Computing has grown to be a pervasive element of science, business, society, and entertainment. The availability of relatively inexpensive high performance computing capabilities, ubiquitous high speed and wireless networking, and mobile computing have powered 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 worthy of study unto itself, and/or as an empowering discipline to be studied in conjunction with other areas. Core CS careers include (but are not limited to) software design and development, computer systems engineering or administration, and information security. Application areas span a wide range of fields and disciplines such as robotics, medical or health informatics, gaming/entertainment, business computing, and scientific research to name a few.

Because computer science is a highly diverse and broadly applied field, studies can proceed in many different directions. Accordingly, the undergraduate and graduate programs in the Department of Computer Science at Johns Hopkins are flexible curricula designed to accommodate a wide range of goals. A student at Johns Hopkins can pursue appropriately customized versions of the following computer science programs: minor, bachelor of science, bachelor of arts, masters of science in engineering, and doctor of philosophy. Most of this 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, health informatics, computational medicine, 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, computational genomics, 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), the Institute for Data Intensive Engineering and Science (IDIES), and the Malone Center for Engineering in Healthcare (MCEH). An important component of the educational process in the department is the opportunity for undergraduate and graduate student participation in the research programs of the faculty. In particular, original research in close association with individual faculty members is emphasized at the graduate level.

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

At the graduate level, the LCSR (http://e-catalog.jhu.edu/departments-program-requirements-and-courses/engineering/robotics-computational-sensing) offers a Master of Science in Engineering (M.S.E.) in Robotics, designed for students from a wide variety of engineering, scientific, and mathematical backgrounds to advance their interdisciplinary knowledge in robotics. Details of this program may be found in the LCSR section of this catalog, or on the web at www.lcsr.jhu.edu/MSE. Lastly, the Master of Science in Security Informatics (M.S.S.I.) is a specialized graduate program offered through the Information Security Institute (http://e-catalog.jhu.edu/departments-program-requirements-and-courses/engineering/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 M.S.S.I. online at www.jhuisi.jhu.edu or in the ISI section of this catalog.

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

Facilities

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

The general department computing facilities include numerous workstations and servers. Undergraduate laboratories combine to provide approximately 24 Linux workstations, and several Windows stations. One of these labs is a collaboration room allowing students to work in a team-based environment, with several private breakout rooms as well. At the graduate level there is a Master’s Lab consisting of a collaboration area and workstation area, both consisting of several Linux workstations. All Ph.D. students are assigned dedicated desks in their research labs. Additionally, multiple high-speed networked laser printers, as well as a networked color copier and remotely accessible Linux compute servers are 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, high-performance computing clusters, sensor and wireless computing testbeds, robots and computer vision systems, a mock
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 combined 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 (http://www.cs.jhu.edu/undergraduate-studies/academics/ugrad-advising-manual) or on the website at www.cs.jhu.edu.

All undergraduate students majoring or minorin 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 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.
- Lead teams and provide vision for innovation.
- 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 particular focus area within the field. Current foci primarily reflect departmental and school research strengths: big data, computational biology, fundamentals of computing, information security, natural language processing, robotics, systems and networking; while a few are directed towards career paths: business computing, software engineering, and video game design. Further information on these focus areas may be found in the computer science undergraduate advising manual (http://www.cs.jhu.edu/undergraduate-studies/academics/ugrad-advising-manual).

Details and course recommendations for these distributional requirements are below. These requirements add up to 100 credits and fulfill general university and WSE requirements, leaving 26 pure elective credits. 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.

The following foundational courses in computer science must be included in a student’s program:

<table>
<thead>
<tr>
<th>Course</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN.600.120</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.104</td>
<td>1</td>
</tr>
<tr>
<td>EN.600.107</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.110</td>
<td>4</td>
</tr>
<tr>
<td>EN.600.226</td>
<td>4</td>
</tr>
<tr>
<td>EN.600.233</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.271</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.363</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.463</td>
<td>3</td>
</tr>
</tbody>
</table>

or EN.600.463 Algorithms I

The following foundational courses in computer science must be included in a student’s program:

<table>
<thead>
<tr>
<th>Course</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN.600.104</td>
<td>1</td>
</tr>
<tr>
<td>EN.600.107</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.120</td>
<td>4</td>
</tr>
<tr>
<td>EN.600.226</td>
<td>4</td>
</tr>
<tr>
<td>EN.600.233</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.271</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.363</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.463</td>
<td>3</td>
</tr>
</tbody>
</table>

or EN.600.463 Algorithms I

The Bachelor of Science in Computer Science degree program is accredited by the Computing Accreditation Commission of ABET, www.abet.org (http://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 trade-offs 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:

<table>
<thead>
<tr>
<th>Department</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science</td>
<td>42</td>
</tr>
<tr>
<td>Mathematics</td>
<td>24</td>
</tr>
<tr>
<td>Basic Sciences</td>
<td>16</td>
</tr>
<tr>
<td>Humanities/Social Sciences</td>
<td>18</td>
</tr>
<tr>
<td>Two Writing Intensive Courses</td>
<td></td>
</tr>
<tr>
<td>Electives</td>
<td>26</td>
</tr>
<tr>
<td>Total Credits</td>
<td>126</td>
</tr>
</tbody>
</table>
At least 16 credit hours must be at the 300-level or above, including 13 EN.600.363/463. 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. The course satisfying this requirement may overlap other requirements:

EN.600.250 User Interfaces and Mobile Applications
EN.600.255 Introduction to Video Game Design
EN.600.321/42 Object Oriented Software Engineering
EN.600.355 Video Game Design Project
EN.600.411 Computer Science Innovation & Entrepreneurship II
EN.600.446 Computer Integrated Surgery II

EN.600.520 Senior Honors Thesis

Eight additional credits of Computer Science are required.** 8

Mathematics (24 credits)
The following courses must be included:

AS.110.108 Calculus I 4
AS.110.109 Calculus II (For Physical Sciences and Engineering) 4
EN.550.171 Discrete Mathematics 4

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, Linear Algebra, Calculus III, 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, Natural Sciences designated engineering courses, and some Natural Sciences designated courses in Neuroscience & Cognitive Science, but not all. At most 2 credits from satisfactory/unsatisfactory intersession courses may be used to fulfill this requirement.

Humanities/Social Sciences (18 credits)
Six courses in the Humanities and Social and Behavioral Sciences must be taken, with each course at least 3 credits. These courses must have either an Humanities or Social and Behavioral Sciences area designator on them; however, foreign language courses (without an Humanities or Social and Behavioral Sciences) may also be used to satisfy this requirement.

Writing Requirement
Students are required to fulfill the university’s requirement of two writing intensive courses, each at least 3 credits. Students must receive at least a C- grade or better in these writing courses. At least one course must be explicitly focused on writing skills (eg, Expository Writing, Professional Communication, Fiction & Poetry). These courses may overlap other requirements.

Electives
Electives are to be chosen by the student with guidance and approval of his/her advisor.

Total Credits 126

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:

Computer Science 32
Mathematics 20
Basic Sciences 12
Humanities/Social Sciences 18
Foreign Language 6
4 Writing Intensive Courses 32

Total Credits 120

Details and course recommendations of these distributional requirements are below. These requirements add up to 88 credits and fulfill general university distribution requirements. 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 (32 credits) *

The following foundational courses in computer science must be included in a student’s program:

EN.600.107 Introductory Programming in Java (or equivalent) 3
EN.600.120 Intermediate Programming 4
EN.600.226 Data Structures 4
EN.600.233 Computer System Fundamentals 3
EN.600.271 Automata & Computation Theory 3
EN.600.363 Introduction To Algorithms 3
or EN.600.463 Algorithms I

At least 15 credit hours must be at the 300-level or above, including EN.600.363/463.

Mathematics (20 credits)
The following courses must be included:

AS.110.108 Calculus I 4
AS.110.109 Calculus II (For Physical Sciences and Engineering) 4

* No more than 6 independent type credits (courses numbered 600.xxx) and no more than 3 credits of short courses (1-credit special topics courses) can be counted toward this requirement. However, B.S. students doing the Senior Honors Thesis (EN.600.519-EN.600.520) may use an additional three credits of independent work toward their CS requirements, for a total of 9 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.

** Up to 6 of the 42 required credits may be from an approved list of relevant courses in other departments, which includes courses cross-listed in CS. These courses may only count as “CS other” credits, not upper level CS credits (regardless of course level in the other department). See department website for the list.

*** AP Statistics credits may not be used to satisfy these credit requirements; however, they do meet the need for coverage of statistics (not probability).
The remaining courses must be chosen from Mathematics (110.xxx) or Applied Math and Statistics (550.xxx). At least one course must be 200-level or above. **

### 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, ‘N’ designated engineering courses, and some ‘N’ designated courses in Neuroscience and Cognitive Science, but not all. At most 2 credits from (S/U) intersession courses may be used to fulfill this requirement.

### 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 200-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 Humanities or Social Sciences area designators may fulfill these distributional requirements.

### Foreign Language (6 credits)

Two courses in a foreign language, with a total of at least 6 credits are required. These foreign language courses are in addition to the 18 required Humanities/Social Sciences credits.

### Writing Requirement

All primary computer science majors pursuing a B.A. degree are required to fulfill the university’s requirement of four writing intensive courses, each at least 3 credits. At least one course must be explicitly focused on writing skills (eg, Expository Writing, Professional Communication, Fiction & Poetry). Students must receive at least a C- grade in these courses. These courses may overlap other requirement areas.

### Electives

Electives are to be chosen by the student with guidance and approval of his/her advisor.

Total Credits 120

* No more than 3 independent type credits (courses numbered 600.5xx) and no more than 3 credits of short courses (1-credit special topics courses) can be counted toward this requirement. However, B.A. students doing the Senior Honors Thesis (EN.600.519-EN.600.520) 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.

** Highly recommended: Probability/Statistics, Linear Algebra, Calculus III, Differential Equations. Note that students will need at least five courses to fulfill the credit requirement.

### 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 23 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 area in depth.

While not explicitly required, we also strongly recommend taking EN.550.171 Discrete Mathematics as preparation for several computer science courses, including EN.600.271 Automata & Computation Theory and EN.600.363 Introduction To Algorithms.

### Core Courses *

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN.600.107</td>
<td>Introductory Programming in Java (or AP credit)</td>
<td>3</td>
</tr>
<tr>
<td>EN.600.120</td>
<td>Intermediate Programming</td>
<td>4</td>
</tr>
<tr>
<td>EN.600.226</td>
<td>Data Structures</td>
<td>4</td>
</tr>
</tbody>
</table>

Choose one of the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN.600.233</td>
<td>Computer System Fundamentals</td>
</tr>
<tr>
<td>EN.600.271</td>
<td>Automata &amp; Computation Theory</td>
</tr>
</tbody>
</table>

### Upper-Level Courses

Minors must take three CS courses at the 300-level or above. 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. **

Total Credits 23

* 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.

** Each upper-level course description in this catalog includes its area for reference. In addition, a current listing of courses with area designators is provided on the departmental website.

Some upper level courses have math pre-requisites as well (Probability/Statistics or Linear Algebra most commonly), so plan ahead.

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

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.

### 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.

**Combined Bachelor’s/Master’s Program**

As early as the end of their sophomore year, qualified students may apply for admission to a combined bachelor’s/master’s program which combines a B.S. or B.A. degree (in any department) with a master of science in engineering degree in Computer Science. This program allows students to simultaneously pursue both an undergraduate and a graduate degree program of study. Generally, the combined B.S./M.S.E. or B.A./M.S.E. program is accomplished in five years, although some students take more or less time. Applicants are judged on the basis of their performance in courses and their letters of recommendation. Double counting of at most two courses is subject to current WSE and departmental policies. Students may not take a 600.3xx course as an undergraduate and the corresponding 600.4xx course for the M.S.E.; likewise for 600.4xx/6xx course offerings. 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 the Graduate Programs tab on this page, or ask in the departmental office for the document that lists those requirements.

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 daytime in-person 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/remote study should refer to Engineering for Professionals at ep.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 expected to have coursework (or equivalent experience) in intermediate programming (C++ and Java), data structures, computer system fundamentals, 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 courses (3 class hours (credits) each) 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 with area designators (http://cs.jhu.edu/courses/courseareas.pdf) 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 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.

Computer Science graduate students may count 400-level courses in our department towards their degree, in addition to the standard 600-level and above graduate courses. The coursework program must be approved by the student’s faculty advisor.

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 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 EN.600.601 Computer Science Seminar-EN.600.602 Computer Science Seminar, 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 taken at a graduate-appropriate level. In the Department of Computer Science, this includes courses that are 600-level and above, as well as 400-level courses.
- At most, two courses with grades less than B- may be counted toward the coursework requirements. No courses with grades less than C- may be counted.
- The overall grade point average of the courses counted toward the coursework requirements must be a 3.0 or higher (B average).
- At most, two independent study courses can be counted toward the course requirements.
- Other than independent study courses, no courses with grades of P can be counted toward the coursework requirement. Courses with grades of P 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.

• Any master’s student engaged in research for payment or to help meet degree requirements is required to complete Responsible Conduct of Research training. Students receiving payment from NIH training grants or fellowships must take the in person course — AS.360.625 Responsible Conduct of Research. All other students can take the course online—AS.360.624 Responsible Conduct of Research (Online). Instructions for accessing and signing up for the course can be found here: http://engineering.jhu.edu/wse-research/resources-policies-forms/responsible-conduct-of-research/online-training-course-for-the-responsible-conduct-of-research/

Additional information regarding this training can be found here: http://eng.jhu.edu/wse/page/conduct-of-research-training. Students who are required to complete this training will not receive a diploma until the course has been completed.

Tuition Support
M.S.E. students are not normally eligible for tuition waivers, but will be able to work on campus up to 19.99 hours per week for hourly rates. There are also course assistant positions 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. For more detailed policies regarding the PhD program, please visit the Advising Manual (http://www.cs.jhu.edu/graduate-studies/phd-requirements) on our departmental website.

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.

Responsible Conduct of Research
All doctoral students are required to take AS.360.625 Responsible Conduct of Research. Students are expected to complete the course by the end of the first year but have until the start of the fourth semester to meet this requirement. Failure to do so may result in a loss of funding. Additional information regarding this requirement can be found here: http://eng.jhu.edu/wse/page/conduct-of-research-training.

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 courses (3 class hours each) 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 with area designators (http://cs.jhu.edu/courses/courseareas.pdf) 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 courses.

Computer Science graduate students may count 400-level courses in our department towards their degree, in addition to the standard 600-level and above graduate courses. The coursework program must be approved by the student’s faculty advisor. 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, no courses with grades of P can be counted toward the coursework 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. Students are expected to complete the qualifying projects by the end of their third 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, ideally taken in the student’s third year. 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 formatted and submitted to the Milton S. Eisenhower Library for electronic publication (http://guides.library.jhu.edu/etd).

Teaching Requirement
Beginning in Fall 2015, all Ph.D. students are required to serve as a Teaching Assistant at least one semester during their program of study. As part of the requirement, the supervising course instructor must give the TA an opportunity to be in front of a group of students at least once during the course. Students are required to sign-up for the course EN.600.807 Teaching Practicum during the semester in which the requirement is being fulfilled, and at the end of the semester his/her performance will be evaluated by the course instructor.

Student Progress Review
Ph.D. students are reviewed annually by their advisor(s) and the department, and notified in writing as to their standing in the program. Students deemed to not be making satisfactory progress may be placed on probation.

Financial Aid
Financial aid is available for candidates of high promise. Fellowships provide a student with a stipend plus tuition. Research assistantships are available on sponsored research projects directed by members of the faculty. Teaching assistantships normally consist of tuition plus stipend commensurate with the teaching or grading duties assigned. 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.

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

Faculty Chair
Yair Amir
Professor: distributed systems, resilient clouds and critical infrastructure, overlay networks, distributed algorithms.

Professor Emeritus
Gerald M. Masson
Fault tolerant computing, systems diagnosis theory, distributed computing, real-time error monitoring of hardware/software, software design for testability, interconnection networks.

Professors
Randal Burns
Storage systems, high performance and scientific computing, and database federations.

Jason M. Eisner
Computational linguistics (syntax and phonology), natural language processing, statistical machine learning, programming language design.

Gregory D. Hager
Mandell Bellmore Professor; Director, Malone Center for Engineering in Healthcare: vision, robotics, human-machine systems, computer-integrated medicine.

Philipp Koehn
Statistical machine translation.

S. Rao Kosaraju
Edward J. Schaefer Professor in Engineering: design of algorithms, pattern matching, and derandomization.

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

Steven Salzberg
Bloomberg Distinguished Professor: genomics, bioinformatics, genome assembly, gene finding, sequence analysis algorithms.

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

Alexander Szalay
Bloomberg Distinguished Professor: data intensive computing, theoretical astrophysics, galaxy formation.

Russell H. Taylor
John C. Malone Professor; Director, LCSR & CISST ERC: medical robotics, computer-integrated interventional medicine, medical image analysis, human-machine robotic systems.

David Yarowsky
Natural language and speech processing, information retrieval, machine translation, and machine learning.

Alan Yuille
Bloomberg Distinguished Professor: computational models of vision, mathematical models of cognition, artificial intelligence, neural networks.
**Associate Professors**

Michael Kazhdan  
Computer graphics, surface reconstruction, image and geometry processing.

Michael Schatz  
Bloomberg Distinguished Associate Professor: novel algorithms for comparative genomics, human genetics, and personalized medicine.

**Assistant Professors**

Yanif Ahmad  
Data management, stream processing, declarative languages, parallel and distributed databases.

Raman Arora  
Machine learning, statistical signal processing, stochastic approximation algorithms, applications to speech and language processing.

Alexis Battle  
Genetics of complex traits, graphical models, transfer learning, structured regularization methods.

Vladimir Braverman  
Algorithms, massive data sets, data streams, and database systems.

Michael Dinitz  
Theoretical computer science, approximation algorithms, applications to networks and distributed computing.

Matthew Green  
Applied cryptography, cryptographic protocol design, analysis of practical security systems, privacy-preserving storage and identification technologies.

Abhishek Jain  
Cryptography, security, theoretical computer science.

Benjamin Langmead  
Computational genomics, sequence alignment, text indexing, parallel and high performance computing.

Xin Li  
Theory of computation, randomness, complexity theory, distributed computing, cryptography.

Suchi Saria  
Machine learning, computational medicine, health informatics, and applications of machine learning in natural language processing, activity recognition and human-machine systems.

Ilya Shpitser  
John C. Malone Assistant Professor: causal inference and missing data, graphical models, longitudinal data analysis in medicine and epidemiology.

Benjamin Van Durme  
Artificial intelligence, natural language processing (computational semantics), and streaming algorithms.

**Research Professors**

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

Peter Kazanzides  
Medical robots, computer-assisted surgery, real-time systems.

Nassir Navab  
Augmented reality, vision, medical image computing and computer assisted interventions.

**Associate Research Professors**

Philippe Burlina  
Computer vision, visual analysis and communications, multi-modality image exploitation, enterprise software systems for content and e-process management.

Chris Callison-Burch  
Statistical natural language processing, machine translation, paraphrasing, evaluation of human language technologies.

Susan Hohenberger-Waters  
Cryptography, computer security, algorithms, and complexity theory.

James Mayfield  
Information retrieval, cross-language retrieval, information extraction, natural language processing.

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

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

**Assistant Research Professors**

Mark Dredze  
Machine learning, natural language processing, health informatics.

Kevin Duh  
Natural language processing, machine learning.

Peng Huang  
Computer systems, software engineering, programming languages.

Xin Jin  
Networked systems, cloud computing, networking.

Simon Leonard  
Visual servoing, hand-eye coordination.

Christine Piatko  
Computational geometry, information visualization, information retrieval.

Austin Reiter  
Application of computer vision to robotics, machine vision, 3D reconstruction, image registration, visual recognition.

I-Jeng Wang  
Wireless networking, Bayesian networks, probabilistic models.

**Associate Teaching Professors**

Sara Miner More  
Foundations of computing, computer science education, cryptography, information flow.

Joanne Selinski  
Director of Undergraduate Studies: CS education, software engineering.
**Senior Lecturer**
Peter Fröhlich
Programming languages, software engineering, systems software, video game design, web applications.

**Part-time Lecturers**
Sheela Kosaraju
Computer ethics.

Harold Lehmann
Medical informatics.

**Visiting Professor**
Mitra Basu
Computational biology, pattern recognition, neural networks, artificial intelligence.

**Associate Research Scientist**
Anton Dahbura
Executive Director, Information Security Institute: Information security, fault-tolerant computing, distributed systems, testing.

**Assistant Research Scientist**
Matt Post
Machine translation, syntax, parsing and language modeling.

**Adjunct Professors**
Gabor Fichtinger
Applied surgical robotics, surgical CAD/CAM systems, percutaneous therapies, stereotactic radiosurgery.

Peter Freeman
Future internet, history and future of computing, innovation and science policy.

John W. Sheppard
Artificial intelligence, machine learning, data mining.

**Adjunct Assistant Professor**
Han Liu
Statistical machine learning, high dimensional nonparametric learning and massive data analysis, multiple hypotheses testing, time series analysis, genomics, proteomics, cognitive neuroscience.

**Adjunct Associate Research Professors**
Seth Nielson
Network security.

Lanier Watkins
Critical infrastructure security, network security.

**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.

Tamas Budavari
Assistant Professor (Applied Mathematics and Statistics): computational statistics, bayesian inference, low-dimensional embeddings, streaming and randomized algorithms.

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

Noah Cowan
Associate 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.

Liliana Florea
Assistant Professor (McKusick-Nathans Institute for Genetic Medicine): application of computation techniques towards modeling and solving problems in biology and genetic medicine.

Rachel Karchin
Associate 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

Marin Kobilarov
Assistant Professor (Mechanical Engineering): intelligent robotic vehicles, analytical and computational methods for mechanics, control, motion planning, and reasoning under uncertainty.

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

Mihaela Pertea
Assistant Professor (Medicine): computational tools for RNA-seq analysis, gene finding, splice site prediction, sequence motif finding.

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.

Rebecca Schulman
Assistant Professor (Chemical and Biomolecular Engineering): molecular programming, DNA nanotechnology, self-assembly (theory and experiment), theoretical and systems biology, smart materials, nanoscale robotics.

Ralph Semmel
Professor (Director, APL): artificial intelligence, database systems, software engineering.

Jeff Siewerdsen
Professor (Biomedical Engineering): imaging physics, diagnostic radiology, image-guided interventions.
James Taylor  
Associate Professor (Biology): genome informatics.

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

Joshua T. Vogelstein  
Assistant Professor (Biomedical Engineering): big data science, connectomics, statistical neuroscience.

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.

Thomas B. Woolf  
Professor (Physiology): bioinformatics, computational biophysics.

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

Courses

**EN.600.104. Computer Ethics. 2.00 Credits.**  
Students will examine a variety of topics regarding policy, legal, and moral issues related to the computer science profession itself and to the proliferation of computers in all aspects of society, especially in the era of the Internet. The course will cover various general issues related to ethical frameworks and apply those frameworks more specifically to the use of computers and the Internet. The topics will include privacy issues, computer crime, intellectual property law – specifically copyright and patent issues, globalization, and ethical responsibilities for computer science professionals. Work in the course will consist of weekly assignments on one or more of the readings and a final paper on a topic chosen by the student and approved by the instructor.

Instructor(s): S. Kosaraju  
Area: Humanities.

**EN.600.105. M & Ms: Freshman Experience. 1.00 Credit.**  
This course is required for all freshmen 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.

Instructor(s): J. Selinski

**EN.600.107. Introductory Programming in Java. 3.00 Credits.**  
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 EN.600.108 concurrently in Fall/Spring semesters.

Prerequisites: Students may receive credit for EN.600.107 or EN.600.112, but not both.

Instructor(s): S. More  
Area: Engineering.

**EN.600.108. Introduction to Programming Lab. 1.00 Credit.**  
This course is intended for novice programmers, and must be taken in conjunction with EN.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. Must have familiarity with computers. Satisfactory/Unsatisfactory only.

Prerequisites: Coreq for EN.600.108: EN.600.107  
Corequisites: Coreq: EN.600.107

Instructor(s): S. More  
Area: Engineering.

**EN.600.112. Introductory Programming for Scientists and Engineers. 3.00 Credits.**  
An introductory learning by doing programming course for scientists, engineers, and everybody else who will need basic programming skills in their studies and careers. We cover the fundamentals of structured, modular, and (to some extent) object-oriented programming as well as important design principles and software development techniques. We will apply our shiny new programming skills by developing computational solutions in the Python programming language to a number of real-world problems from a variety of disciplines. This course may not be used for the CS major or minor requirements, except as a substitute for 600.107. Summer only: Students will be expected to do significant programming (15-20 hours/wk). Attendance and participation is required. Students must have a laptop on which to install and run the programming environment.

Prerequisites: Students who have taken EN.600.107 may not register for EN.600.112; Students who have taken EN.600.111 may not register for EN.600.112.

Instructor(s): J. Selinski  
Area: Engineering.

**EN.600.113. IPSE Lab. 1.00 Credit.**  
This course is intended for novice programmers, and must be taken in conjunction with EN.600.112. The purpose of this course is to give first-time programmers hands-on practice with guided supervision. Students may receive credit for EN.600.108 or EN.600.113, but not both. Corequisite: EN.600.112. Satisfactory/Unsatisfactory only.

Prerequisites: Corequisite: EN.600.112; Students may receive credit for EN.600.108 or EN.600.113, but not both.

Instructor(s): J. Selinski  
Area: Engineering.

**EN.600.120. Intermediate Programming. 4.00 Credits.**  
This course teaches intermediate to advanced programming, using C and C++. (Prior knowledge of these languages is not expected.) We will cover low-level programming techniques, as well as object-oriented class design, and the use of class libraries. Specific topics include pointers, dynamic memory allocation, polymorphism, overloading, inheritance, templates, collections, exceptions, and others as time permits. Students are expected to learn syntax and some language specific features independently. Course work involves significant programming projects in both languages. Recommended Course Background: AP CS, EN.600.107, EN.600.111, EN.600.112 or equivalent.

Instructor(s): M. Kazhdan; S. More  
Area: Engineering.
EN.600.145. Mini Term: Introduction to Computer-Integrated Surgery. 1.00 Credit.
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. No computer programming will be necessary to complete the assignments.
Instructor(s): S. Leonard
Area: Engineering.

EN.600.226. Data Structures. 4.00 Credits.
This course covers the design and implementation of data structures including arrays, stacks, queues, linked lists, binary trees, heaps, balanced trees (e.g. 2-3 trees, AVL-trees) and graphs. Other topics include sorting, hashing, memory allocation, and garbage collection. Course work involves both written homework and Java programming assignments.
Prerequisites: EN.600.107 or EN.600.120 or permission of instructor.
Instructor(s): J. Selinski; P. Froehlich; S. More
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.233. Computer System Fundamentals. 3.00 Credits.
[Formerly EN.600.333/433] We study the design and performance of a variety of computer systems from simple 8-bit micro-controllers through 32/64-bit RISC architectures all the way to ubiquitous x86 CISC architecture. We'll start from logic gates and digital circuits before delving into arithmetic and logic units, registers, caches, memory, stacks and procedure calls, pipelined execution, super-scalar architectures, memory management units, etc. Along the way we'll study several typical instruction set architectures and review concepts such as interrupts, hardware and software exceptions, serial and other peripheral communications protocols, etc. A number of programming projects, frequently done in assembly language and using various processor simulators, round out the course. [Systems]. Students may receive credit for only one of EN.600.233, EN.600.333 or EN.600.433.
Instructor(s): P. Froehlich
Area: Engineering.

EN.600.250. User Interfaces and Mobile Applications. 3.00 Credits.
This course will provide students with a rich development experience, focused on the design and implementation of user interfaces and mobile applications. A brief overview of human computer interaction will provide context for designing, prototyping and evaluating user interfaces. Students will invent their own mobile applications and implement them using the Android SDK, which is JAVA based. An overview of the Android platform and available technologies will be provided, as well as XML for layouts, and general concepts for effective mobile development. Students will be expected to explore and experiment with outside resources in order to learn technical details independently. There will also be an emphasis on building teamwork skills, and on using modern development techniques and tools.
Prerequisites: EN.600.120 AND EN.600.226
Instructor(s): J. Selinski
Area: Engineering.

EN.600.255. Introduction to Video Game Design. 3.00 Credits.
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, archetypical 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. Section 1 requires technical skills, including at least one programming course (preferably 2 or more). Section 2 requires artistic skills, including at least one multimedia course (preferably 2 or more). Open to sophomores and above.
Corequisites: Co-req: 600.256
Instructor(s): P. Froehlich
Area: Engineering.

EN.600.256. Introduction to Video Game Design Lab. 1.00 Credit.
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.
Corequisites: EN.600.255(C)
Instructor(s): P. Froehlich.

EN.600.271. Automata & Computation Theory. 3.00 Credits.
This course is an introduction to the theory of computing. topics include design of finite state automata, pushdown automata, linear bounded automata, Turing machines and phrase structure grammars; correspondence between automata and grammars; computable functions, decidable and undecidable problems, P and NP problems, NP-completeness, and randomization. Students may not receive credit for EN.600.271 and EN.600.471 for the same degree.
Prerequisites: EN.550.171
Instructor(s): X. Li
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.315. Databases. 3.00 Credits.
Introduction to database management systems and database design, focusing on the relational and object-oriented data models, query languages and query optimization, transaction processing, parallel and distributed databases, recovery and security issues, commercial systems and case studies, heterogeneous and multimedia databases, and data mining. [Systems] (www.cs.jhu.edu/~yarowsky/cs415.html) Students may receive credit for EN.600.315 or EN.600.415, but not both.
Prerequisites: Students may receive credit for 600.315 or 600.415, but not both.
Corequisite: EN.600.256
Instructor(s): D. Yarowsky
Area: Engineering.
EN.600.316. Database Systems. 3.00 Credits.

This course serves as an introduction to the architecture and design of modern database management systems. Topics include query processing algorithms and data structures, data organization and storage, query optimization and cost modeling, transaction management and concurrency control, high-availability mechanisms, parallel and distributed databases, and a survey of modern architectures including NoSQL, column-oriented and streaming databases. Course work includes programming assignments and experimentation in a simple database framework written in Java. [Systems] Students may receive credit for EN.600.316 or EN.600.416, but not both. Recommended Course Background: EN.600.120 AND EN.600.226; Students may receive credit for EN.600.316 or EN.600.416, but not both.

Instructor(s): Y. Ahmad
Area: Engineering.

EN.600.318. Operating Systems. 3.00 Credits.

This course covers fundamental topics related to operating systems theory and practice. Topics include processor management, storage management, concurrency control, multi-processing 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. Students may receive credit for EN.600.318 or EN.600.418 but not both. Recommended Course Background: EN.600.21

Prerequisites: Students may receive credit for EN.600.318 or EN.600.418 but not both. [Prereqs: EN.600.120 AND EN.600.226 AND (EN.600.233 OR EN.600.333)]

Instructor(s): P. Froehlich
Area: Engineering.

EN.600.320. Parallel Programming. 3.00 Credits.

This course prepares the programmer to tackle the massive data sets and huge problem size of modern scientific and enterprise computing. Google and IBM have commented that undergraduate CS majors are unable to break the single server mindset (http://www.google.com/intl/en/press/pressrel/20071008.ibm_univ.html). Students taking this course will abandon the comfort of serial algorithmic thinking and learn to harness the power of cutting-edge software and hardware technologies. The issue of parallelism spans many architectural levels. Even “single server” systems must parallelize computation in order to exploit the inherent parallelism of recent multi-core processors. The course will examine different forms of parallelism in four sections. These are: (1) massive data-parallel computations with Hadoop; (2) programming compute clusters with MPI; (3) thread-level parallelism in Java; and, (4) GPGPU parallel programming with NVIDIA’s Cuda. Each section will be approximately 3 weeks and each section will involve a programming project. The course is also suitable for undergraduate and graduate students from other science and engineering disciplines that have prior programming experience. [Systems] Students may receive credit for EN.600.320 or EN.600.420, but not both. Recommended Course Background: EN.600.233

Instructor(s): R. Burns
Area: Engineering.

EN.600.321. Object Oriented Software Engineering. 3.00 Credits.

This course covers object-oriented software construction methodologies and their application. The main component of the course is a large team project on a topic of your choosing. Course topics covered include object-oriented analysis and design, UML, design patterns, refactoring, program testing, code repositories, team programming, and code reviews. [Systems or Applications] http://pl.cs.jhu.edu/oose/index.shtml Students may receive credit for EN.600.321 or EN.600.421, but not both. Prerequisites: Students may receive credit for EN.600.320 or EN.600.421, but not both. EN.600.120 AND EN.600.226

Instructor(s): S. Smith
Area: Engineering.

EN.600.323. Data-Intensive Computing. 3.00 Credits.

Data-Intensive Computing is an experiential education course in computing with massive data sets that covers the software, algorithms, and systems used to ingest, store, and analyze. Specific topics include: NoSQL software systems including key/value stores and graph databases, scientific python, array databases, (semi)-external memory array and graph algorithms, extract-transform-load (ETL) processing, spatial indexing, OpenCL GPU code acceleration, and performance management of clusters. The course will utilize the unique computing resources at JHU, including the DataScope (5PB of storage), the GPU cluster (110 TFlops), and the Homewood High Performance Computing Cluster (1600 cores). The entire course will take place in several lengthy lab sessions each week. Course time will be divided roughly into team projects (30%), ad-hoc tasks (50%), presentation (10%) and using collaboration tools for concurrent reading and authoring and interactive self-assessment. [Systems]

Prerequisites: EN.600.320 OR EN.600.420

Instructor(s): R. Burns
Area: Engineering.

EN.600.325. Declarative Methods. 3.00 Credits.

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. [Analysis] Students can only receive credit for EN.600.325 or EN.600.425, not both.

Instructor(s): J. Eisner
Area: Engineering.

EN.600.328. Compilers and Interpreters. 3.00 Credits.

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. [Systems] Co-listed with EN.600.428

Prerequisites: EN.600.120 AND EN.600.226

Instructor(s): P. Froehlich
Area: Engineering.
EN.600.335. Artificial Intelligence. 3.00 Credits.
The course situates the study of Artificial Intelligence (AI) first in the broader context of Philosophy of Mind and Cognitive Psychology and then treats in-depth methods for automated reasoning, automatic problem solvers and planners, knowledge representation mechanisms, game playing, machine learning, and statistical pattern recognition. The class is a recommended for all scientists and engineers with a genuine curiosity about the fundamental obstacles to getting machines to perform tasks such as deduction, learning, and 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. [Applications] Prereq: 600.226, 550.171; Recommended: linear algebra, prob/stat. Students can only receive credit for 600.335 or 600.435, not both.
Prerequisites: EN.600.226[C] AND EN.550.171[C]
Instructor(s): P. Koehn
Area: Engineering.

EN.600.337. Distributed Systems. 3.00 Credits.
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. [Systems] Students may receive credit for EN.600.337 or EN.600.437 but not both.
Prerequisites: Students may receive credit for 600.337 or 600.437, but not both.;
Instructor(s): Y. Amir
Area: Engineering.

EN.600.340. Introduction to Genomic Research. 3.00 Credits.
This course will use a project-based approach to introduce undergraduates to research in computational biology and genomics. During the semester, students will take a series of large data sets, all derived from recent research, and learn all the computational steps required to convert raw data into a polished analysis. Data challenges might include the DNA sequences from a bacterial genome project, the RNA sequences from an experiment to measure gene expression, the DNA from a human microbiome sequencing experiment, and others. Topics may vary from year to year. In addition to computational data analysis, students will learn to do critical reading of the scientific literature by reading high-profile research papers that generated groundbreaking or controversial results. [Applications] Recommended Course Background: Knowledge of the Unix operating system and programming expertise in a language such as Perl or Python.
Instructor(s): S. Salzberg
Area: Engineering.

EN.600.344. Computer Network Fundamentals. 3.00 Credits.
Topics covered will include application layer protocols (e.g. HTTP, FTP, SMTP), transport layer protocols (UDP, TCP), network layer protocols (e.g. IP, ICMP), link layer protocols (e.g. Ethernet) and wireless protocols (e.g. IEEE 802.11). The course will also cover routing protocols such as link state and distance vector, multicast routing, and path vector protocols (e.g. BGP). The class will examine security issues such as firewalls and denial of service attacks. We will also study DNS, NAT, Web caching and CDNs, peer to peer, and protocol tunneling. Finally, we will explore security protocols (e.g. TLS, SSH, IPsec), as well as some basic cryptography necessary to understand these. Grading will be based on hands-on programming assignments, homeworks and two exams. [Systems] Students can only receive credit for EN.600.344 or EN.600.444, not both.
Prerequisites: Prereqs For EN.600.344: EN.600.233/333/433 or EN.600.255/355/455, not both.
Instructor(s): A. DeSimone
Area: Engineering.

EN.600.355. Video Game Design Project. 3.00 Credits.
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. [General] May involve travel to MICA. Junior or senior standing recommended.
Prerequisites: EN.600.255 AND EN.600.256 or permission of instructor
Instructor(s): P. Froehlich
Area: Engineering.

EN.600.357. Computer Graphics. 3.00 Credits.
This course introduces computer graphics techniques and applications, including image processing, rendering, modeling and animation. [Applications] Students may receive credit for EN.600.357 or EN.600.457, but not both. No Audits.
Prerequisites: EN.600.120 AND EN.600.226
Instructor(s): M. Kazhdan
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.363. Introduction To Algorithms. 3.00 Credits.
This course concentrates on the design of algorithms and the rigorous analysis of their efficiency. topics include the basic definitions of algorithmic complexity (worst case, average case); basic tools such as dynamic programming, sorting, searching, and selection; advanced data structures and their applications (such as union-find); graph algorithms and searching techniques such as minimum spanning trees, depth-first search, shortest paths, design of online algorithms and competitive analysis. [Analysis] Students may receive credit for EN.600.363 or EN.600.463, but not both.
Prerequisites: EN.600.226 and EN.550.171 or permission;Students may receive credit for EN.600.363 or EN.600.463, but not both.
Instructor(s): V. Braverman
Area: Engineering, Quantitative and Mathematical Sciences.
EN.600.365. Knowledge Discovery from Text. 3.00 Credits.
The world is full of text: webpages, emails, newspaper articles, tweets, medical records, and so on. The purpose of text is for people to convey knowledge to other people. This course focuses on how computers analyze large, potentially streaming, text collections to automatically discover knowledge on their own (and to help people better find it themselves). Lectures and assignments will cover relevant topics in automatic classification (applied machine learning), linguistics, high-performance computing, and systems engineering, working with software systems for automatic question answering, populating knowledge bases, and aggregate analysis of social media such as Twitter. [Applications]
Prerequisites: EN.600.120 AND EN.600.226
Instructor(s): B. Van Durme; T. Lippincott
Area: Engineering.

EN.600.371. Software Tools Practicum. 3.00 Credits.
This course will survey the wide range of tools, frameworks and packages that form a foundation for much of the modern Internet. Each week, a group of students will present a particular software package to the class, exploring the architectural, historical and competitive aspects. The weekly lab work and assignments will focus on testing the software and understanding its contributions to providing the services that support the Internet. The software will be chosen from a list of popular packages like Hudson, Drupal, Docker, WordPress, Node.js, jQuery, Django, Ruby on Rails, Git, Coffeescript, Ember, Angular, MongoDB, Couch, and others. Students will select one package and work with others to explore the packages, learn how they work and then summarize this knowledge by creating a presentation for the class. [General] CS majors only.
Prerequisites: EN.600.120 AND EN.600.226
Instructor(s): P. Wayner
Area: Engineering.

EN.600.402. Digital Health and Biomedical Informatics. 1.00 Credit.
Advances in technology are driving a change in medicine, from personalized medicine to population health. Computers and information technology will be critical to this transition. We shall discuss some of the coming changes in terms of computer technology, including computer-based patient records, clinical practice guidelines, and region-wide health information exchanges. We will discuss the underlying technologies driving these developments - databases and warehouses, controlled vocabularies, and decision support.
Instructor(s): H. Lehmann
Area: Engineering.

EN.600.411. Computer Science Innovation & Entrepreneurship II. 3.00 Credits.
This course is the second half of a two-course sequence and is a continuation of course EN.660.410.01, CS Innovation and Entrepreneurship, offered by the Center for Leadership Education (CLE). In this sequel course the student groups, directed by CS faculty, will implement the business idea which was developed in the first course and will present the implementations and business plans to an outside panel made up of practitioners, industry representatives, and venture capitalists. [General]
Prerequisites: EN.660.410 AND ( EN.660.321 OR EN.660.421 )
Instructor(s): A. Dahbura; L. Aronhime
Area: Engineering.

EN.600.415. Databases. 3.00 Credits.
Graduate level version of EN.600.315 [Systems]. Students may receive credit for EN.600.315 or EN.600.415, but not both. Recommended Course Background: EN.600.226
Instructor(s): D. Yarowsky
Area: Engineering.

EN.600.416. Database Systems. 3.00 Credits.
Similar material as EN.600.316, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for EN.600.316 or EN.600.416, but not both. Recommended Course Background: EN.600.120 and EN.600.226
Prerequisites: Students may receive credit for EN.600.316 or EN.600.416, but not both.
Instructor(s): Y. Ahmad
Area: Engineering.

EN.600.418. Operating Systems. 3.00 Credits.
Similar material as EN.600.318, covered in more depth. Intended for upper-level undergraduates and graduate students. Students may receive credit for EN.600.318 or EN.600.418, but not both. [Systems]
Prerequisites: Students may receive credit for EN.600.318 or EN.600.418 but not both.
Instructor(s): P. Froehlich
Area: Engineering.

EN.600.420. Parallel Programming. 3.00 Credits.
Graduate level version of EN.600.320. Students may receive credit for EN.600.320 or EN.600.420, but not both. Recommended Course Background: EN.600.120 or equivalent.
Instructor(s): R. Burns
Area: Engineering.

EN.600.421. Object Oriented Software Engineering. 3.00 Credits.
Graduate level version of EN.600.321 [Systems or Applications]. Students may receive credit for EN.600.321 or EN.600.421, but not both. Recommended Course Background: EN.600.226 and EN.600.120
Instructor(s): S. Smith
Area: Engineering.

EN.600.423. Data-Intensive Computing. 3.00 Credits.
Graduate student version of EN.600.323. [Systems] Students may receive credit for EN.600.323 or EN.600.423, but not both.
Prerequisites: EN.600.320 OR EN.600.420
Instructor(s): R. Burns
Area: Engineering.

EN.600.424. Network Security. 3.00 Credits.
This course focuses on communication security in computer systems and networks. The course is intended to provide students with an introduction to the field of network security. The course covers network security services such as authentication and access control, integrity and confidentiality of data, firewalls and related technologies, Web security and privacy. Course work involves implementing various security techniques. A course project is required. [Systems] Recommended. Course Background: EN.600.120 (or equivalent)
Prerequisites: EN.600.226 AND ( EN.600.344 OR EN.600.444 ) or permission
Instructor(s): S. Nielsen
Area: Engineering.

EN.600.425. Declarative Methods. 3.00 Credits.
Students can only receive credit for EN.600.325 or EN.600.425, not both. Graduate level version of EN.600.325. Recommended Course Background: EN.600.226, EN.600.271, AS.110.107/AS.110.109
Instructor(s): J. Eisner
Area: Engineering.
EN.600.426. Principles of Programming Languages. 3.00 Credits.
Functional, object-oriented, and other language features are studied independent of a particular programming language. Students become familiar with these features by implementing them. Most of the implementations are in the form of small language interpreters. Some type checkers and a small compiler will also be written. The total amount of code written will not be overly large, as the emphasis is on concepts. The ML programming language is the implementation language used. [Analysis] Requires include 600.226. No Freshmen or Sophomores.
Instructor(s): S. Smith
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.428. Compilers & Interpreters. 3.00 Credits.
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. Co-listed with EN.600.328. Students should have knowledge of C/C++ programming and data structures. Graduate version of EN.600.328. Students may receive credit for EN.600.328 or EN.600.428, but not both.
Prerequisites: EN.600.120 AND EN.600.226
Instructor(s): P. Froehlich
Area: Engineering.

EN.600.429. Functional Programming at Work - Haskell and Domain-Specific Languages. 3.00 Credits.
This course studies pure functional programming in the Haskell language and the use of functional programming to build domain specific languages (DSLs): customized, application specific programming languages. This course starts with an introduction to Haskell and its essential ideas of lazy evaluation and type inference. Advanced functional programming topics will include type classes, monads and monad transformers, arrows, templates, dependent types, parser combinators, and multiple parameter type classes. The class will study existing DSLs and DSL implementation techniques, including languages for reactive programming, computer vision, hardware design, computer music, and parallel processing. Students will implement a DSL of their choice in Haskell.
Instructor(s): J. Peterson
Area: Engineering.

EN.600.430. Ontologies and Knowledge Representation. 3.00 Credits.
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 the some of the currently prevalent environments, including Universal Modeling Language, OWL and Protege', RDFS and semantic web applications. [Analysis] Recommended Course Background: EN.600.107
Instructor(s): R. Rynasiewicz
Area: Humanities, Quantitative and Mathematical Sciences.

EN.600.433. Computer Systems. 3.00 Credits.
Graduate version of 600.333. Students may receive credit for 600.333 or 600.433, but not both. [Systems]
Instructor(s): P. Froehlich
Area: Engineering.

EN.600.435. Artificial Intelligence. 3.00 Credits.
The course situates the study of Artificial Intelligence (AI) first in the broader context of Philosophy of Mind and Cognitive Psychology and then treats in-depth methods for automated reasoning, automatic problem solvers and planners, knowledge representation mechanisms, game playing, machine learning, and statistical pattern recognition. The class is a recommended for all scientists and engineers with a genuine curiosity about the fundamental obstacles to getting machines to perform tasks such as deduction, learning, and 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. [Applications] Prereq: 600.226; Recommended: linear algebra, prob/stat. Students can only receive credit for 600.335 or 600.435, not both.
Prerequisites: Have not taken EN.600.335;EN.600.226
Instructor(s): P. Koehn
Area: Engineering.

EN.600.436. Algorithms for Sensor-Based Robotics. 3.00 Credits.
This course surveys the development of robotic systems for navigating in an environment from an algorithmic perspective. It will cover basic kinematics, configuration space concepts, motion planning, and localization and mapping. It will describe these concepts in the context of the ROS software system, and will present examples relevant to mobile platforms, manipulation, robotics surgery, and human-machine systems. [Analysis] Formerly EN.600.336. Students may receive credit for only one of EN.600.336, EN.600.436 and EN.600.636.
Prerequisites: EN.600.226 and Linear Algebra and Probability; Students may receive credit for only one of EN.600.336, EN.600.436 and EN.600.636.
Instructor(s): S. Leonard
Area: Engineering.

EN.600.437. Distributed Systems. 3.00 Credits.
Graduate version of 600.337 Systems. Students may receive credit for 600.337 or 600.437 but not both. Recommended Course Background: EN.600.120, EN.600.226
Instructor(s): Y. Amir
Area: Engineering.

EN.600.438. Computational Genomics: Data Analysis. 3.00 Credits.
Genomic data has the potential to reveal causes of disease, novel drug targets, and relationships among genes and pathways in our cells. However, identifying meaningful patterns from high-dimensional genomic data has required development of new computational tools. This course will cover current approaches in computational analysis of genomic data with a focus on statistical methods and machine learning. Topics will include disease association, prediction tasks, clustering and dimensionality reduction, data integration, and network reconstruction. There will be some programming and a project component. [Applications] Recommended Course Background: EN.600.226 or other programming experience, probability and statistics, linear algebra or calculus. Students may receive credit for EN.600.438 or EN.600.638, but not both.
Prerequisites: Students may receive credit for EN.600.438 or EN.600.638, but not both.
Instructor(s): A. Battle
Area: Engineering.
EN.600.439. Computational Genomics. 3.00 Credits.
Your genome is the blueprint for the molecules in your body. It’s also a string of letters (A, C, G and T) about 3 billion letters long. How does this string give rise to you? Your heart, your brain, your health? This, broadly speaking, is what genomics research is about. This course will familiarize you with a breadth of topics from the field of computational genomics. The emphasis is on current research problems, real-world genomics data, and efficient software implementations for analyzing data. Topics will include: string matching, sequence alignment and indexing, assembly, and sequence models. Course will involve significant programming projects. [Applications]
Prerequisites: Students may receive credit for 600.439 or 600.639, but not both.
Prerequisite: EN.600.120 AND EN.600.226
Instructor(s): B. Langmead
Area: Engineering.

EN.600.441. Machine Learning for Genomic Data - Trends and Applications. 3.00 Credits.
Genomic data is becoming available in large quantities, but understanding how genetics contributes to human disease and other traits remains a major challenge. Machine learning approaches allow us to automatically analyze and combine genomic data, build predictive models, and identify genetic elements important to disease and cellular processes. This course will cover uses of machine learning in diverse genomic applications. Students will present and discuss current literature. Topics include predicting disease risk from genomic data, integrating diverse genomic data types, gene network reconstruction, and other topics guided by student interest. The course will include a project component with the opportunity to explore publicly available genomic data. Recommended course background: coursework in data mining, machine learning. [Applications] Students may receive credit for 600.441 or 600.641, but not both.
Prerequisites: Students may receive credit for 600.441 or 600.641, but not both.
Instructor(s): A. Battle
Area: Engineering.

EN.600.442. Modern Cryptography. 3.00 Credits.
Modern Cryptography includes seemingly paradoxical notions such as communicating privately without a shared secret, proving things without leaking knowledge, and computing on encrypted data. In this challenging but rewarding course we will start from the basics of private and public key cryptography and go all the way up to advanced notions such as zero-knowledge proofs, functional encryption and program obfuscation. The class will focus on rigorous proofs and require mathematical maturity. [Analysis] Recommended Course Background: EN.600.363/463, EN.600.271/471 and EN.550.171 or equiv.
Instructor(s): A. Jain
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.443. Security & Privacy in Computing. 3.00 Credits.
Lecture topics will include computer security, network security, basic cryptography, system design methodology, and privacy. There will be a heavy work load, including written homework, programming assignments, exams and a comprehensive final. The class will also include a semester-long project that will be done in teams and will include a presentation by each group to the class. [Applications] Recommended Course Background: A basic course in operating systems and networking, or permission of instructor.
Instructor(s): A. Rubin
Area: Engineering.

EN.600.444. Computer Networks. 4.00 Credits.
This course considers intersystem communications issues. Topics covered 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. Recommended Course Background: EN.600.120 and EN.600.233.
Students can only receive credit for EN.600.344 or EN.600.444, not both.
Prerequisites: Students can only receive credit for EN.600.344 or EN.600.444, not both.
Instructor(s): M. Green
Area: Engineering.

EN.600.445. Computer Integrated Surgery I. 4.00 Credits.
This course focuses on computer-based techniques, systems, and applications exploiting quantitative information from medical images and sensors to assist clinicians in all phases of treatment from diagnosis to preoperative planning, execution, and follow-up. It emphasizes the relationship between problem definition, computer-based technology, and clinical application and includes a number of guest lectures given by surgeons and other experts on requirements and opportunities in particular clinical areas. Required Course Background: AS.110.201 or permission of instructor. Recommended Course Background: EN.600.120, EN.600.457, EN.600.461, image processing.
Prerequisites: Students may receive credit for 600.445 or 600.645, but not both.
Instructor(s): R. Taylor
Area: Engineering.

EN.600.446. Computer Integrated Surgery II. 3.00 Credits.
This weekly lecture/seminar course addresses similar material to EN.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 EN.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 EN.600.452. Students may also take this course as EN.600.646. The only difference between EN.600.446 and EN.600.646 is the level of project undertaken. Typically, EN.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. [Applications] Students may receive credit for EN.600.446 or EN.600.646, but not both.
Prerequisites: Prereg for EN.600.446: EN.600.445 or EN.600.645 or permission
Instructor(s): R. Taylor
Area: Engineering.
EN.600.450. Network Embedded Systems & Sensor Networks. 3.00 Credits.
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. Recommended Course Background: EN.600.226, EN.600.120, and EN.600.344/EN.600.444
Instructor(s): M. Chang
Area: Engineering.

EN.600.451. Introduction to Bitcoin and Other Crypto-currencies. 3.00 Credits.
This course covers the basics of Bitcoin and the underlying technologies driving it. The course is intended for students interested in the cryptographic techniques devised to make digital currencies and payment systems secure. Topics include Bitcoin transactions, the blockchain, mining, and decentralized consensus. The course will include a brief introduction to public-key cryptography, digital signatures, hash functions, proof of work/space, multisignatures, and elliptic curve cryptography. The course concludes with an overview of the Bitcoin scripting language and Bitcoin 2.0 platforms. [Systems] Recommended Course Background: EN.600.344/444 (Computer Networks) and EN.550.171 (Discrete Math)
Prerequisites: EN.600.226
Instructor(s): G. Atienese
Area: Engineering.

EN.600.452. Seminar: Computer Integrated Surgery II. 1.00 Credit.
Students may receive credit for EN.600.446 or EN.600.452, but not both. Lecture only version of EN.600.446 (no project). Recommended Course Background: EN.600.445 or instructor permission required.
Instructor(s): R. Taylor
Area: Engineering.

EN.600.454. Practical Cryptographic Systems. 3.00 Credits.
This semester-long course will teach systems and cryptographic design principles by example: by studying and identifying flaws in widely-deployed cryptographic products and protocols. Our focus will be on the techniques used in practical security systems, the mistakes that lead to failure, and the approaches that might have avoided the problem. We will place a particular emphasis on the techniques of provable security and the feasibility of reverse-engineering undocumented cryptographic systems. [Systems]
Instructor(s): M. Green
Area: Engineering.

EN.600.457. Computer Graphics. 3.00 Credits.
This course introduces computer graphics techniques and applications, including image processing, rendering, modeling and animation. [Applications] Permission of instructor is required for students not satisfying a pre-requisite. Students may receive credit for EN.600.357 or EN.600.457, but not both.
Prerequisites: Students may receive credit for EN.600.357 or EN.600.457, but not both. EN.600.120 AND EN.600.226
Instructor(s): M. Kazhdan
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.459. Computational Geometry. 3.00 Credits.
This course will provide an introduction to computational geometry. It will cover a number of topics in two- and three-dimensions, including polygon triangulations and partitions, convex hulls, Delaunay and Voronoi diagrams, arrangements, and spatial queries. Time-permitting, we will also look at Kd-trees, general BSP-trees, and quadtrees. [Analysis] Recommended Course Background: EN.600.120 AND EN.600.226 AND (EN.600.363 OR EN.600.463).
Prerequisites: Students may receive credit for EN.600.459 or EN.600.659, but not both.
Instructor(s): M. Kazhdan
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.460. Software Vulnerability Analysis. 3.00 Credits.
This course will examine vulnerabilities in C source, stack overflows, writing shell code, etc. Also, vulnerabilities in web applications: SQL Injection, cookies, as well as vulnerabilities in C binary fuzzing, and exploit development without source among other topics. Co-listed with EN.650.460
Instructor(s): S. Checkoway
Area: Engineering.

EN.600.461. Computer Vision. 3.00 Credits.
This course gives an overview of fundamental methods in computer vision from a computational perspective. Methods studied include: camera systems and their modelling, computation of 3-D geometry from binocular stereo, motion, and photometric stereo; and object recognition. Edge detection and color perception are covered as well. Elements of machine vision and biological vision are also included. Students may receive credit for at most one of EN.600.361 or EN.600.461 or EN.600.661. [Applications] Prerequisites (soft): intro programming, linear algebra, and prob/stat.
Prerequisites: If you have completed EN.600.361 OR EN.600.661 you cannot enroll in EN.600.461.
Instructor(s): A. Reiter
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.463. Algorithms I. 3.00 Credits.
This course concentrates on the design of algorithms and the rigorous analysis of their efficiency. topics include the basic definitions of algorithmic complexity (worst case, average case); basic tools such as dynamic programming, sorting, searching, and selection; advanced data structures and their applications (such as union-find); graph algorithms and searching techniques such as minimum spanning trees, depth-first search, shortest paths, design of online algorithms and competitive analysis. [Analysis] Students may receive credit for EN.600.363 or EN.600.463, but not both.
Instructor(s): V. Braverman
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.464. Randomized and Big Data Algorithms. 3.00 Credits.
The course emphasizes algorithmic design aspects, and how randomization can be a helpful tool. The topics covered include: tail inequalities, linear programming relaxation & randomized rounding, de-randomization, existence proofs, universal hashing, markov chains, metropolis and metropolis-hastings methods, mixing by coupling and by eigenvalues, counting problems, semi-definite programming and rounding, lower bound arguments, and applications of expanders. [Analysis] Recommended Course Background: EN.600.363 or EN.600.463.
Prerequisites: Students may receive credit for 600.464 or 600.664, but not both.
Instructor(s): V. Braverman
Area: Engineering, Quantitative and Mathematical Sciences.
EN.600.465. Natural Language Processing. 4.00 Credits.
This course is an in-depth overview of techniques for processing human language. How should linguistic structure and meaning be represented? What algorithms can recover them from text? And crucially, how can we build statistical models to choose among the many legal answers? The course covers methods for trees (parsing and semantic interpretation), sequences (finite-state transduction such as morphology), and words (sense and phrase induction), with applications to practical engineering tasks such as information retrieval and extraction, text classification, part-of-speech tagging, speech recognition, and machine translation. There are a number of structured but challenging programming assignments. [Applications] Recommended Course Background: EN.600.226
Instructor(s): J. Eisner
Area: Engineering.

EN.600.466. Information Retrieval and Web Agents. 3.00 Credits.
An in-depth, hands-on study of current information retrieval techniques and their application to developing intelligent WWW agents. Topics include a comprehensive study of current document retrieval models, mail/news routing and filtering, document clustering, automatic indexing, query expansion, relevance feedback, user modeling, information visualization and usage pattern analysis. In addition, the course explores the range of additional language processing steps useful for template filling and information extraction from retrieved documents, focusing on recent, primarily statistical methods. The course concludes with a study of current issues in information retrieval and data mining on the World Wide Web. Topics include web robots, spiders, agents and search engines, exploring both their practical implementation and the economic and legal issues surrounding their use. Recommended Course Background: EN.600.226
Instructor(s): D. Yarowsky
Area: Engineering.

EN.600.468. Machine Translation. 3.00 Credits.
Google translate can instantly translate between any pair of over fifty human languages (for instance, from French to English). How does it do that? Why does it make the errors that it does? And how can you build something better? Modern translation systems learn to translate by reading millions of words of already translated text, and this course will show you how they work. The course covers a diverse set of fundamental building blocks from linguistics, machine learning, algorithms, data structures, and formal language theory, along with their application to a real and difficult problem in artificial intelligence. Recommended Course Background: prob/stat, EN.600.226, EN.600.465
Instructor(s): P. Koehn
Area: Engineering.

EN.600.469. Approximation Algorithms. 3.00 Credits.
This course provides an introduction to approximation algorithms. Topics include vertex cover, TSP, Steiner trees, cuts, greedy approach, linear and semi-definite programming, primal-dual method, and randomization. Additional topics will be covered as time permits. There will be a final project. Students may receive credit for EN.600.469 or EN.600.669, but not both. [Analysis]
Instructor(s): M. Dinitz
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.470. Combinatorics & Graph Theory in Computer Science. 3.00 Credits.
This is a graduate level course studying the applications of combinatorics and graph theory in computer science. We will start with some basic combinatorial techniques such as counting and pigeon hole principle, and then move to advanced techniques such as the probabilistic method, spectral graph theory and additive combinatorics. We shall see their applications in various areas in computer science, such as proving lower bounds in computational models, randomized algorithms, coding theory and pseudorandomness. [Analysis] Recommended Course Background: probability theory and linear algebra
Prerequisites: EN.550.171
Instructor(s): X. Li
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.471. Theory of Computation. 3.00 Credits.
This is a graduate-level course studying the theoretical foundations of computer science. Topics covered will be models of computation from automata to Turing machines, computability, complexity theory, randomized algorithms, inapproximability, interactive proof systems and probabilistically checkable proofs. Students may not take both EN.600.271 and EN.600.471, unless one is for an undergrad degree and the other for grad. [Analysis] Recommended Course Background: EN.550.171 or instructor permission.
Instructor(s): X. Li
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.473. Algorithmic Game Theory. 3.00 Credits.
This course provides an introduction to algorithmic game theory: the study of games from the perspective of algorithms and theoretical computer science. There will be a particular focus on games that arise naturally from economic interactions involving computer systems (such as economic interactions between large-scale networks, online advertising markets, etc.), but there will also be broad coverage of games and mechanisms of all sorts. Topics covered will include a) complexity of computing equilibria and algorithms for doing so, b) (in)efficiency of equilibria, and c) algorithmic mechanism design. [Analysis] Students may receive credit for EN.600.473 or EN.600.673, but not both.
Prerequisites: EN.600.363 OR EN.600.463 OR permission.; Students may receive credit for EN.600.473 or EN.600.673, but not both.
Instructor(s): M. Dinitz
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.475. Machine Learning. 3.00 Credits.
Machine learning is subfield of computer science and artificial intelligence, whose goal is to develop computational systems, methods, and algorithms that can learn from data to improve their performance. This course introduces the foundational concepts of modern Machine Learning, including core principles, popular algorithms and modeling platforms. This will include both supervised learning, which includes popular algorithms like SVMs, logistic regression, boosting and deep learning, as well as unsupervised learning frameworks, which include Expectation Maximization and graphical models. Homework assignments include a heavy programming components, requiring students to implement several machine learning algorithms in a common learning framework. Additionally, analytical homework questions will explore various machine learning concepts, building on the pre-requisites that include probability, linear algebra, multi-variate calculus and basic optimization. Students in the course will develop a learning system for a final project. [Analysis or Applications]
Instructor(s): R. Arora
Area: Engineering.
EN.600.476. Machine Learning: Data to Models. 3.00 Credits.
How can robots localize themselves in an environment when navigating? Can we predict which patients are at greatest-risk for complications in the hospital? Which movie should I recommend to this user given his history of likes? Many such big data questions can be answered using the paradigm of probabilistic models in machine learning. These are especially useful when common off-the-shelf algorithms such as support vector machines and k-means fail. You will learn methods for clustering, classification, structured prediction, recommendation and inference. We will use Murphy's book, Machine Learning: a Probabilistic Perspective, as the text for this course. Assignments are solved in groups of size 1-3 students. The class will have 4 interactive sessions during which we brainstorm how to solve example open-ended real-world problems with the tools learnt in class. Students are also required to do a project of their choice within which they experiment with the ideas learnt in class. [Analysis or Applications] Students may receive credit for EN.600.476 or EN.600.676, but not both. Requisites include Intro Prob/Stat, Linear Algebra and Intro Machine Learning as well as strong background in s.
Instructor(s): S. Saria
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.477. Causal Inference. 3.00 Credits.
Big data is not necessarily high quality data. Systematically missing records, unobserved confounders, and selection effects present in many datasets make it harder than ever to answer scientifically meaningful questions. This course will teach mathematical tools to help you reason about causes, effects, and bias sources in data with confidence. We will use graphical causal models, and potential outcomes to formalize what causal effects mean, describe how to express these effects as functions of observed data, and use regression model techniques to estimate them. We will consider techniques for handling missing values, structure learning algorithms for inferring causal directionality from data, and connections between causal inference and reinforcement learning. [Analysis] Pre requisites: familiarity with the R programming language, multivariate calculus, basics of linear algebra and probability. Students may receive credit for EN.600.477 or EN.600.677, but not both.
Prerequisites: Students may receive credit for EN.600.477 or EN.600.677, but not both.
Instructor(s): I. Shpitser
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.479. Representation Learning. 3.00 Credits.
Often the success of a machine learning project depends on the choice of features used. Machine learning has made great progress in training classification, regression and recognition systems when good representations, or features, of input data are available. However, much human effort is spent on designing good features which are usually knowledge-based and engineered by domain experts over years of trial and error. A natural question to ask then is Can we automate the learning of useful features from raw data? Representation learning algorithms such as principal component analysis aim at discovering better representations of inputs by learning transformations of data that disentangle factors of variation in data while retaining most of the information. The success of such data-driven approaches to feature learning depends not only on how much data we can process but also on how well the features that we learn correlate with the underlying unknown labels (semantic content in the data). This course will focus on scalable machine learning approaches for learning representations from large amounts of unlabeled, multi-modal, and heterogeneous data. We will cover topics including deep learning, multi-view learning, dimensionality reduction, similarity-based learning, and spectral learning. Students may receive credit for 600.479 or 600.679 but not both. [Analysis or Applications] Required course background: machine learning or basic probability and linear algebra.
Prerequisites: If you have completed EN.600.679 you may not enroll in EN.600.479.
Instructor(s): R. Arora
Area: Engineering.

EN.600.484. Augmented Reality. 3.00 Credits.
Undergraduate level version of EN.600.684. Students may take EN.600.384 or EN.600.684, but not both.
Prerequisites: Preqs: EN.600.120 AND EN.600.226 AND (AS.110.201 OR EN.550.291)
Instructor(s): N. Navab
Area: Engineering.

EN.600.485. Probabilistic Models of the Visual Cortex. 3.00 Credits.
The course gives an introduction to computational models of the mammalian visual cortex. It covers topics in low-, mid-, and high-level vision. It briefly discusses the relevant evidence from anatomy, electrophysiology, imaging (e.g., fMRI), and psychophysics. It concentrates on mathematical modelling of these phenomena taking into account recent progress in probabilistic models of computer vision and developments in machine learning, such as deep networks. [Applications or Analysis] Recommended Course Background: AS.110.106 or AS.110.108; Programming experience (Python preferred).
Instructor(s): A. Yuille
Area: Quantitative and Mathematical Sciences.
EN.600.488. Foundations of Computational Biology & Bioinformatics II. 3.00 Credits.
This course will introduce probabilistic modeling and information theory applied to biological sequence analysis, focusing on statistical models of protein families, alignment algorithms, and models of evolution. Topics will include probability theory, score matrices, hidden Markov models, maximum likelihood, expectation maximization and dynamic programming algorithms. Homework assignments will require programming in Python. Foundations of Computational Biology I is not a prerequisite. [Analysis] Co-listed with EN.580.488. Recommended Course Background: multivariable calculus and differential equations, at least one prob/stat course, EN.580.221 or equivalent, EN.600.226 or equivalent.
Instructor(s): R. Karchin
Area: Engineering, Natural Sciences.

EN.600.503. Independent Study. 1.00 - 3.00 Credits.
Individual guided study for undergraduate students under the direction of a faculty member in the department. The program of study, including the credit to be assigned, must be worked out in advance between the student and the faculty member involved. Permission required.
Instructor(s): Staff.

EN.600.504. Undergraduate Independent Study. 3.00 Credits.
For undergraduate students. Permission of faculty sponsor is required.
Instructor(s): Staff.

EN.600.507. Undergraduate Research. 1.00 - 3.00 Credits.
Individual research for undergraduates under the direction of a faculty member in the department. The program of research, including the credit to be assigned, must be worked out in advance between the student and the faculty member involved. Permission required.
Instructor(s): Staff.

EN.600.508. Undergraduate Research. 3.00 Credits.
Permission of faculty sponsor is required.
Instructor(s): Staff.

EN.600.509. Computer Science Internship. 1.00 Credit.
Individual work in the field with a learning component, supervised by a faculty member in the department. The program of study and study credit assigned must be worked out in advance between the student and the faculty member involved. Students may not receive credit for work that they are paid to do. As a rule of thumb, 40 hours of work is equivalent to one credit. Permission required.
Instructor(s): Staff.

EN.600.510. Computer Science Internship. 0.00 - 4.00 Credits.
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, which is the limit per semester. Permission of faculty sponsor is required.
Instructor(s): Staff.

EN.600.517. Group Undergraduate Research. 1.00 - 3.00 Credits.
Independent research for undergraduates under the direction of a faculty member in the department. This course has a weekly research group meeting that students are expected to attend. The program of research, including the credit to be assigned, must be worked out in advance between the student and the faculty member involved.
Instructor(s): Staff.

EN.600.519. Senior Honors Thesis. 3.00 Credits.
The student will undertake a substantial independent research project under the supervision of a faculty member, potentially leading to the notation Departmental Honors with Thesis on the final transcript. Students are expected to enroll in both semesters of this course during their senior year. Project proposals must be submitted and accepted in the preceding spring semester (junior year) before registration. Students will present their work publicly before April 1st of senior year. They will also submit a first draft of their project report (thesis document) at that time.Faculty will meet to decide if the thesis will be accepted for honors. Computer science majors only. Students should have a 3.5 GPA in computer science courses at the end of their junior year and permission of faculty sponsor. See EN.600.491 for faculty section numbers.
Instructor(s): Staff.

EN.600.520. Senior Honors Thesis. 0.00 - 3.00 Credits.
For computer science majors only, a continuation of EN.600.519. Recommended Course Background: EN.600.519
Instructor(s): Staff.

EN.600.546. Senior Thesis in CIS. 3.00 Credits.
Prerequisites: EN.600.445 or perm req’d.
Instructor(s): R. Taylor.

EN.600.591. Computer Science Workshop I. 1.00 - 3.00 Credits.
An applications-oriented, computer science project done under the supervision and with the sponsorship of a faculty member in the Department of Computer Science. Computer Science Workshop provides a student with an opportunity to apply theory and concepts of computer science to a significant project of mutual interest to the student and a Computer Science faculty member. Permission to enroll in CSW is granted by the faculty sponsor after his/her approval of a project proposal from the student. Interested students are advised to consult with Computer Science faculty members before preparing a Computer Science Workshop project proposal.
Instructor(s): D. Yarowsky; J. Selinski; P. Froehlich; S. Smith.

EN.600.592. Computer Science Workshop II. 1.00 Credit.
Permission of faculty sponsor is required.
Instructor(s): D. Yarowsky; J. Selinski; P. Froehlich; S. Smith.

EN.600.595. Independent Study-Summer. 3.00 Credits.
Instructor(s): Staff.

EN.600.597. Research-Summer. 3.00 Credits.
Instructor(s): Staff.

EN.600.599. Internship. 1.00 Credit.
Instructor(s): Staff.

EN.600.601. Computer Science Seminar. 1.00 Credit.
Required for all full-time CS PhD students. Recommended for MSE students.
Instructor(s): S. Smith.

EN.600.602. Computer Science Seminar. 1.00 Credit.
Required for all CS PhD students. Strongly recommended for MSE students.
Instructor(s): S. Smith.
EN.600.615. Big Data, Small Languages, Scalable Systems. 3.00 Credits.  
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). Recommended Course Background: EN.600.315/EN.600.415 or equivalent.  
Instructor(s): Y. Ahmad.

EN.600.624. Advanced Topics in Data-Intensive Computing. 3.00 Credits.  
The advent of cloud computing has lead to an explosion of storage system and data analysis software, including NoSQL databases, bulk-synchronous processing, graph computing engines, and stream processing. This course will explore scale-out software architectures for data-processing tasks. It will examine the algorithms and data-structures that underlie scalable systems and look at how hardware and networking trends influence the design and deployment of cloud computing. Recommended Course Background: EN.600.320/420 or permission of instructor.  
[Systems]  
Instructor(s): R. Burns.

EN.600.625. Events Semantics in Theory and Practice. 3.00 Credits.  
This course explores selected topics in the nature of event representations from the perspective of cognitive science, computer science, linguistics, and philosophy. These fields have developed a rich array of scientific theories about the representation of events, and how humans make inferences about them – we investigate how (and if) such theories could be applied to current research topics and tasks in computational semantics such as inference from text, automated summarization, veridicality assessment, and so on. In addition to classic articles dealing with formal semantic theories, the course considers available machine-readable corpora, ontologies, and related resources that bear on event structure, such as WordNet, PropBank, FrameNet, etc. The course is aimed to marry theory with practice: students with either a computational or linguistic background are encouraged to participate.  
[Applications]  
Instructor(s): B. Van Durme; K. Rawlins.

EN.600.629. Wireless Networks. 3.00 Credits.  
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), and learn about medium access control protocols, wireless local area networks (IEEE 802.11), and wireless mobile networks and applications, including cellular networks (cellular concept, channel reuse, capacity limits, and cellular systems), fourth generation systems, long term evolution (LTE), mobile applications, and mobile IP.  
[Systems] Recommended Course Background: EN.600.344/EN.600.444.  
Instructor(s): M. Zadeh.

EN.600.636. Algorithms for Sensor-Based Robotics. 3.00 Credits.  
Graduate level version of EN.600.436 (see description above). Formerly EN.600.436. Students may receive credit for only one of EN.600.336, EN.600.436 or EN.600.636. Recommended Course Background: EN.600.226, AS.110.106, and Prob/Stat.  
Prerequisites: Students may receive credit for only one of EN.600.336, EN.600.436 and EN.600.636.  
Instructor(s): S. Leonard.

EN.600.638. Computational Genomics: Data Analysis. 3.00 Credits.  
Graduate level version of EN.600.438.  
[Applications] Recommended Course Background: EN.600.226 or other programming experience, probability and statistics, linear algebra or calculus. Students may receive credit for EN.600.438 or EN.600.638 but not both.  
Prerequisites: Students may receive credit for EN.600.438 or EN.600.638, but not both.  
Instructor(s): A. Battle  
Area: Engineering.

EN.600.639. Computational Genomics. 3.00 Credits.  
Graduate version of EN.600.439  
[Applications] Students may earn credit for EN.600.439 or EN.600.639, but not both. Recommended Course Background: EN.600.120 and EN.600.226  
Instructor(s): B. Langmead  
Area: Engineering.

EN.600.640. Frontiers of Sequencing Data Analysis. 3.00 Credits.  
Public archives now contain petabytes of valuable but hard-to-analyze DNA sequencing data. Analyzing even small datasets is complicated by sequencing errors, differences between individuals, and the fragmentary nature of the the sequencing reads. In this course, we study recent algorithms and methods that seek to make sense of DNA sequencing datasets from small to very large. Topics covered will vary from year to year, but could include RNA sequencing data analysis, other functional genomics data analysis, metagenomics analysis, data compression, indexing, applications of streaming algorithms and sketch data structures, assembly, etc. There will be homework assignments and a course project.  
[Applications]  
Instructor(s): B. Langmead.

EN.600.641. Advanced Topics in Genomic Data Analysis. 3.00 Credits.  
Genomic data is becoming available in large quantities, but understanding how genetics contributes to human disease and other traits remains a major challenge. Machine learning approaches allow us to automatically analyze and combine genomic data, build predictive models, and identify genetic elements important to disease and cellular processes. This course will cover uses of machine learning in diverse genomic applications. Students will present and discuss current literature. Topics include predicting disease risk from genomic data, integrating diverse genomic data types, gene network reconstruction, and other topics guided by student interest. The course will include a project component with the opportunity to explore publicly available genomic data. Recommended course background: coursework in data mining, machine learning.  
[Applications] Students may receive credit for 600.441 or 600.641, but not both.  
Instructor(s): A. Battle.

EN.600.642. Advanced Topics in Cryptography. 3.00 Credits.  
This course will focus on advanced cryptographic topics with an emphasis on open research problems and student presentations.  
Instructor(s): A. Jain.
EN.600.643. Advanced Topics in Computer Security. 3.00 Credits.
Topics will vary from year to year, but will focus mainly on network perimeter protection, host-level protection, authentication technologies, intellectual property protection, formal analysis techniques, intrusion detection and similarly advanced subjects. Emphasis in this course is on understanding how security issues impact real systems, while maintaining an appreciation for grounding the work in fundamental science. Students will study and present various advanced research papers to the class. There will be homework assignments and a course project.
Instructor(s): A. Rubin.

EN.600.645. Computer Integrated Surgery I. 4.00 Credits.
Graduate version of EN.600.445 (see description). Students may earn credit for EN.600.445 or EN.600.645, but not both. Prerequisites: data structures, and linear algebra, or permission; Recommended Course Background: intermediate programming in C/C++, EN.600.457, EN.600.461, image processing. [Applications]
Instructor(s): R. Taylor
Area: Engineering.

EN.600.646. Computer Integrated Surgery II. 3.00 Credits.
Students may receive credit for EN.600.446 or EN.600.646, but not both. Advanced version of EN.600.446. [Applications]
Prerequisites: EN.600.445 OR EN.600.645 OR PERMISSION OF INSTRUCTOR
Instructor(s): R. Taylor.

EN.600.649. Computational Genomics: Applied Comparative Genomics. 3.00 Credits.
The goal of this course is to study the leading computational and quantitative approaches for comparing and analyzing genomes starting from raw sequencing data. The course will focus on human genomics and human medical applications, but the techniques will be broadly applicable across the tree of life. The topics will include genome assembly & comparative genomics, variant identification & analysis, gene expression & regulation, personal genome analysis, and cancer genomics. The grading will be based on assignments, a midterm & final exam, class presentations, and a significant class project. [Applications]
Expected course background: familiarity with UNIX scripting and/or programming.
Instructor(s): M. Schatz.

EN.600.657. Advanced Topics for Computer Graphics. 3.00 Credits.
This course presents advanced methodologies and their applications to computer graphics. Recommended Course Background: any computer science courses above the 400-level in computer graphics & linear algebra; or permission of instructor.
Instructor(s): M. Kazhdan.

EN.600.659. Introduction to Computational Geometry. 3.00 Credits.
This course will provide an introduction to computational geometry. It will cover a number of topics in two- and three-dimensions, including polygon triangulations and partitions, convex hulls, Delaunay and Voronoi diagrams, arrangements, and spatial queries. [Analysis] Recommended Course Background: EN.600.363/463
Instructor(s): M. Kazhdan.

EN.600.660. FFT in Graphics & Vision. 3.00 Credits.
In this course, we will study the Fourier Transform from the perspective of representation theory. We will begin by considering the standard transform defined by the commutative group of rotations in 2D and translations in two- and three-dimensions, and will proceed to the Fourier Transform of the non-commutative group of 3D rotations. Subjects covered will include correlation of images, shape matching, computation of invariances, and symmetry detection. Recommended Course Background: AS.110.201 and comfort with mathematical derivations.
Instructor(s): M. Kazhdan.

EN.600.661. Computer Vision. 3.00 Credits.
Graduate version of EN.600.461. Students may receive credit for at most one of EN.600.361 or EN.600.461 or EN.600.661. [Applications]
Prerequisites (soft): intro programming, linear algebra, and prob/stat.
Prerequisites: If you have completed EN.600.361 OR EN.600.461 you cannot enroll for EN.600.661.
Instructor(s): A. Reiter.

EN.600.662. Topics in Illumination and Reflectance Modeling for Computer Vision and Medical Imaging Applications. 3.00 Credits.
The vast majority of all imagery on which computer vision is performed starts with a source of illumination in conjunction with a material reflectance property. Having a rigorous understanding of these phenomena is important for most students who want to be involved with further research in computer vision and computer integrated surgery, particularly for experimentation and development of new systems. This short course is for individuals who have already taken Computer Vision, and want to delve deeper into underlying physical modeling of light illumination, reflection, colorimetry, polarization and even sensor fusion of images taken at different Wavelengths.
Prerequisites: EN.600.361 OR EN.600.461 or Instructor permission required.
Instructor(s): L. Wolff.

EN.600.664. Randomized and Big Data Algorithms. 3.00 Credits.
Graduate level version of EN.600.464. Recommended Course Background: EN.600.363 or EN.600.463. Students may receive credit for EN.600.464 or EN.600.664, but not both.
Instructor(s): V. Braverman
Area: Engineering.

EN.600.666. Information Extraction. 3.00 Credits.
Introduction to statistical methods of speech recognition (automatic transcription of speech) and understanding. The course is a natural continuation of EN.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. Co-listed as EN.520.666. Recommended Course Background: EN.600.120 and EN.550.310 or equivalent, expertise in C or C++ programming
Instructor(s): S. Khudanpur.
EN.600.667. Advanced Distributed Systems & Networks. 3.00 Credits.
The course explores the state of the art in distributed systems, networks and Internet research and practice, trying to see what it would take to push the envelop a step further. The course is conducted as a discussion group, where the professor and students brainstorm and pick interesting semester-long projects with high potential future impact. Example areas include robust scalable infrastructure (distributed datacenters, cloud networking, scada systems), real-time performance (remote surgery, trading systems), hybrid networks (mesh networks, 3-4G/Wifi/Bluetooth). Students should feel free to bring their own topics of interest and ideas. Recommended Course Background: a systems course (distributed systems, operating systems, computer networks, parallel programming) or permission of instructor.
Instructor(s): Y. Amir.

EN.600.668. Advanced Topics in Software Security. 3.00 Credits.
Topics vary but mainly focus on recent advances in exploitation techniques and defenses for software including software running on embedded systems software, browsers, and nontraditional devices such as microcontrollers in PCs. Recommended Course Background: EN.600.460 or EN.650.442 or permission of instructor.
Instructor(s): M. Dinitz.

EN.600.669. Approximation Algorithms. 3.00 Credits.
Graduate version of EN 600.469. Students may receive credit for EN.600.469 or EN.600.669, but not both.
Prerequisites: EN.600.363 OR EN.600.463 OR permission of instructor.
Instructor(s): S. Checkoway.

EN.600.670. Pseudorandomness and Combinatorial Constructions. 3.00 Credits.
Randomness is very useful in almost all areas of computer science, such as algorithms, distributed computing and cryptography. However, computers generally do not have access to truly uniform random bits. To deal with this, we rely on various pseudorandom objects to reduce either the quantity or the quality of the random bits needed. In this course, we will develop provably good pseudorandom objects for a variety of tasks. We will frequently require explicit combinatorial constructions. That is, we will want to efficiently and deterministically construct such objects. Along the way, we will also explore the close connections of such objects to many other areas in computer science and mathematics, such as graph theory, coding theory, complexity theory and arithmetic combinatorics. [Analysis] Recommended Course Background: EN.600.271/417, EN.600.363/463 and probability.
Instructor(s): X. Li.

EN.600.675. Statistical Machine Learning. 3.00 Credits.
This is a second graduate level course in machine learning. It will provide a formal and an in-depth coverage of topics at the interface of statistical theory and computational sciences. We will revisit popular machine learning algorithms and understand their performance in terms of the size of the data (sample complexity), memory needed (space complexity), as well as the overall computational runtime (computation or iteration complexity). We will cover topics including nonparametric methods, kernel methods, online learning and reinforcement learning, as well as introduce students to current topics in large-scale machine-learning and randomized projections. Topics will vary from year-to-year but the general focus would be on combining methodology with theoretical and computational foundations. [Analysis or Applications]
Prerequisites: EN.600.475 OR EN.600.476 OR EN.600.676 OR permission of the instructor.
Instructor(s): R. Arora.

EN.600.676. Machine Learning: Data to Models. 3.00 Credits.
Students in the class will be asked to do assignments in Matlab. Matlab is typically easy to pick up if one is already familiar with a different programming language. Students are expected to be mathematically mature. One should have taken at least an introductory course in probability theory and linear algebra. Though not required, exposure to optimization or machine learning is recommended. Proficiency in at least one programming language is expected. When in doubt, send the instructor a copy of your transcript to see if the class is appropriate for you. Also, sit through the first few sessions and first homework to get a sense of fit. Requisites include Intro Prob/Stat, Linear Algebra and Intro Machine Learning as well as strong background in s.
Instructor(s): S. Saria.

EN.600.677. Causal Inference. 3.00 Credits.
Advanced graduate version of EN.600.477. [Analysis] Pre-requisites: familiarity with the R programming language, multivariate calculus, basics of linear algebra and probability. Students may receive credit for EN.600.477 or EN.600.677, but not both.
Prerequisites: Students may receive credit for EN.600.477 or EN.600.677, but not both.
Instructor(s): I. Shpitser.
Area: Engineering, Quantitative and Mathematical Sciences.

EN.600.678. Advanced Topics in Causal Inference. 3.00 Credits.
This course will cover advanced topics on all areas of causal inference, including learning causal effects, path-specific effects, and optimal policies from data featuring biases induced by missing data, confounders, selection, and measurement error, techniques for generalizing findings to different populations, complex probabilistic models relevant for causal inference applications, learning causal structure from data, and inference under interference and network effects. The course will feature a final project which would involve either an applied data analysis problem (with a causal inference flavor), a literature review, or theoretical work. [Analysis] Pre-requisite: EN.600.477/677 or permission.
Prerequisites: EN.600.477 OR EN.600.677.
Instructor(s): I. Shpitser.

EN.600.679. Representation Learning. 3.00 Credits.
Graduate level version of EN.600.477. Students may receive credit for 600.477 or 600.679 but not both. [Analysis or Applications] Required course background: machine learning or basic probability and linear algebra. Co-listed with EN.600.479
Prerequisites: If you have completed EN.600.479 you may not enroll in EN.600.679.
Instructor(s): R. Arora.

EN.600.682. Deep Learning for Image Understanding. 3.00 Credits.
This course discusses advanced topics on the recent progresses using deep learning, specifically deep convolutional neural networks in computer vision and medical image analysis. Topics will be selected from most recent papers from CVPR/ICCV/ArXiv/NIPS/MICCAI, with the core focus on object/scene recognition, object detection, domain transfer learning and computer-aided diagnosis. This course is targeted toward graduate students who are interested in mastering the understanding of the recent massive amount of literature and applying the skills to a course project (with lectures, paper reading, in-class presentation & discussion and a final research project). [Applications]
Prerequisites: Prereq: EN.600.461 OR EN.600.661
Instructor(s): L. Lu.
EN.600.683. Vision as Bayesian Inference. 3.00 Credits.
This is an advanced course on computer vision from a probabilistic and machine learning perspective. It covers techniques such as linear and non-linear filtering, geometry, energy function methods, markov random fields, conditional random fields, graphical models, probabilistic grammars, and deep neural networks. These are illustrated on a set of vision problems ranging from image segmentation, semantic segmentation, depth estimation, object recognition, object parsing, scene parsing, action recognition, and text captioning. [Analysis or Applications] Required course background: calculus, linear algebra (AS.110.201 or equiv.), probability and statistics (AS.550.311 or equiv.), and the ability to program in Python and C++. Background in computer vision (EN.600.461/661) and machine learning (EN.600.475) suggested but not required.
Instructor(s): A. Yuille.

EN.600.684. Augmented Reality. 3.00 Credits.
This course introduces students to the field of Augmented Reality. It reviews its basic definitions, principles and applications. It then focuses on Medical Augmented Reality and its particular requirements. The course also discusses the main issues of calibration, tracking, multi-modal registration, advanced visualization and display technologies. Homework in this course will relate to the mathematical methods used for calibration, tracking and visualization in medical augmented reality. Students may also be asked to read papers and implement various techniques within group projects. Recommended Course Background: EN.600.120, EN.600.226, and AS.110.201. [Applications] Prerequisites: Students may take only EN.600.384 or EN.600.684, not both.
Instructor(s): N. Navab.

EN.600.688. Foundations of Computational Biology & Bioinformatics II. 3.00 Credits.
This course will introduce probabilistic modeling and information theory applied to biological sequence analysis, focusing on statistical models of protein families, alignment algorithms, and models of evolution. Topics will include probability theory, score matrices, hidden Markov models, maximum likelihood, expectation maximization and dynamic programming algorithms. Homework assignments will require programming in Python. Co-listed with EN.580.688. Recommended Course Background: math through linear algebra and differential equations, at least one prob/stat course, EN.580.221 or equivalent, EN.600.226 or equivalent.
Instructor(s): R. Karchin.

EN.600.692. Unsupervised Learning: From Big Data to Low-Dimensional Representations. 3.00 Credits.
In the era of data deluge, the development of methods for discovering structure in high-dimensional data is becoming increasingly important. This course will cover state-of-the-art methods from algebraic geometry, sparse and low-rank representations, and statistical learning for modeling and clustering high-dimensional data. The first part of the course will cover methods for modeling data with a single low-dimensional subspace, such as PCA, Robust PCA, Kernel PCA, and manifold learning techniques. The second part of the course will cover methods for modeling data with multiple subspaces, such as algebraic, statistical, sparse and low-rank subspace clustering techniques. The third part of the course will cover applications of these methods in image processing, computer vision, and biomedical imaging. Requisites include Linear Algebra, Optimization, and prior exposure to Machine I.
Instructor(s): R. Vidal.

EN.600.707. Selected Topics in CS Education. 0.50 Credit.
This course will explore current issues and research in computer science education. Topics will be drawn from literature, news items, and participant experience. Current faculty and students with interests in academic careers are encouraged to attend.
Instructor(s): J. Selinski.

EN.600.716. Selected Topics on Innovative Data Systems. 1.00 Credit.
This weekly reading group will survey and dissect the cutting-edge on innovative data systems research. Topics will encompass methods and abstraction in core systems and data management areas (eg, cloud computing, scalable programming and storage), as well as use-cases and war stories from industry, and science and engineering applications. View on web: Our semester schedule is posted at damsel.cs.jhu.edu/blockparty
Instructor(s): Y. Ahmad.

EN.600.726. Selected Topics in Programming Languages. 1.00 Credit.
This seminar course covers recent developments in the foundations of programming language design and implementation. Topics covered include type theory, process algebra, higher-order program analysis, and constraint systems. Students will be expected to present papers orally.
Instructor(s): S. Smith.

EN.600.728. Selected Topics in Category Theory. 1.00 Credit.
Students in this course will read a sampling of standard texts in Category Theory (e.g. the books by Awodey, Mac Lane, Pierce, or others) and papers relevant to the research of participants.
Instructor(s): N. Filardo.

EN.600.745. Seminar in Computational Sensing and Robotics. 1.00 Credit.
This weekly seminar will focus on research issues in computer integrated surgery, including subjects such as medical image analysis, statistical modeling, visualization, vision/sensing, surgical planning, 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. Formerly Seminar in Computer Integrated Surgery (CISST)
Instructor(s): P. Kazanzides.

EN.600.746. Seminar: Medical Image Analysis. 1.00 Credit.
This weekly seminar will focus on research issues in medical image analysis, including image segmentation, registration, 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. But individual students will be assigned on a rotating basis to lead the discussion on particular papers or sections of papers. Co-listed with En.520.746.
Instructor(s): J. Prince; R. Taylor.

EN.600.757. Selected Topics in Computer Graphics. 1.00 Credit.
In this course we will review current research in computer graphics. We will meet for an hour once a week and one of the participants will lead the discussion for the week.
Instructor(s): M. Kazhdan.
EN.600.760. CS Theory Seminar. 1.00 Credit.
Seminar series in theoretical computer science. Topics include algorithms, complexity theory, and related areas of TCS. Speakers will be a mix of internal and external researchers, mostly presenting recently published research papers.
Instructor(s): M. Dinitz; V. Braverman; X. Li.

EN.600.764. Seminar in Algorithms. 1.00 Credit.
This course will explore algorithms and theoretical computer science with a focus on algorithms for massive data. Examples of topics include streaming algorithms, approximation algorithms, online algorithms. Students will be encouraged to select a paper and lead a discussion. External speakers will be invited to present current work as well. This course is a good opportunity for motivated students to learn modern algorithmic methods. Recommended Course Background: EN.600.463 or equivalent.
Instructor(s): V. Braverman
Area: Engineering.

EN.600.765. Selected Topics in Natural Language Processing. 1.00 Credit.
A reading group exploring important current research in the field and potentially relevant material from related fields. Enrolled students are expected to present papers and lead discussion.
Prerequisites: EN.600.465 or permission of instructor
Instructor(s): J. Eisner

EN.600.766. Selected Topics in Meaning, Translation and Generation of Text. 1.00 Credit.
A seminar focused on current research and survey articles on computational semantics.
Instructor(s): B. Van Durme; K. Rawlins.

EN.600.767. Selected Topics in Systems Research. 1.00 Credit.
Students will review, present, and discuss current research in computer systems, distributed systems, and computer networks, in the contexts of dependability, performance and scalability.
Instructor(s): Y. Amir.

EN.600.768. Selected Topics in Machine Translation. 1.00 Credit.
Students in this course will review, present, and discuss current research in machine translation. Permission of instructor.
Instructor(s): P. Koehn.

EN.600.770. Selected Topics in Algorithms for Metric Spaces. 1.00 Credit.
This course will focus on algorithms that use, simplify, or exploit metric spaces. Examples of topics we will address include metric embeddings and their applications, algorithms to exploit low dimensionality, graph spanners and sparsifiers, and data structures such as distance oracles and compact routing schemes that allow us to efficiently find distances and paths. This course will mostly be in the form of a reading group, and students will present a paper and lead a discussion. Recommended Course Background: EN.600.363 or EN.600.463 or permission of the instructor.
Instructor(s): M. Dinitz.

EN.600.771. Probability on Strings, Trees, and Sequences. 1.00 Credit.
Many areas of practical computer science focus on discrete data that is sequential or tree-shaped: natural language processing (sentences and their analyses), computational biology (DNA and protein structures), programming languages (computer programs and their interpretations), and compression (sequences of bits). When the data is noisy or ambiguous, decision-making requires probabilistic methods. We will survey formal tools for manipulating sets of strings, trees, sequences, and defining probabilistic models over these sets. Much of the material covers advanced topics at the intersection of formal language and automata theory, probability, and algorithms. Respectively, these three areas will enable us to represent sets, represent uncertainty, and process everything efficiently.
Instructor(s): A. Lopez.

EN.600.772. Selected Topics in Linear Programming and Semi-definite Programming. 1.00 Credit.
Linear programming and semi-definite programming are powerful techniques in convex optimization. They have been used to achieve the best known approximation results for many important combinatorial optimization problems, such as vertex cover, max cut, sparsest cut and MAX-2-SAT. In this course we will together explore the applications of these techniques in computer science, as well as some recent results about their limitations. Time permitting, we may also discuss their connections to the well-known unique games conjecture. This course will be in the form of a reading group, and students are encouraged to select a paper and lead a discussion. Recommended course background: EN.600.463 and EN.600.464.
Instructor(s): X. Li.

EN.600.775. Selected Topics in Machine Learning. 1.00 Credit.
This seminar is recommended for all students interested in data intensive computing research areas (e.g., machine learning, computer vision, natural language processing, speech, computational social science). The meeting format is participatory. Papers that discuss best practices and the state-of-the-art across application areas of machine learning and data intensive computing will be read. Student volunteers lead individual meetings. Faculty and external speakers present from time-to-time. Recommended Course Background: machine learning or permission of the instructor.
Instructor(s): M. Dredze; R. Arora; S. Saria.

EN.600.780. Selected Topics in Computational Genomics. 1.00 Credit.
This course will survey current areas where computer science approaches have been applied to genomics research. Chieftly, the course focuses on DNA sequencing data analysis, including sequence alignment, de novo assembly, error correction, and DNA data compression. Subject matter will be partially guided by student interests. Students will present papers orally.
Instructor(s): B. Langmead.

EN.600.801. PhD Research. 3.00 - 20.00 Credits.
Instructor(s): Staff.

EN.600.802. PhD Research. 3.00 - 20.00 Credits.
Instructor(s): Staff.

EN.600.803. Masters Research. 3.00 - 10.00 Credits.
Permission required. Independent research for masters or pre-dissertation PhD students.
Instructor(s): Staff.

EN.600.804. Masters Research. 1.00 - 3.00 Credits.
Independent research for masters or pre-dissertation PhD students. Permission required.
Instructor(s): Staff.
EN.600.807. Teaching Practicum. 1.00 - 3.00 Credits.
PhD students will gain valuable teaching experience, working closely with their assigned faculty supervisor. Successful completion of this course fulfills the PhD teaching requirement. (grad students) Permission req’d.
Instructor(s): J. Selinski.

EN.600.809. Independent Study. 1.00 - 3.00 Credits.
Permission required. Individual study in an area of mutual interest to a graduate student and a faculty member in the department.
Instructor(s): Staff.

EN.600.810. Graduate Independent Study. 1.00 - 3.00 Credits.
Permission required.
Instructor(s): Staff.

EN.600.891. Independent Study-Summer. 1.00 - 3.00 Credits.
Instructor(s): Staff.

EN.600.895. Research-Summer. 1.00 - 3.00 Credits.
Instructor(s): Staff.

Cross Listed Courses

Cognitive Science
AS.050.814. Research Seminar in Computer Vision.
This course covers advanced topics in computational vision. It discusses and reviews recent progress and technical advances in visual topics such as object recognition, scene understanding, and image parsing.
Instructor(s): A. Yuille.

Physics Astronomy
AS.171.205. Introduction to Practical Data Science: Beautiful Data. 3.00 Credits.
The class will provide an overview of data science, with an introduction to basic statistical principles, databases, fundamentals of algorithms and data structures, followed by practical problems in data analytics. Recommend Course Background: Familiarity with principles of computing.
Instructor(s): S. Szalay
Area: Natural Sciences, Quantitative and Mathematical Sciences.

General Engineering
EN.500.745. Seminar in Computational Sensing and Robotics. 1.00 Credit.
Seminar series in robotics. Topics include: Medical robotics, including computer-integrated surgical systems and image-guided intervention. Sensor based robotics, including computer vision and biomedical image analysis. Algorithmic robotics, robot control and machine learning. Autonomous robotics for monitoring, exploration and manipulation with applications in home, environmental (land, sea, space), and defense areas. Biorobotics and neuromechanics, including devices, algorithms and approaches to robotics inspired by principles in biomechanics and neuroscience. Human-machine systems, including haptic and visual feedback, human perception, cognition and decision making, and human-machine collaborative systems. Cross-listed Mechanical Engineering, Computer Science, Electrical and Computer Engineering, and Biomedical Engineering.
Instructor(s): L. Whitcomb; N. Cowan; P. Kazanzides; R. Etienne Cummings; R. Vidal.

Electrical Computer Engineering
EN.520.434. Modern Biomedical Imaging Instrumentation and Techniques. 3.00 Credits.
An intermediate biomedical imaging course covering modern biomedical imaging instrumentation and techniques as applied to diagnostic radiology and other biomedical applications. It includes recent advances in various biomedical imaging modalities, multi- modality imaging and molecular imaging. The course is team taught by experts in the respective fields and provides a broad based knowledge of modern biomedical imaging to prepare students for graduate studies and research in biomedical imaging. Also, the course will offer tours and practical experience with modern biomedical imaging equipments in clinical and research settings. Co-listed with EN.580.473
Prerequisites: EN.520.432 OR EN.580.472
Instructor(s): B. Tsui.

EN.520.447. Information Theory. 3.00 Credits.
This course will address some basic scientific questions about systems that store or communicate information. Mathematical models will be developed for (1) the process of error-free data compression leading to the notion of entropy, (2) data (e.g. image) compression with slightly degraded reproduction leading to rate-distortion theory and (3) error-free communication of information over noisy channels leading to the notion of channel capacity. It will be shown how these quantitative measures of information have fundamental connections with statistical physics (thermodynamics), computer science (string complexity), economics (optimal portfolios), probability theory (large deviations), and statistics (Fisher information, hypothesis testing).
Instructor(s): S. Khudanpur
Area: Engineering, Quantitative and Mathematical Sciences.

EN.520.701. Current Topics in Language and Speech Processing. 1.00 Credit.
This biweekly seminar will cover a broad range of current research topics in human language technology, including automatic speech recognition, natural language processing and machine translation. The Tuesday seminars will feature distinguished invited speakers, while the Friday seminars will be given by participating students. A minimum of 75% attendance and active participation will be required to earn a passing grade. Grading will be S/U.
Instructor(s): S. Khudanpur.

EN.520.702. Current Topics in Language and Speech Processing. 1.00 Credit.
This biweekly seminar will cover a broad range of current research topics in human language technology, including automatic speech recognition, natural language processing and machine translation. The Tuesday seminars will feature distinguished invited speakers, while the Friday seminars will be given by participating students. A minimum of 75% attendance and active participation will be required to earn a passing grade. Cross-listed with Computer Science. Grading will be S/U.
Instructor(s): S. Khudanpur
Area: Engineering.
EN.530.707. Robot System Programming. 3.00 Credits.
This course seeks to introduce students to open-source software tools that are available today for building complex experimental and fieldable robotic systems. The course is grouped into sections, each of which building on the previous in increasing complexity and specificity: tools and frameworks supporting robotics research, robotics-specific software frameworks, integrating complete robotic systems, and culminates with an independent project of the student’s own design using small mobile robots or other independent robots. Students will need to provide a computer or a virtual-box (with at least a few GB of memory and a several tens of GB of disc space) running Ubuntu (https://www.ubuntu.com or one of its variants such as Xubuntu) and ROS (http://ros.org/). Students should have an understanding of intermediate programming in C/C++ (including data structures and dynamic memory allocation) Familiarity with Linux programming. Familiarity with software version control systems (e.g. subversion, mercurial, git), linear algebra. Recommended Course Background: EN.530.646 Robot Devices, Kinematics, Dynamics, and Control and EN.600.636 Algorithms for Sensor Based Robotics. Students should see the 2016 course homepage http://dscl.lcsr.jhu.edu/MES30707_2016 for more information and to get started with the course. Recommended Course Background: EN.530.646 and EN.600.436.
Instructor(s): R. Karchin.

EN.580.468. The Art of Data Science. 3.00 Credits.
In this course, we will cover the fundamentals of doing data science research, explaining best practices for each step, that collectively comprise an upward spiral. These steps include: (i) asking an interesting question, (ii) determining the degree to which the answer is known, (iii) assessing there currently exists data to likely obtain a satisfactory answer, (iv) exploring the data set, (v) cleaning up the dataset, (vi) formalizing a statistical inquiry, (vii) positing a statistical model which we hope will yield satisfactory answers, (viii) devising a test to assess the answer, (ix) building an estimator to assess the model, (x) checking the model, (xi) reporting the results, (xii) suggesting the next experiment to perform or question to answer to further enhance the model. Note that this course will largely be project based; each student will be expected to complete each of the above steps on some real data of interest to the student. Lectures will be minimal, giving introductory explanations one day, hopefully only part of the time. The rest of the time, we will work independently or in small groups to complete the weekly portion of the overall project. Please come ready to do science! If you don’t have questions that you want answered, you can work in small groups, but each student will need to write the code and reports on their own. Recommended Course Background: No courses are formally required, though students will need to write numerical code (in R, Python, or Julia), and make reports using LaTeX, knitr, or Jupyter notebooks or similar. Instructor(s): J. Vogelstein
Area: Engineering.

EN.580.473. Modern Biomedical Imaging Instrumentation and Techniques. 3.00 Credits.
An intermediate biomedical imaging course covering modern biomedical imaging instrumentation and techniques as applied to diagnostic radiology and other biomedical applications. It includes recent advances in various biomedical imaging modalities, multi-modality imaging and molecular imaging. The course is team taught by experts in the respective fields and provides a broad based knowledge of modern biomedical imaging to prepare students for graduate studies and research in biomedical imaging. Also, the course will offer tours and practical experience with modern biomedical imaging equipment in clinical and research settings. Co-listed with EN.520.434 Recommended course background: EN.520.432 or EN.580.472
Prerequisites: EN.520.432 OR EN.580.472
Instructor(s): B. Tsui
Area: Engineering, Natural Sciences.

EN.580.668. The Art of Data Science. 3.00 Credits.
In this course, we will cover the fundamentals of doing data science research, explaining best practices for each step, that collectively comprise an upward spiral. These steps include: (i) asking an interesting question, (ii) determining the degree to which the answer is known, (iii) assessing there currently exists data to likely obtain a satisfactory answer, (iv) exploring the data set, (v) cleaning up the dataset, (vi) formalizing a statistical inquiry, (vii) positing a statistical model which we hope will yield satisfactory answers, (viii) devising a test to assess the answer, (ix) building an estimator to assess the model, (x) checking the model, (xi) reporting the results, (xii) suggesting the next experiment to perform or question to answer to further enhance the model. Note that this course will largely be project based; each student will be expected to complete each of the above steps on some real data of interest to the student. Lectures will be minimal, giving introductory explanations one day, hopefully only part of the time. The rest of the time, we will work independently or in small groups to complete the weekly portion of the overall project. Please come ready to do science! If you don’t have questions that you want answered, you can work in small groups, but each student will need to write the code and reports on their own. Recommended background: No courses are formally required, though students will need to write numerical code (in R, Python, or Julia), and make reports using LaTeX, knitr, or Jupyter notebooks or similar. Instructor(s): J. Vogelstein
Area: Engineering.

EN.580.688. Foundations of Computational Biology & Bioinformatics II. 3.00 Credits.
This course will introduce probabilistic modeling and information theory applied to biological sequence analysis, focusing on statistical models of protein families, alignment algorithms, and models of evolution. Topics will include probability theory, score matrices, hidden Markov models, maximum likelihood, expectation maximization and dynamic programming algorithms. Homework assignments will require programming in Python. Recommended Course Background: Math through linear algebra and differential equations, EN.580.221 or equivalent, EN.600.226 or equivalent.
Instructor(s): R. Karchin.
EN.580.689. **Computational Personal Genomics.** 3.00 Credits.
What can we learn from the genome sequence of an individual? Genomic technology now makes it possible to generate huge amounts of DNA sequence data for a single individual at a relatively low cost. To make sense of this data, we need to employ sophisticated computational methods to identify genetic variations that influence an individual's health. In this course, we will first review the state of the art in sequencing technology, and discuss how this technology is being applied to study human biology and disease. We will then explore the computational methods used to turn raw sequence data into knowledge. Topics will include genetic variant detection; discovery of chromosomal rearrangements and fusions; methods to measure gene expression from RNA; and measurements of the microbiome living inside our bodies.
Recommended Course Background: EN.600.439/639, EN.600.363/463, EN.600.688, EN.580.688 (any one is sufficient), or permission of the instructor. Course is also open to undergraduate students.
Instructor(s): S. Salzberg
Area: Engineering.

EN.580.694. **Statistical Connectomics.** 3.00 Credits.
This course will cover the basics of an exciting emerging field of statistical connectomics (aka, brain-graphs). It is so new, that we are going to make some of it up in this class! The first week will be introductory lectures that I give. The rest of the semester will be run like a seminar; each week will focus on a different topic. On Tuesdays we will hear about a statistical method that operates on graphs, and on Thursdays we will read about some neuroscience data upon which one could apply these techniques. The final project will consist of implementing a statistical method devised for graphs on a brain-graph problem. Recommended background: coursework in probability, linear algebra, and numerical programming (eg, R, Python, Matlab).
Instructor(s): J. Vogelstein
Area: Engineering.

**Information Security Institute**

EN.650.624. **Advanced Network Security.** 3.00 Credits.
This course focuses on advanced security topics and research in computer networks. It builds on the basic overview of network security covered in previous security courses. Beyond the basics of developing security network communications and applications, this advanced course dives deeper into the theory and practice behind network attack, the growing reality of weaponized zero-day vulnerabilities, and the current state-of-the-art responses. Course work includes reviewing contemporary security research papers, hands-on experiments in defending/attacking networks, and writing analyses.
Prerequisites: EN.650.424 OR EN.600.424 or permission of the instructor.
Instructor(s): S. Nielson
Area: Engineering.