computations using the limited instruction set and framework of the graphics pipeline. This short course, which meets one hour per week from the Spring semester, will introduce students to GPGPU computing using NVIDIA's CUDA platform. Prerequisites: 600.120 and 600.333/433; computer graphics and linear algebra recommended. Short course.
Staff 1 credit

600.409 (E) Digital Preservation
This course explores how digital information may be stored, maintained, and retrieved over decades or centuries. It examines both the technical and social aspects of preservation, drawing material from both Computer Science and the Digital Library community. Lecture topics will include architectures for long-term archival, data provenance, information representation, metadata semantics, replica maintenance, authenticity and privacy, and business models for sustainable archives. Students will define and execute a research project investigating a hot unsolved problem related to data preservation. The course is suitable for upper-level undergraduates and graduate students from all disciplines that have had an introductory programming course. Prerequisites: 600.107 or equivalent. Short course.
Staff 1 credit

600.412 (E) Security and Privacy in Cloud Computing
This course focuses on the security and privacy issues in Cloud Computing systems. While the cloud computing paradigm gains more popularity, there are many issues related to confidentiality, integrity, and availability of data and computations involving a cloud. In this course, we examine cloud computing models, look into the threat model and security issues related to data and computation outsourcing, and explore practical applications of secure cloud computing. Short course.
Hasan 1 credit

600.415 (E) Databases
Similar material as 600.315, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.315 or 600.415, but not both. Prerequisite: 600.226. [Systems]
Yarowsky 3 credits fall

600.416 (E) Transaction Processing Systems
Similar material as 600.316, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.316 or 600.416, but not both. Prerequisite: 600.315/415 or equivalent, 600.120. [Systems] [Last offered 2009.]
Burns 3 credits

600.417 (E) Data Stream Processing
Data stream processing has emerged as a model for building computing applications that face tremendous volumes of dynamically changing data, and are required to process such data in a timely fashion. Examples include a variety of web-driven applications, such as web advertising based on Facebook and Twitter status streams, and more generally, monitoring and analysis applications including algorithmic trading on stock ticks and order books, network monitoring for denial of service attacks, and location-based applications working with GPS data streams. This course will study data stream processing from a data management and algorithms perspective. Students will be introduced to the fundamentals of data stream processing systems and architectures, incremental (windowed) stream processing languages, and stream algorithms that embody the principle of "you only get one look" when having to continually deal with data arriving at high rates. This course will provide students with significant implementation experience, in the spirit of a practicum. Students will proceed through a series of homework projects to build a data stream processor from scratch, and will use the resulting stream engine along with stream mining algorithms to analyze Twitter feeds. This course is aimed at upper-level undergraduates with prior programming experience. Graduate students should consider taking 600.617 instead. Students may receive credit for 600.417 or 600.617, but not both. Prerequisites: 600.120, 600.226, and 600.315/415. [Systems]
Ahmad 3 credits

600.418 (E) Operating Systems
Similar material as 600.318, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.318 or 600.418, but not both. Prerequisites: 600.120, 600.211 (or equivalent C experience), 600.226, 600.333. [Systems]
Staff 3 credits spring

600.419 (E) Storage Systems
Similar material as 600.319, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.319 or 600.419, but not both. Prerequisites: 600.226 and 600.333/433. [Systems] [Last offered 2008.]
Burns 3 credits

600.420 (E) Parallel Programming
Similar material as 600.320, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.320 or 600.420, but not both. Prerequisites: 600.120 or equiv. [Systems]
Burns 3 credits

600.421 (E) Object-Oriented Software Engineering
Similar material as 600.321, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.321 or 600.421, but not both. Prerequisites: 600.226 and 600.120. [Systems or Applications]
Smith 3 credits fall

600.424 (E) Network Security
Similar material as 600.324, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.324 or 600.424, but not both. Prerequisites: 600.226, 600.344/444 or permission; 600.120 (or equivalent) recommended. [Systems]
Staff 3 credits fall

600.425 (E) Declarative Methods
Similar material as 600.325, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.325 or 600.425, but not both. Prerequisites: 600.226, Calc II. [Analysis]
Eisner 3 credits spring

600.426 (E) Principles of Programming Languages
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. Prerequisites: 600.226. Freshmen/sophomores by permission only. [Analysis]
Smith 3 credits spring

600.427 (E) Data Organization: Storage and External Memory Systems
This course will examine the complex relationship between computer architectures and software systems that store, organize, and access data. Storage systems have always co-evolved with technology. But, today's computing landscape places unique demands on next generation storage systems. Technology drivers include: new storage devices, such as solid-state drives and phase-change memory, cloud computing, virtualization, and modern multi-core and manycore processors with steep hierarchies of shared caches. The course will provide an overview of modern storage systems, including parallel file systems, key/value stores, scan engines, in-memory databases, archival storage, and content-based storage. It will cover the techniques used to organize storage in these systems, such as indexes, replication and coding, spatial trees, and space-filling curves. The course will also explore external memory data structures and algorithms that provide a framework for analyzing and costing storage designs. Prerequisites: 600.226, 600.315/415, and 600.333/433 or permission of instructor. [Systems]
Burns 3 credits

600.428 (E) Compilers and Interpreters
Similar material as 600.328, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.328 or 600.428, but not both. Prerequisite: 600.120 and 600.226. [Systems]
Fröhlich 3 credits

600.430 (HQ) Ontologies and Knowledge Representation
Knowledge representation (KR) deals with the possible structures by which the content of what is known can be formally represented in such a way that queries can be posed and inferences drawn. Ontology concerns the hierarchical classification of entities from given domains of knowledge together with the relations between various classes or subclasses. We begin with KR, examining the standard variety of frameworks developed or implemented over the last twenty years, including 1st-order logic and automated theorem proving, networks, frames, and description logics. Then we move on to a study of the problems inherent in ontology development and examine some of the currently prevalent environments, including Universal Modeling Language, OWL and Protege', RDFS and semantic web applications. [Analysis]
Rynasiewicz 3 credits

600.433 (E) Computer Systems
Similar material as 600.333, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.333 or 600.433, but not both. Prerequisite: 600.107 or 600.109. [Systems]
Masson 3 credits fall/summer

600.435 (E) Artificial Intelligence
Similar material as 600.335, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.335 or 600.435, but not both. Prerequisite: 600.226, 550.171; linear algebra, probability/statistics recommended. [Applications]
Staff 3 credits spring

600.436 (E) Algorithms for Sensor-Based Robotics
Similar material as 600.336, covered in more depth. Intended for upper-level undergraduates and graduate

students. Students may receive credit for 600.336 or 600.436, but not both. Prerequisite: 600.226, calculus, probability/statistics. [Analysis]
Hager 3 credits

600.437 (E) Distributed Systems
Similar material as 600.337, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.337 or 600.437, but not both. Prerequisites: 600.120 and 600.226. [Systems]
Amir 3 credits fall

600.438 (E) Advanced Operating Systems
This course is for people who wish to explore operating systems in greater detail in a hands-on fashion. Student partners (pairs) implement a UNIX-inspired thread library and kernel. Prerequisite: 600.318/418, 600.439 or perm. [Systems]
Staff 4 credits

600.442 (E,Q) Modern Cryptography
This course focuses on cryptographic algorithms, formal definitions, hardness assumptions, and proofs of security. Topics include number-theoretic problems, pseudo-randomness, block and stream ciphers, public-key cryptography, message authentication codes, and digital signatures. Prerequisites: 600.271 and 550.171 or equiv, 600.226 and a 300-level or above systems course. [Analysis]
Ateniese 3 credits

600.443 (E) Security and Privacy in Computing
Lecture topics will include computer security, network security, basic cryptography, system design methodology, and privacy. There will be a heavy workload, 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. Prerequisite: a basic course in operating systems and networking, or permission of instructor. [Applications]
Rubin 3 credits spring

600.444 (E) Computer Networks
Similar material as 600.344, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.344 or 600.444, but not both. [Systems]
Terzis 3 credits spring

600.445 (E) Computer-Integrated Surgery I
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. An optional term project may be undertaken under supervision of the instructor and clinician end users. Although this course is primarily intended for graduate students and advanced undergraduate students interested in doing research in this area, it may also be of interest to medical or qualified premedical students wanting to obtain a broader background in this emerging field. Prerequisites: 600.120, 600.226 (or equivalent programming experience) and linear algebra or permission of instructor; recommended: 600.457, 600.461, image processing. [Applications]
Taylor 4 credits fall

600.446 (E) Computer-Integrated Surgery II
This weekly lecture/seminar course addresses similar material to 600.445, 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 600.445, 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 600.452. Students may also take this course as 600.646. The only difference between 600.446 and 600.646 is the level of project undertaken. Typically, 600.646 projects require a greater degree of mathematical, image processing, or modeling background. Prospective students should consult with the instructor as to which course number is appropriate. Prerequisite: 600.445 or permission of instructor. [Applications]
Taylor 3 credits spring

600.450 (E) Network Embedded Systems and Sensor Networks
This course is an introduction to fundamental concepts of networked embedded systems and wireless sensor networks. It is intended for juniors, seniors and first-year graduate students in computer science and other engineering majors with the prerequisite background. Covered topics include embedded systems programming concepts, low power and power aware design, radio technologies, communication protocols for ubiquitous computing systems, and some of the mathematical foundation of sensor behavior. Laboratory work consists of a set of programming assignments that consider a set of the issues described in class. Prerequisites: 600.226, 600.120, and 600.344. [Systems]
Terzis 3 credits fall

600.451 (E) Performance of Computer-Communication Networks & Protocols
This is an advanced course in networks and protocols that examines the performance evaluation, design, and management of networks, including wireless networks. This course may have additional newer topics such as network calculus and randomized algorithms as well as other algorithms for networking. The course uses analytical and simulation methods to evaluate, design and manage net-

works and protocols. Topics include introduction to and application of queuing theory, queueing networks, introduction to and application of graph theory, optimization techniques for routing and flow control; introduction to and application of simulation methods; performance of multiple access, TCP/IP, Wireless Cellular, Ad hoc and Sensor Networks; design of backbone and access networks. Prerequisites: 600.344/444 & 550.310. [Analysis]
Mishra 3 credits

600.452 (E) Computer-Integrated Surgery Seminar
Essentially, 600.452 is identical to 600.446/646 without the term project. Students may receive credit for only one of 600.446/452/646. Prerequisite: 600.445 or permission of instructor. [Applications]
Taylor 1 credit spring

600.454 (E) Practical Cryptographic Systems
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. [Co-listed with 650.445] [Systems]
Green 3 credits spring

600.457 (E,Q) Computer Graphics
Similar material as 600.357, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.357 or 600.457, but not both. Prerequisites: 600.120, 600.226, linear algebra; or permission of instructor.[Applications
Kazhdan 3 credits spring

600.461 (E,Q) Computer Vision
Similar material as 600.361, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.361 or 600.461, but not both. Prerequisites: 600.226. [Applications]
Hager 3 credits fall

600.463 (E,Q) Algorithms I
Similar material as 600.363, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for 600.363 or 600.463, but not both. Prerequisite: 600.226. [Analysis]
Braverman 3 credits spring

600.464 (E,Q) Randomized Algorithms
The course concentrates on the design and analysis of randomized algorithms. Problems from graph theory, computational geometry and information routing in networks will be treated. Some knowledge of probability theory and deterministic algorithmic techniques is helpful. Prerequisite: 600.363 or 600.463. [Analysis]
Kosaraju 3 credits fall

600.465 (E) Introduction to Natural Language Processing
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. Prerequisite: 600.226; previous exposure to probability or linguistics may be helpful. [Applications]
Eisner 3 credits fall

600.466 (E) Information Retrieval and Web Agents
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. Prerequisite: 600.226. [Applications]
Yarowsky 3 credits spring

600.467 (E) Wireless Networks
This course covers the basics of mobile communication and wireless networking for computer science majors by keeping a balance between communication and networking topics. In this course the students will be exposed to wireless transmission fundamentals (path loss, shadowing, modulation, coding and channel models), wireless cellular networks (cellular concept, channel reuse, capacity limits, and cellular systems such as GSM, GPRS and UMTS), and learn about mobile network and transport layers, medium access control protocols, wireless local area networks (IEEE 802.11), wireless mesh networks (IEEE 802.16), and emerging dynamic spectrum access networks based on cognitive radios. Prerequisites: 600.344/444 recommended. [Systems]
Mishra 3 credits

600.471 (E,Q) Theory of Computation
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 600.271 and 600.471, unless one is for an undergrad degree and the other for grad. Prerequisite: 550.171 or permission. [Analysis]
Hohenberger 3 credits

600.472 (E, Q) Theoretical Cryptography
The focus of this course is on the definitions and constructions of various cryptographic primitives and protocols, such as one-way functions, pseudo-random generators, digital signature schemes, encryption schemes, zero-knowledge and multiparty computation. We will study how to formulate definitions that capture desired security properties as well as techniques for designing and then proving that a construction realizes these properties. Students should be comfortable with the basics of number theory and proof writing. Prerequisite: 600.471 recommended. [Analysis]
Hohenberger 3 credits

600.475 (E) Machine Learning
This course takes an application driven approach to current topics in machine learning. The course covers supervised learning (classification/structured prediction/regression/ranking), unsupervised learning (dimensionality reduction, bayesian modeling, clustering) and semi-supervised learning. Additional topics may include reinforcement learning and learning theory. The course will also consider challenges resulting from learning applications, such as transfer learning, multi-task learning and large datasets. We will cover popular algorithms (naive Bayes, SVM, perceptron, HMM, winnow, LDA, k-means, maximum entropy) and will focus on how statistical learning algorithms are applied to real world applications. Students in the course will implement several learning algorithms and develop a learning system for a final project. Prerequisite: multi-variate calculus. [Applications]
Dredze 3 credits

600.478 (E) Visual Imaging in Surgery and Medicine
A survey course in visual imaging registration and fusion methods and its applications in surgery and medicine. Such applications are common in medical imaging including integration of CT, MRI, ultrasound, PET, and other sensing. However, compared to these sensing technologies visual imaging requires more efficient computation and stronger emphasis on contextual and temporal information. Key goals for such methods include multi-resolution, and multi-temporal registration and superresolution. A large body of work and practical applications using visual imaging exist in remote sensing, surveillance, and robot vision, but methods for surgical visualization are relatively rare and new. This course aims to provide background on devices, methods, and applications for visual imaging in medicine and surgery including recent work in the field. Students will design and implement registration methods based on data sets provided as part of a semester-long team project. Prerequisite: 600.226, 600.461; recommended: linear algebra, 600.445. [Applications]
Kumar 3 credits

600.488 (E) Foundations of Computational Biology and Bioinformatics II
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. Foundations of Computational Biology I is not a prerequisite. Prerequisites: math through linear algebra and differential equations, 580.221 or equiv., 600.226 or equiv. [Co-listed with 580.488.] [Analysis]
Karchin 4 credits

600.491-492 (E) Computer Science Workshop I, II
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: consent of faculty supervisor.
Staff 3 credits

600.498 Programming Contest Laboratory
This laboratory course meets for 2.5 hours/week. The intent is to develop the skills needed to achieve victory in programming contests. Prerequisite: permission of instructor. [General]
Smith 1 credit

600.503-504, 576, 597 Independent Study
Individual guided study for undergraduates, 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. May be taken fall (503), spring (504), intersession (576), summer (597).

600.507-508, 574, 595 Independent Research
Independent 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. May be taken fall (507), spring (508), intersession (574), summer (595).

600.509-510, 550, 599 Computer Science Internship
Individual work in the field with a learning component, supervised by a faculty member in the department. The program of study 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. S/U only, 1 credit only. May be taken fall (509), spring (510), intersession (550), summer (599).

600.519-520 (E) Senior Honors Thesis
For computer science majors only. 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 1 of senior year. They will also submit a first draft of their project report (thesis documentation) at that time. Faculty will meet to decide if the thesis will be accepted for honors. Prerequisite: 3.5 GPA in Computer Science after spring of the junior year and permission of faculty supervisor.
Staff 3 credits/semester (taken twice)

600.546 (E) Senior Thesis in Computer Integrated Surgery
The student will undertake a substantial independent research project in the area of computer-integrated surgery, under joint supervision of a WSE faculty advisor and a clinician or clinical researcher at the Johns Hopkins Medical School. This project will typically require background literature research, design, and execution of an experimental study or substantial implementation effort, and write-up of the results. The written reports will be published as reports by the CISST Engineering Research Center and may be used by the students as the basis for further academic publication. Because of the interdisciplinary, team oriented nature of much CIS research, students may work in small groups or with other members of the advisor's research group. Students will be expected to establish a research plan and schedule and may be required by their advisor to provide interim documentation and meet interim deadlines, as appropriate. This requirement will be especially pertinent for two-semester projects. Prerequisite: 600.445 or permission required. [Applications]
Taylor 3 credits/semester (may be taken twice)

Graduate Courses

600.601-602 Computer Science Seminars
This course is offered satisfactory/unsatisfactory each semester. A grade of satisfactory can be attained by attending a minimum of the smallest integer greater than or equal to 2N/3 seminars in the Department of Computer Science, where N is the total number of seminars which are presented between and including the first and last class days of the semester and which are officially announced at least one week in advance. An email message and/or display of a poster outside the department office describing the seminar will constitute its official announcement. This course is required for all full-time graduate students in Computer Science. [General]
Staff fall, spring

600.603-604 Current Topics in Language and Speech Processing
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, which 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. (Last offered Fall 2010—students should take 520.701/702 instead.) [General]
Khudanpur fall, spring

600.615 Big Data, Small Languages, Scalable Systems
This class will study domain-specific data management tools, focusing on extremely scalable system design based on the domain's semantic and structural properties. We will study a variety of data models including stream, graph, array, and probabilistic data, and their processing on modern architectures such as column- and key-value stores, stream, and XQuery engines. Further topics include the use of novel hardware such as solid state disks, phase change memory, GPUs, and FPGAs. The class includes a semester long group project to develop a query processor for an application of the group's choice (e.g. on system log, finance, web, sensor, speech data). Prerequisite: 600.315/415 or equivalent. [Systems]
Ahmad 3 hours

600.619 Advanced Storage and Transaction Processing Systems
In this course, we will examine advanced research topics in storage systems, file systems, transaction processing, and network data management. The readings are taken from the current research literature and articles of historical significance. This course is intended for graduate students interested in conducting research on or related to these topics and for students who face management, availability or performance issues with data in their own research. Students will conduct a semester long research project and present their results to the class. In addition to the scheduled meetings, students will have weekly one-on-one meetings with the professor. Prerequisite: 600.419 or permission of instructor. [Systems]
Burns 3 hours

600.620 External Memory Data Structures and Algorithms
This course will cover data structures and algorithms for managing external memory with applications to file systems, databases, parallel architectures, and high-performance computing. Topics will include cost models for external memory, elementary algorithms (scanning, sorting, permuting), data structures (lists, arrays, B-trees), spatial data structures, algorithms for tree, graph, geometric, and spatial data, and the parallelization of data structures and algorithms. This course is intended for students interested in conducting research on or related to these topics. Students will conduct a semester long research project and present their results to the class. In addition to the scheduled meetings, students will have weekly one-

on-one meetings with the professor. Prerequisite: 600.363/463 and one of 600.315/415, 600.318/418, 600.316/416, 600.319/419 or permission of instructor. [Systems or Analysis]
Burns 3 hours

600.630 Computer Vision Seminar
This seminar course surveys recent research results in algorithms for dynamic vision and their applications. Specific emphasis will be placed on approaches which derive novel and efficient algorithms using generalizable mathematical and/or computational principles. Sample topics include: color and texture, segmentation and grouping, motion and tracking, stereo and structure from motion, image-based modeling, illumination and reflectance modeling, shape reconstruction, object and event recognition, face/gesture/gait modeling, statistical methods and learning, medical imaging, image and video retrieval, etc. Students will be expected to participate in class by reading, presenting, and discussing research papers. Prerequisites: 600.461 or 530.646 or permission of instructor. [Applications]
Hager 3 hours

600.643 Advanced Topics in Computer Security
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. Prerequisite: any 600.4xx course in computer security or cryptography including 600.442, 600.443 or 600.424; or permission of instructor. [Systems or Applications]
Rubin 3 hours

600.646 Advanced Computer-Integrated Surgery II
(See description under 600.446.)
This is substantially the same course as 600.446, but with an expectation that the term project will be more substantial. Prerequisite: 600.445 or permission of instructor. [Applications]
Taylor 3 hours spring

600.647 Advanced Topics in Wireless Networks
This class will survey current research in wireless communication networks. These types of networks have been growing exponentially in the past several years and include a host of different network types: ad hoc, cell phone, access point, sensor, etc. The class will build understanding of all layers of wireless networking and the interactions between them (including physical, data link, medium access control, routing, transport, and application). The topics of security, energy efficiency, mobility, scalability, and their unique characteristics in wireless networks will be discussed. Prerequisites: 600.344/444 and 600.363/463. [Systems or Analysis]
Awerbuch, Mishra 3 hours

600.651 Haptic Systems for Teleoperation and Virtual Reality
Open to undergraduates with permission. Graduate-level introduction to the field of haptics, focusing on teleoperated and virtual environments that are displayed through the sense of touch. Topics covered include human haptic sensing and control, design of haptic interfaces (tactile and force), haptics for teleoperation, haptic rendering and modeling of virtual environments, control and stability issues, and medical applications such as tele-surgery and surgical simulation. Course work includes reading and discussion of research papers, presentations, and a final project. Appropriate for students in any engineering discipline with interests in robotics, virtual reality, or computer-integrated surgical systems. (Co-listed with 530.561.) [Applications]
Okamura 3 hours

600.657 Advanced Topics for Computer Graphics
This course will present advanced methodologies and their applications to computer graphics. Topics will vary by semester. Students will be expected to present several papers throughout the semester and to participate in group discussions of the assigned readings. Prerequisite: any 600.4xx course in computer graphics & linear algebra; or permission of instructor. [Applications]
Kazhdan 3 hours

600.660 FFT in Graphics and Vision
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. Prerequisites: linear algebra and comfort with mathematical derivations. [Applications or Analysis]
Kazhdan 3 hours

600.663 Pattern Matching Algorithms
Pattern matching problems are among the oldest in computer science. Yet, the area is still a fertile ground for very active current research. Part of its appeal is in its many application domains, such as text editing, computer vision, or molecular biology. Another aspect is that pattern matching has produced or incorporated some novel and powerful algorithmic techniques. We will investigate various pattern matching problems with particular emphasis on the techniques employed for their solutions. Prerequisite: 600.363/463 or equivalent. [Analysis]
Amir 3 hours

600.664 Randomized Algorithms
Similar material as 600.464, presented in more depth. Intended for graduate students. Students may receive credit for 600.464 or 600.664, but not both. Pass/Fail only. Prerequisite: 600.463. [Analysis]
Kosaraju 3 hours fall

600.666 Information Extraction from Speech and Text
Introduction to statistical methods of speech recognition (automatic transcription of speech) and understanding. The course is a natural continuation of 600.465 but is independent of it. Topics include elementary information theory, hidden Markov models, the Baum and Viterbi algorithms, efficient hypothesis search methods, statistical decision trees, the estimation-maximization (EM) algorithm, maximum entropy estimation and estimation of discrete probabilities from sparse data for acoustic and language modeling. Weekly assignments and several programming projects. Prerequisites: 550.310 or equivalent, expertise in C or C++ programming. (Co-listed with 050.666 and 520.666.) [Applications]
Khudanpur 3 hours

600.667 Advanced Distributed Systems and Networks
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 envelope 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. Prerequisite: 600.437 or permission of instructor. [Systems]
Amir 3 hours spring

600.671 Special Topics on Bio-Nano Computing
This course will cover nanotechnology, bio-nanotechnology, introductory structural biology, molecular bioengineering, DNA computing, molecular electronics, and related fields with a focus on the design, fabrication, use, and development of systems with molecular-scale components. Previous knowledge of chemistry or macromolecular structure is not required. The course is appropriate for graduate and advanced undergraduate students in engineering, computer science, chemistry, and information technology-related fields. This course will be in lecture and discussion format. Students will read and discuss seminal papers in the field. [Applications]
Basu 3 hours

600.681 Advanced Topics in Computer Vision
This course covers state-of-the-art methods in dynamic vision, with an emphasis on segmentation, reconstruction and recognition of static and dynamic scenes. Topics include: reconstruction of static scenes (tracking and correspondence, multiple view geometry, self calibration), reconstruction of dynamic scenes (2-D and 3-D motion sementation, nonrigid motion analysis), recognition of visual dynamics (dynamic textures, face and hand gestures, human gaits, crowd motion analysis), as well as geometric and statistical methods for clustering and unsupervised learning, such as K-means, Expectation Maximization, and Generalized Principal Component Analysis. Applications in robotics and biomedical imaging are also included. Prerequisite: 600.461 & linear algebra or permission. [Co-listed as 580.681] [Applications]
Vidal 3 hours

600.726 Selected Topics in Programming Languages
This seminar course covers recent developments in the foundations of programming language design and implementation. Topics vary from year to year. Students will present papers orally. Prerequisite: permission of instructor.
Smith 1 hour

600.735 Selected Topics in Machine Learning
This seminar course will look at research in machine learning. Topics will be selected from those of mutual interest between students and the instructor. Sample topics include reinforcement learning, kernel methods, experimental methods in machine learning, computational learning theory, lazy learning, evolutionary computation, and neural networks. Students are expected to select papers and lead discussion. Prerequisite: permission of the instructor.
Sheppard 1 hour

600.742 Advanced Topics in Cryptography
(formerly 600.642)
This course will focus on advanced cryptographic topics with an emphasis on open research problems and student presentations. Prerequisite: 600.442 or 600.471 or permission of the instructor. [Applications]
Ateniese/Hohenberger 3 hours

600.743 Selected Topics in Systems
Weekly discussion based on current topics in the broad systems area. The goal of this effort is to expose all of us to current research and to foster greater communication and cooperation among the different groups doing research in the systems area here at Hopkins. Each student is responsible for reading the papers and participating in the discussion. Furthermore, every week one student will be responsible for creating a short presentation about the paper and leading the discussion. Prerequisite: permission of instructor.
Terzis/Burns 1 hour

600.745 Seminar in Computational Sensing and Robotics
This weekly seminar will focus on research issues in computer integrated surgery and robotics, including subjects such as medical image analysis, statistical modeling, visualization, vision/sensing, surgical planning, medical and non-medical robotics, and clinical applications. The purpose of the course is to widen the knowledge and awareness of the participants in current research in these areas, as well as to promote greater awareness and interaction between multiple research groups within the university and beyond. The format of the course is informal presentation by a pre-eminent invited speaker, followed by free discussion. (Co-listed as 520.744)
Kazanzides 1 hour

600.746 Selected Topics in Medical Image Analysis
This weekly seminar will focus on research issues in medical image analysis, including image segmentation, registra-

tion, statistical modeling, and applications. It will also include selected topics relating to medical image acquisition, especially where they relate to analysis. The purpose of the course 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 within the 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. Individual students will be assigned on a rotating basis to lead the discussion on particular papers or sections of papers. Co-listed in ECE as 520.746.
Taylor/Prince 1.5 hours

600.754 Selected Topics in Statistical Anatomic Models, Registration, and Reconstruction
This weekly research seminar will focus generally on statistical modeling of anatomic structures, image and model registration, 3D image reconstruction methods, and their interrelationships. We will concentrate primarily, though not exclusively, on x-ray based imaging modalities (x-ray fluoroscopy, CT, cone-beam tomography, "hybrid" reconstruction methods, etc.).
Taylor 1 hour

600.757 Selected Topics in Computer Graphics
This seminar course reviews current research in computer graphics. Prerequisite: permission of instructor.
Kazhdan 1 hour

600.758 Selected Topics in Computational Geometry
This course will provide a rapid and intense 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. Students will be expected to complete the assigned reading before class, and the course time will be spent on discussions and exercises.
Kazhdan 1 hour

600.761 Computer Vision Techniques for Multi-Sensor Image Fusion
With the continuing advancement of various sensor technologies, multiple imaging modalities are more often becoming simultaneously available for deriving information from the world. In medical imaging, MRI, CAT, and PET modalities can be separately used to image the same tissue, providing complementary information for visualization and diagnosis. Cameras using objective lenses are now available that image in the visible, Near-infrared, ShortWave-infrared and Thermal Infrared spectrums; in combinations of two or more modalities these can provide vastly enhanced information about the physical world. This seminar will study a variety of computer vision techniques for both visual image fusion, such as for enhancing human visual perception beyond the visible spectrum, as well as analytic image fusion such as for enhancing the performance of automated object and face recognition. Recommended: 600.641 or equivalent.
Wolff 2 hours

600.762 Selected Topics in Visual Medical Imaging
Visual imaging is used extensively in medical applications, including endoscopy, minimally invasive surgery, and newer systems for stereo imaging and more recent techniques such as NOTES, and capsule endoscopy. This weekly seminar will focus on current research issues in visual imaging for medical applications. The purpose of the course is to allow the participants to develop a deeper understanding of the current research in methods, device, and applications of these technologies as well as to motivate interaction between research groups. Participants will select from a set of papers to summarize from a list of course papers and participate in the discussion.
Kumar 1 hour

600.765 Selected Topics in Natural Language Processing
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: 600.465 or permission of instructor.
Eisner 1 hour

600.766 Selected Topics in Meaning, Translation, and Generation of Text
The 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. Prerequisite: permission of instructor.
Callison-Burch, VanDurme 1 hour

600.775 Current Topics in Machine Learning
A reading group exploring current research topics in machine learning. Topics will be selected based on interests of the students and the instructor. Papers will include current research and tutorials. Our focus will be on core machine learning methods as opposed to applications. Enrolled students are expected to present papers and lead discussion. Prerequisite: 600.475 or another machine learning course suggested.
Dredze 1 hour

600.801-802 Dissertation Research

600.803-804, 874, 895 Graduate Research
Independent research for master's or pre-dissertation Ph.D. students.

600.809-810, 876, 891 Graduate Independent Study
Individual study in an area of mutual interest to a graduate student and a faculty member in the department.

Robotics Courses

This listing is provided to help graduate students with an interest in robotics choose appropriate courses for their program of study in consultation with their faculty advisor.

Biomedical Engineering

580.631 Biomechanics and Motor Control

Computer Science

600.435 Artificial Intelligence
600.445 Computer-Integrated Surgery I
600.446 Computer-Integrated Surgery II
600.452 Computer-Integrated Surgery Seminar
600.461 Computer Vision
600.630 Computer Vision Seminar
600.646 Advanced Computer-Integrated Surgery II
600.651 Haptic Systems for Teleoperation and Virtual Reality
600.681 Advanced Topics in Computer Vision
600.745 Seminar in Computational Sensing and Robotics
600.746 Selected Topics in Medical Image Analysis
600.754 Selected Topics in Statistical Anatomic Models, Registration, and Reconstruction
600.762 Selected Topics in Visual Medical Imaging

Electrical and Computer Engineering

520.353 Control Systems
520.454 Control Systems Design
520.608 Image Reconstruction and Restoration
520.621 Introduction to Nonlinear Systems

Mechanical Engineering

530.343 Design and Analysis of Dynamic Systems
530.420 Robot Actuators and Sensors
530.421 Mechatronics
530.424 Dynamics of Robots and Spacecraft
530.646 Introduction to Robotics
530.647 Adaptive Systems
530.649 Robot Motion Planning
530.651 Haptic Systems for Teleoperation and Virtual Reality

Courses in Language and Speech Processing

This listing is provided to help graduate students with an interest in language and speech processing choose appropriate courses for their program of study in consultation with their faculty advisor.

Cognitive Science

050.370/670 Formal Methods in Cognitive Science: Language
050.371/671 Formal Methods in Cognitive Science: Inference
050.372/672 Formal Methods in Cognitive Science: Neural Networks
050.317/617 Semantics I
050.320/620 Syntax I
050.321/621 Syntax II
050.325/625 Phonology I
050-327/627 Phonology II
050.333 Psycholinguistics
050.630 Topics in Language Processing

Computer Science

600.465 Introduction to Natural Language Processing
600.466 Information Retrieval and Web Agents
600.765 Selected Topics in Natural Language Processing
600.766 Selected Topics in Meaning, Translation, and Generation of Text
600.775 Current Topics in Machine Learning

Electrical and Computer Engineering

520.419 Theory and Design of Iterative Algorithms
520.447 Introduction to Information Theory and Coding
520.478 Theory and Practice of Large Vocabulary Speech Recognition
520.666 Information Extraction from Speech and Text
520.674 Information Theoretic Methods in Statistics
520.735 Sensory Information Processing