

# Other Departments and Institutes Courses

## About Course Numbers:

Each Carnegie Mellon course number begins with a two-digit prefix that designates the department offering the course (i.e., 76-xxx courses are offered by the Department of English). Although each department maintains its own course numbering practices, typically, the first digit after the prefix indicates the class level: xx-1xx courses are freshmen-level, xx-2xx courses are sophomore level, etc. Depending on the department, xx-6xx courses may be either undergraduate senior-level or graduate-level, and xx-7xx courses and higher are graduate-level. Consult the Schedule of Classes (<https://enr-apps.as.cmu.edu/open/SOC/SOCServlet/>) each semester for course offerings and for any necessary pre-requisites or co-requisites.

## Computational Biology Courses

### 02-201 Programming for Scientists

Fall and Spring: 10 units

Provides a practical introduction to programming for students with little or no prior programming experience who are interested in science. Fundamental scientific algorithms will be introduced, and extensive programming assignments will be based on analytical tasks that might be faced by scientists, such as parsing, simulation, and optimization. Principles of good software engineering will also be stressed. The course will introduce students to the Go programming language, an industry-supported, modern programming language, the syntax of which will be covered in depth. Other assignments will be given in other programming languages such as Python and Java to highlight the commonalities and differences between languages. No prior programming experience is assumed, and no biology background is needed. Analytical skills and mathematical maturity are required. Course not open to CS majors.

### 02-218 Introduction to Computational Medicine

All Semesters: 3 units

This course is an introduction to computational methods relevant to the diagnosis and treatment of human diseases. It is the microcourse version of 02-518, Computational Medicine. The course begins with an introduction to the field of Medicine, and an overview of the primary clinical tasks associated with Computational Medicine (phenotyping; biomarker discovery; predictive modeling). Next, we provide an introduction to several Machine Learning techniques, and how those techniques can be used to perform the clinical tasks. For the remainder of the course, students will be guided through the analysis of a clinical data set to gain experience with these techniques. No prior experience with Medicine, Machine Learning, or computer programming is required. Students will be graded based on quizzes and one homework.

### 02-223 Personalized Medicine: Understanding Your Own Genome

Fall: 9 units

Do you want to know how to discover the tendencies hidden in your genome? Since the first draft of a human genome sequence became available at the start of this century, the cost of genome sequencing has decreased dramatically. Personal genome sequencing will likely become a routine part of medical exams for patients for prognostic and diagnostic purposes. Personal genome information will also play an increasing role in lifestyle choices, as people take into account their own genetic tendencies. Commercial services such as 23andMe have already taken first steps in this direction. Computational methods for mining large-scale genome data are being developed to unravel the genetic basis of diseases and assist doctors in clinics. This course introduces students to biological, computational, and ethical issues concerning use of personal genome information in health maintenance, medical practice, biomedical research, and policymaking. We focus on practical issues, using individual genome sequences (such as that of Nobel prize winner James Watson) and other population-level genome data. Without requiring any background in biology or CS, we begin with an overview of topics from genetics, molecular biology, stats, and machine learning relevant to the modern personal genome era. We then cover scientific issues such as how to discover your genetic ancestry and how to learn from genomes about migration and evolution of human populations. We discuss medical aspects such as how to predict whether you will develop diseases such as diabetes based on your own genome, how to discover disease-causing genetic mutations, and how genetic information can be used to recommend clinical treatments.

### 02-250 Introduction to Computational Biology

Spring: 12 units

This class provides a general introduction to computational tools for biology. The course is divided into two halves. The first half covers computational molecular biology and genomics. It examines important sources of biological data, how they are archived and made available to researchers, and what computational tools are available to use them effectively in research. In the process, it covers basic concepts in statistics, mathematics, and computer science needed to effectively use these resources and understand their results. Specific topics covered include sequence data, searching and alignment, structural data, genome sequencing, genome analysis, genetic variation, gene and protein expression, and biological networks and pathways. The second half covers computational cell biology, including biological modeling and image analysis. It includes homework requiring modification of scripts to perform computational analyses. The modeling component includes computer models of population dynamics, biochemical kinetics, cell pathways, and neuron behavior. The imaging component includes the basics of machine vision, morphological image analysis, image classification and image-derived models. The course is taught under two different numbers. The lectures are the same for both but recitations and examinations are separate. 02-250 is intended primarily for computational biology, computer science, statistics or engineering majors at the undergraduate or graduate level who have had prior experience with computer science or programming. 03-250 is intended primarily for biological sciences or biomedical engineering majors who have had limited prior experience with computer science or programming. Students may not take both 02-250 and 03-250 for credit. Prerequisite: (02-201 or 15-110 or 15-112), or permission of the instructors. Prerequisites: (02-201 or 15-110 or 15-112) and (03-151 or 03-131 or 03-121)

Course Website: <http://www.cbd.cmu.edu/education/undergraduate-courses/introduction-to-computational-biology/>

### 02-251 Great Ideas in Computational Biology

Spring: 12 units

This 12-unit course provides an introduction to many of the great ideas that have formed the foundation for the recent transformation of life sciences into a fully-fledged computational discipline. Extracting biological understanding from both large and small data sets now requires the use and design of novel algorithms, developed in the field of computational biology. This gateway course is intended as a first exposure to computational biology for first-year undergraduates in the School of Computer Science, although it is open to other computationally minded students who are interested in exploring the field. Students will learn fundamental algorithmic and machine learning techniques that are used in modern biological investigations, including algorithms to process string, graph, and image data. They will use these techniques to answer questions such as "How do we reconstruct the sequence of a genome?", "How do we infer evolutionary relationships among many species?", and "How can we predict each gene's biological role?" on biological data. Previous exposure to molecular biology is not required, as the instructors will provide introductory materials as needed. After completion of the course, students will be well equipped to tackle advanced computational challenges in biology.

Prerequisites: (15-112 or 02-201) and (15-151 or 21-127 or 21-128)

### 02-261 Quantitative Cell and Molecular Biology Laboratory

Fall and Spring

This is an introductory laboratory-based course designed to teach basic biological laboratory skills used in exploring the quantitative nature of biological systems and the reasoning required for performing research in computational biology. Over the course of the semester, students will design and perform multiple modern experiments and quantitatively analyze the results of these experiments. During this course students will also have an opportunity to use techniques learned during the course to experimentally answer an open question. Designing the experiments will require students to think critically about the biological context of the experiments as well as the necessary controls to ensure interpretable experimental results. During this course students will gain experience in many aspects of scientific research, including: sequencing DNA, designing and performing PCR for a variety of analyses, maintaining cell cultures, taking brightfield and fluorescent microscopy images, developing methods for automated analysis of cell images, communicating results to peers and colleagues. Course Outline: (1) 3-hour lab per week, (1) 1-hour lecture per week.

Prerequisites: 15-112 or 02-201

**02-262 Computation and Biology Integrated Research Lab**

All Semesters

Modern biological research is heavily interdisciplinary in nature requiring the use of a diverse set of experimental techniques and computational analysis. This course provides students with a modern research experience while training them to communicate and collaborate in an interdisciplinary setting to better prepare them to join the workforce as members of interdisciplinary teams. This will be accomplished by focusing efforts on a real research problem requiring sophisticated experimentation and computation for success. Class time will include both laboratory research time (wet lab and computational) and activities designed to teach and practice communication methods for interdisciplinary teams. Students are expected to have a strong background in biology or computation and an interest in both. Pre-requisites include either (03-117 or 03-124 or 03-343) or (15-112 or equivalent)

**02-317 Algorithms in Nature**

Intermittent: 9 units

Computer systems and biological processes often rely on networks of interacting entities to reach joint decisions, coordinate and respond to inputs. There are many similarities in the goals and strategies of biological and computational systems which suggest that each can learn from the other. These include the distributed nature of the networks (in biology molecules, cells, or organisms often operate without central control), the ability to successfully handle failures and attacks on a subset of the nodes, modularity and the ability to reuse certain components or sub-networks in multiple applications and the use of stochasticity in biology and randomized algorithms in computer science. In this course we will start by discussing classic biologically motivated algorithms including neural networks (inspired by the brain), genetic algorithms (sequence evolution), non-negative matrix factorization (signal processing in the brain), and search optimization (ant colony formation). We will then continue to discuss more recent bi-directional studies that have relied on biological processes to solve routing and synchronization problems, discover Maximal Independent Sets (MIS), and design robust and fault tolerant networks. In the second part of the class students will read and present new research in this area. Students will also work in groups on a final project in which they develop and test a new biologically inspired algorithm. No prior biological knowledge required. Prerequisites: 15-210 and 15-251

Course Website: <http://www.algorithmsinnature.org>**02-319 Genomics and Epigenetics of the Brain**

Fall: 9 units

This course will provide an introduction to genomics, epigenetics, and their application to problems in neuroscience. The rapid advances in single cell sequencing and other genomic technologies are revolutionizing how neuroscience research is conducted, providing tools to study how different cell types in the brain produce behavior and contribute to neurological disorders. Analyzing these powerful new datasets requires a foundation in molecular neuroscience as well as key computational biology techniques. In this course, we will cover the biology of epigenetics, how proteins sitting on DNA orchestrate the regulation of genes. In parallel, programming assignments and a project focusing on the analysis of a primary genomic dataset will teach principles of computational biology and their applications to neuroscience. The course material will also serve to demonstrate important concepts in neuroscience, including the diversity of neural cell types, neural plasticity, the role that epigenetics plays in behavior, and how the brain is influenced by neurological and psychiatric disorders. Although the course focuses on neuroscience, the material is accessible and applicable to a wide range of topics in biology. Prerequisites: (03-121 or 03-151) and (03-220 or 03-221) and (15-121 or 15-112 or 15-110 or 02-201)

**02-331 Modeling Evolution**

Spring: 12 units

Some of the most serious public health problems we face today, from drug-resistant bacteria, to cancer, all arise from a fundamental property of living systems and #8212; their ability to evolve. Since Darwin's theory of natural selection was first proposed, we have begun to understand how heritable differences in reproductive success drive the adaptation of living systems. This makes it intuitive and tempting to view evolution from an optimization perspective. However, genetic drift, phenotypic trade-offs, constraints, and changing environments, are among the many factors that may limit the optimizing force of natural selection. This tug-of-war between selection and drift, between the forces that produce variation in a population, and the forces suppressing this variation, make evolutionary processes much more complex to model and understand than previously thought. The aim of this class is to provide an introduction into the theoretical formalism necessary to understand how biological systems are shaped by the forces and constraints driving evolutionary dynamics.

Prerequisites: 15-112 and 21-241 and (36-218 or 21-325 or 36-225 or 15-259)

**02-402 Computational Biology Seminar**

Fall and Spring: 3 units

This course consists of weekly invited presentations on current computational biology research topics by leading scientists. Attendance is mandatory for a passing grade. You must sign in and attend at least 80% of the seminars. See course website for seminar locations. Some will be at the University of Pittsburgh and some will be at Carnegie Mellon.

**02-403 Special Topics in Bioinformatics and Computational Biology**

Intermittent: 6 units

A decade ago, mass spectrometry (MS) was merely a qualitative research technique allowing the analysis of samples regarding the presence of specific biomolecules. However, as MS has turned quantitative, more sophisticated experiments can be performed, such as the recording of signal transduction kinetics and the analysis of the composition of protein complexes and organelles. This makes MS-based proteomics a powerful method to study spatiotemporal protein dynamics. The development of relative quantification approaches, which generally use <sup>2</sup>H, <sup>13</sup>C or <sup>15</sup>N isotope labels, has especially led to an increase in quantification accuracy and set off numerous new experimental approaches to study protein regulation. In this mini-course, we will cover mass spectrometry principles, discuss classical as well as current primary literature addressing method development and quantitative analysis, and highlight state-of-the-art biological studies that employ MS. A combination of lectures, student presentations, and written exercises will establish a thorough knowledge of current bio-analytical MS approaches.

Prerequisites: (02-250 Min. grade C or 03-250 Min. grade C) and 03-121 Min. grade C

**02-414 String Algorithms**

Intermittent: 12 units

Provides an in-depth look at modern algorithms used to process string data, particularly those relevant to genomics. The course will cover the design and analysis of efficient algorithms for processing enormous amounts of collections of strings. Topics will include string search; inexact matching; string compression; string data structures such as suffix trees, suffix arrays, and searchable compressed indices; and the Borrows-Wheeler transform. Applications of these techniques in genomics will be presented, including genome assembly, transcript assembly, whole-genome alignment, gene expression quantification, read mapping, and search of large sequence databases.

Prerequisites: 21-128 or 15-151 or 15-127

**02-421 Algorithms for Computational Structural Biology**

Intermittent: 12 units

Some of the most interesting and difficult challenges in computational biology and bioinformatics arise from the determination, manipulation, or exploitation of molecular structures. This course will survey these challenges and present a variety of computational methods for addressing them. Topics will include: molecular dynamics simulations, computer-aided drug design, and computer-aided protein design. The course is appropriate for both students with backgrounds in computer science and those in the life sciences.

**02-425 Computational Methods for Proteogenomics and Metabolomics**

Spring: 9 units

Proteomics and metabolomics are the large scale study of proteins and metabolites, respectively. In contrast to genomes, proteomes and metabolomes vary with time and the specific stress or conditions an organism is under. Applications of proteomics and metabolomics include determination of protein and metabolite functions (including in immunology and neurobiology) and discovery of biomarkers for disease. These applications require advanced computational methods to analyze experimental measurements, create models from them, and integrate with information from diverse sources. This course specifically covers computational mass spectrometry, structural proteomics, proteogenomics, metabolomics, genome mining and metagenomics.

Prerequisites: 02-250 or 02-604 or 02-251

**02-450 Automation of Scientific Research**

Spring: 9 units

Automated scientific instruments are used widely in research and engineering. Robots dramatically increase the reproducibility of scientific experiments, and are often cheaper and faster than humans, but are most often used to execute brute-force sweeps over experimental conditions. The result is that many experiments are "wasted" on conditions where the effect could have been predicted. Thus, there is a need for computational techniques capable of selecting the most informative experiments. This course will introduce students to techniques from Artificial Intelligence and Machine Learning for automatically selecting experiments to accelerate the pace of discovery and to reduce the overall cost of research. Real-world applications from Biology, Bioengineering, and Medicine will be studied. Grading will be based on homeworks and two exams. The course is intended to be self-contained, but students should have a basic knowledge of biology, programming, statistics, and machine learning.  
Prerequisites: (10-701 or 10-315) and 15-122

**02-499 Independent Study in Computational Biology**

Fall and Spring

The student will, under the individual guidance of a faculty member, read and digest process papers or a textbook in an advanced area of computational biology not offered by an existing course at Carnegie Mellon. The student will demonstrate their mastery of the material by a combination of one or more of the following: oral discussions with the faculty member; exercises set by the faculty member accompanying the readings; and a written summary synthesizing the material that the student learned. Permission required.

**02-500 Undergraduate Research in Computational Biology**

Fall and Spring

This course is for undergraduate students who wish to do supervised research for academic credit with a Computational Biology faculty member. Interested students should first contact the Professor with whom they would like to work. If there is mutual interest, the Professor will direct you to the Academic Programs Coordinator who will enroll you in the course. 02-250 is a suggested pre-requisite.

Course Website: <https://forms.gle/S1AJX65btkTxwNCw9> (<https://forms.gle/S1AJX65btkTxwNCw9>)

**02-510 Computational Genomics**

Spring: 12 units

Dramatic advances in experimental technology and computational analysis are fundamentally transforming the basic nature and goal of biological research. The emergence of new frontiers in biology, such as evolutionary genomics and systems biology is demanding new methodologies that can confront quantitative issues of substantial computational and mathematical sophistication. From the computational side this course focuses on modern machine learning methodologies for computational problems in molecular biology and genetics, including probabilistic modeling, inference and learning algorithms, data integration, time series analysis, active learning, etc. This course counts as a CSD Applications elective  
Prerequisites: 15-122 Min. grade C and (15-259 or 36-225 or 36-218 or 36-325)

**02-512 Computational Methods for Biological Modeling and Simulation**

Fall: 9 units

This course covers a variety of computational methods important for modeling and simulation of biological systems. It is intended for graduates and advanced undergraduates with either biological or computational backgrounds who are interested in developing computer models and simulations of biological systems. The course will emphasize practical algorithms and algorithm design methods drawn from various disciplines of computer science and applied mathematics that are useful in biological applications. The general topics covered will be models for optimization problems, simulation and sampling, and parameter tuning. Course work will include problem sets with significant programming components and independent or group final projects.

Prerequisites: 15-112 or 02-201 or 15-110

**02-514 String Algorithms**

Fall: 12 units

Provides an in-depth look at modern algorithms used to process string data, particularly those relevant to genomics. The course will cover the design and analysis of efficient algorithms for processing enormous collections of strings. Topics will include string search; inexact matching; string compression; string data structures such as suffix trees, suffix arrays, and searchable compressed indices; and the Burrows-Wheeler transform. Applications of these techniques in biology will be presented, including genome assembly, transcript assembly, whole-genome alignment, gene expression quantification, read mapping, and search of large sequence databases. No knowledge of biology is assumed, and the topics covered will be of use in other fields involving large collections of strings. Programming proficiency is required.

Prerequisite: 15-251

**02-515 Advanced Topics in Computational Genomics**

Spring: 12 units

Research in biology and medicine is undergoing a revolution due to the availability of high-throughput technology for probing various aspects of a cell at a genome-wide scale. The next-generation sequencing technology is allowing researchers to inexpensively generate a large volume of genome sequence data. In combination with various other high-throughput techniques for epigenome, transcriptome, and proteome, we have unprecedented opportunities to answer fundamental questions in cell biology and understand the disease processes with the goal of finding treatments in medicine. The challenge in this new genomic era is to develop computational methods for integrating different data types and extracting complex patterns accurately and efficiently from a large volume of data. This course will discuss computational issues arising from high-throughput techniques recently introduced in biology, and cover very recent developments in computational genomics and population genetics, including genome structural variant discovery, association mapping, epigenome analysis, cancer genomics, and transcriptome analysis. The course material will be drawn from very recent literature. Grading will be based on weekly write-ups for critiques of the papers to be discussed in the class, class participation, and a final project. It assumes a basic knowledge of machine learning and computational genomics.

**02-518 Computational Medicine**

Fall: 12 units

Modern medical research increasingly relies on the analysis of large patient datasets to enhance our understanding of human diseases. This course will focus on the computational problems that arise from studies of human diseases and the translation of research to the bedside to improve human health. The topics to be covered include computational strategies for advancing personalized medicine, pharmacogenomics for predicting individual drug responses, metagenomics for learning the role of the microbiome in human health, mining electronic medical records to identify disease phenotypes, and case studies in complex human diseases such as cancer and asthma. We will discuss how machine learning methodologies such as regression, classification, clustering, semi-supervised learning, probabilistic modeling, and time-series modeling are being used to analyze a variety of datasets collected by clinicians. Class sessions will consist of lectures, discussions of papers from the literature, and guest presentations by clinicians and other domain experts. Grading will be based on homework assignments and a project. 02-250 is a suggested pre-requisite.

Course Website: <https://sites.google.com/view/computationalmedicine/>

**02-530 Cell and Systems Modeling**

Fall: 12 units

This course will introduce students to the theory and practice of modeling biological systems from the molecular to the organism level with an emphasis on intracellular processes. Topics covered include kinetic and equilibrium descriptions of biological processes, systematic approaches to model building and parameter estimation, analysis of biochemical circuits modeled as differential equations, modeling the effects of noise using stochastic methods, modeling spatial effects, and modeling at higher levels of abstraction or scale using logical or agent-based approaches. A range of biological models and applications will be considered including gene regulatory networks, cell signaling, and cell cycle regulation. Weekly lab sessions will provide students hands-on experience with methods and models presented in class. Course requirements include regular class participation, bi-weekly homework assignments, a take-home exam, and a final project. The course is designed for graduate and upper-level undergraduate students with a wide variety of backgrounds. The course is intended to be self-contained but students may need to do some additional work to gain fluency in core concepts. Students should have a basic knowledge of calculus, differential equations, and chemistry as well as some previous exposure to molecular biology and biochemistry. Experience with programming and numerical computation is useful but not mandatory. Laboratory exercises will use MATLAB as the primary modeling and computational tool augmented by additional software as needed. Prerequisites: (03-121 or 33-121 or 03-151) and (03-232 or 03-231) and 21-112 and 09-105

**02-540 Bioimage Informatics**

Intermittent: 12 units

With the rapid advance of bioimaging techniques and fast accumulation of bioimage data, computational bioimage analysis and modeling are playing an increasingly important role in understanding of complex biological systems. The goals of this course are to provide students with the ability to understand a broad set of practical and cutting-edge computational techniques to extract knowledge from bioimages.

Prerequisites: 02-620 or 10-301 or 10-315 or 10-601 or 10-701

**02-601 Programming for Scientists**

Fall and Spring: 12 units

Provides a practical introduction to programming for students with little previous programming experience who are interested in science. Fundamental scientific algorithms will be introduced, and extensive programming assignments will be based on analytical tasks that might be faced by scientists, such as parsing, simulation, and optimization. Principles of good software engineering will also be stressed. The course will introduce students to the Go programming language, an industry-supported, modern programming language, the syntax of which will be covered in depth. Other assignments may be given in other programming languages to highlight the commonalities and differences between languages. No biology background is needed. Analytical skills, an understanding of programming basics, and mathematical maturity are required.

Course Website: <http://compeau.cbd.cmu.edu/programming-for-scientists/>**02-602 Professional Issues for Computational and Automated Scientists**

Fall and Spring: 3 units

This course gives Master's in Computational Biology and Master's in Automated Science students the opportunity to develop the professional skills necessary for a successful career in either academia or industry. This course, required in the first semester of both programs, will include assistance with elevator pitches, interview preparation, resume and cover letter writing, networking, and presentation skills. The course will also include opportunities to connect with computational biology professionals as part of industry outreach. The course will meet once a week and is pass/fail only.

**02-604 Fundamentals of Bioinformatics**

Spring: 12 units

How do we find potentially harmful mutations in your genome? How can we reconstruct the Tree of Life? How do we compare similar genes from different species? These are just three of the many central questions of modern biology that can only be answered using computational approaches. This 12-unit course will delve into some of the fundamental computational ideas used in biology and let students apply existing resources that are used in practice every day by thousands of biologists. The course offers an opportunity for students who possess an introductory programming background to become more experienced coders within a biological setting. As such, it presents a natural next course for students who have completed 02-601.

**02-605 Professional Issues in Automated Science**

Spring: 3 units

This course gives MS in Automated Science students an opportunity to develop professional skills necessary for a successful career in computational biology. This course will include assistance with resume writing, interview preparation, presentation skills, and job search techniques. This course will also include opportunities to network with computational biology professionals and academic researchers.

**02-613 Algorithms and Advanced Data Structures**

Fall and Spring: 12 units

The objective of this course is to study algorithms for general computational problems, with a focus on the principles used to design those algorithms. Efficient data structures will be discussed to support these algorithmic concepts. Topics include: Run time analysis, divide-and-conquer algorithms, dynamic programming algorithms, network flow algorithms, linear and integer programming, large-scale search algorithms and heuristics, efficient data storage and query, and NP-completeness. Although this course may have a few programming assignments, it is primarily not a programming course. Instead, it will focus on the design and analysis of algorithms for general classes of problems. This course is not open to CS graduate students who should consider taking 15-651 instead. 02-250 is a suggested prerequisite for undergraduates.

**02-614 String Algorithms**

Intermittent: 12 units

Provides an in-depth look at modern algorithms used to process string data, particularly those relevant to genomics. The course will cover the design and analysis of efficient algorithms for processing enormous amounts of collections of strings. Topics will include string search; inexact matching; string compression; string data structures such as suffix trees, suffix arrays, and searchable compressed indices; and the Borrows-Wheeler transform. Applications of these techniques in genomics will be presented, including genome assembly, transcript assembly, whole-genome alignment, gene expression quantification, read mapping, and search of large sequence databases.

Prerequisites: 15-151 or 15-127 or 21-128

**02-620 Machine Learning for Scientists**

Spring: 12 units

With advances in scientific instruments and high-throughput technology, scientific discoveries are increasingly made from analyzing large-scale data generated from experiments or collected from observational studies. Machine learning methods that have been widely used to extract complex patterns from large speech, text, and image data are now being routinely applied to answer scientific questions in biology, bioengineering, and medicine. This course is intended for graduate students interested in learning machine learning methods for scientific data analysis and modeling. It will cover classification and regression techniques such as logistic regression, random forest regression, Gaussian process regression, decision trees, and support vector machines; unsupervised learning methods such as clustering algorithms, mixture models, and hidden Markov models; probabilistic graphical models and deep learning methods; and learning theories such as PAC learning and VC dimension. The course will focus on applications of these methods in genomics and medicine. Programming skills and basic knowledge of linear algebra, probability, statistics are assumed.

Prerequisite: 02-680

**02-651 New Technologies and Future Markets**

Fall: 12 units

This course focuses on technological trends and how these trends can help shape or disrupt new and existing markets. Students will learn to identify, analyze, and synthesize emerging trends and perform detailed research on how these trends can influence and create markets. By understand the drivers behind these trends students will be able to identify key market opportunity inflection points in biotechnology as well as the relationship between business processes and information technology (IT). Students will also learn to assess some information technologies and the potential of applying them to solve problems and create commercially viable solutions. The course is designed for the student interested in finding new venture opportunities on the cutting edge of technology and finding and evaluating the opportunities for further development. For MS Biotechnology Innovation and Computation students only.

Prerequisite: 11-695

**02-654 Biotechnology Enterprise Development**

Fall: 12 units

In this course students learn how to develop a biotech start-up, create a Minimum Viable Product (MVP), business model and strategy for the product. Students will learn about business modeling, customer development, customer validation, proposal, product branding, and marketing for their product. The course will require students to spend most time to validate their start up concept and prototypes with potential customers and adapt to critical feedback and revise their respective value propositions accordingly. Students learn to balance technical product development with customer requirements, business strategy and budget constraints. This course provides real world, hands-on learning on what it is like to start a company. Different business modeling will be covered. By understand customer discovery and validation concepts will aid students to effectively modify their original concepts to meet market demands. Student teams will learn how to revise, improve their prototype by the end of the term. This is a fast paced course in which students are expected to spend most of the time outside of the classroom to interact with potential customers to validate, test, verify, and integrate essential elements for their start-up business proposal. Up to now, students have been learning some technologies and methods for solving problems in the life science industry and build a prototype for their start-up. However, a new venture proposal is not a collection of isolated bits. It should be thorough validated via customer's inputs and market needs to tell a single story of how the venture will reach its end goals. Final deliverable is creation and presentation of a well explicated, business proposal in addition to a product prototype corresponding to the business proposal.

Prerequisites: 11-695 and 02-651

**02-680 Essential Mathematics and Statistics for Scientists**

Fall: 9 units

This course rigorously introduces fundamental topics in mathematics and statistics to first-year master's students as preparation for more advanced computational coursework. Topics are sampled from information theory, graph theory, proof techniques, phylogenetics, combinatorics, set theory, linear algebra, neural networks, probability distributions and densities, multivariate probability distributions, maximum likelihood estimation, statistical inference, hypothesis testing, Bayesian inference, and stochastic processes. Students completing this course will obtain a broad skillset of mathematical techniques and statistical inference as well as a deep understanding of mathematical proof. They will have the quantitative foundation to immediately step into an introductory master's level machine learning or automation course. This background will also serve students well in advanced courses that apply concepts in machine learning to scientific datasets, such as 02-710 (Computational Genomics) or 02-750 (Automation of Biological Research). The course grade will be computed as the result of homework assignments, midterm tests, and class participation.

**02-699 Independent Study in Computational Biology**

Fall and Spring

The student will, under the individual guidance of a faculty member, read and digest process papers or a textbook in an advanced area of computational biology not offered by an existing course at Carnegie Mellon. The student will demonstrate their mastery of the material by a combination of one or more of the following: oral discussions with the faculty member; exercises set by the faculty member accompanying the readings; and a written summary synthesizing the material that the student learned. Permission required.

**02-700 M.S. Thesis Research**

Fall and Spring

This course is for M.S. students who wish to do supervised research for academic credit with a Computational Biology faculty member. Interested students should first contact the Professor with whom they would like to work. If there is mutual interest, the Professor will direct you to the Academic Programs Coordinator, who will enroll you in the course.

Course Website: <https://forms.gle/tDKDs1EujvXAApYK7> (<https://forms.gle/tDKDs1EujvXAApYK7/>)

**02-702 Computational Biology Seminar**

Fall and Spring: 3 units

This course consists of weekly invited presentations on current computational biology research topics by leading scientists. Attendance is mandatory for a passing grade. You must sign in and attend at least 80 percent of the seminars. See course website for seminar locations. Some will be at the University of Pittsburgh and some will be at Carnegie Mellon.

Course Website: <https://www.compbio.cmu.edu/seminar-series/index.html> (<https://www.compbio.cmu.edu/seminar-series/>)

**02-703 Special Topics in Bioinformatics and Computational Biology**

Intermittent: 6 units

A decade ago, mass spectrometry (MS) was merely a qualitative research technique allowing the analysis of samples regarding the presence of specific biomolecules. However, as MS has turned quantitative, more sophisticated experiments can be performed, such as the recording of signal transduction kinetics and the analysis of the composition of protein complexes and organelles. This makes MS-based proteomics a powerful method to study spatiotemporal protein dynamics. The development of relative quantification approaches, which generally use  $^2\text{H}$ ,  $^{13}\text{C}$  or  $^{15}\text{N}$  isotope labels, has especially led to an increase in quantification accuracy and set off numerous new experimental approaches to study protein regulation. In this mini-course, we will cover mass spectrometry principles, discuss classical as well as current primary literature addressing method development and quantitative analysis, and highlight state-of-the-art biological studies that employ MS. A combination of lectures, student presentations, and written exercises will establish a thorough knowledge of current bio-analytical MS approaches.

**02-710 Computational Genomics**

Spring: 12 units

Dramatic advances in experimental technology and computational analysis are fundamentally transforming the basic nature and goal of biological research. The emergence of new frontiers in biology, such as evolutionary genomics and systems biology is demanding new methodologies that can confront quantitative issues of substantial computational and mathematical sophistication. From the computational side this course focuses on modern machine learning methodologies for computational problems in molecular biology and genetics, including probabilistic modeling, inference and learning algorithms, data integration, time series analysis, active learning, etc. This course counts as a CSD Applications elective

**02-711 Computational Molecular Biology and Genomics**

Spring: 12 units

An advanced introduction to computational molecular biology, using an applied algorithms approach. The first part of the course will cover established algorithmic methods, including pairwise sequence alignment and dynamic programming, multiple sequence alignment, fast database search heuristics, hidden Markov models for molecular motifs and phylogeny reconstruction. The second part of the course will explore emerging computational problems driven by the newest genomic research. Course work includes four to six problem sets, one midterm and final exam. Prerequisites: (03-151 or 03-121) and 15-122

**02-712 Computational Methods for Biological Modeling and Simulation**

Fall: 12 units

This course covers a variety of computational methods important for modeling and simulation of biological systems. It is intended for graduates and advanced undergraduates with either biological or computational backgrounds who are interested in developing computer models and simulations of biological systems. The course will emphasize practical algorithms and algorithm design methods drawn from various disciplines of computer science and applied mathematics that are useful in biological applications. The general topics covered will be models for optimization problems, simulation and sampling, and parameter tuning. Course work will include problem sets with significant programming components and independent or group final projects.

Prerequisites: (15-110 or 15-112) and (02-201 or 02-613)

**02-714 String Algorithms**

Fall: 12 units

Provides an in-depth look at modern algorithms used to process string data, particularly those relevant to genomics. The course will cover the design and analysis of efficient algorithms for processing enormous collections of strings. Topics will include string search; inexact matching; string compression; string data structures such as suffix trees, suffix arrays, and searchable compressed indices; and the Borrows-Wheeler transform. Applications of these techniques in biology will be presented, including genome assembly, transcript assembly, whole-genome alignment, gene expression quantification, read mapping, and search of large sequence databases. No knowledge of biology is assumed, and the topics covered will be of use in other fields involving large collections of strings. Programming proficiency is required.

Prerequisite: 15-251

**02-715 Advanced Topics in Computational Genomics**

Spring: 12 units

Research in biology and medicine is undergoing a revolution due to the availability of high-throughput technology for probing various aspects of a cell at a genome-wide scale. The next-generation sequencing technology is allowing researchers to inexpensively generate a large volume of genome sequence data. In combination with various other high-throughput techniques for epigenome, transcriptome, and proteome, we have unprecedented opportunities to answer fundamental questions in cell biology and understand the disease processes with the goal of finding treatments in medicine. The challenge in this new genomic era is to develop computational methods for integrating different data types and extracting complex patterns accurately and efficiently from a large volume of data. This course will discuss computational issues arising from high-throughput techniques recently introduced in biology, and cover very recent developments in computational genomics and population genetics, including genome structural variant discovery, association mapping, epigenome analysis, cancer genomics, and transcriptome analysis. The course material will be drawn from very recent literature. Grading will be based on weekly write-ups for critiques of the papers to be discussed in the class, class participation, and a final project. It assumes a basic knowledge of machine learning and computational genomics.

**02-717 Algorithms in Nature**

Fall: 12 units

Computer systems and biological processes often rely on networks of interacting entities to reach joint decisions, coordinate and respond to inputs. There are many similarities in the goals and strategies of biological and computational systems which suggest that each can learn from the other. These include the distributed nature of the networks (in biology molecules, cells, or organisms often operate without central control), the ability to successfully handle failures and attacks on a subset of the nodes, modularity and the ability to reuse certain components or sub-networks in multiple applications and the use of stochasticity in biology and randomized algorithms in computer science. In this course we will start by discussing classic biologically motivated algorithms including neural networks (inspired by the brain), genetic algorithms (sequence evolution), non-negative matrix factorization (signal processing in the brain), and search optimization (ant colony formation). We will then continue to discuss more recent bi-directional studies that have relied on biological processes to solve routing and synchronization problems, discover Maximal Independent Sets (MIS), and design robust and fault tolerant networks. In the second part of the class students will read and present new research in this area. Students will also work in groups on a final project in which they develop and test a new biologically inspired algorithm. See also: [www.algorithmsinnature.org](http://www.algorithmsinnature.org) no prior biological knowledge required.

**02-718 Computational Medicine**

Fall: 12 units

Modern medical research increasingly relies on the analysis of large patient datasets to enhance our understanding of human diseases. This course will focus on the computational problems that arise from studies of human diseases and the translation of research to the bedside to improve human health. The topics to be covered include computational strategies for advancing personalized medicine, pharmacogenomics for predicting individual drug responses, metagenomics for learning the role of the microbiome in human health, mining electronic medical records to identify disease phenotypes, and case studies in complex human diseases such as cancer and asthma. We will discuss how machine learning methodologies such as regression, classification, clustering, semi-supervised learning, probabilistic modeling, and time-series modeling are being used to analyze a variety of datasets collected by clinicians. Class sessions will consist of lectures, discussions of papers from the literature, and guest presentations by clinicians and other domain experts. Grading will be based on homework assignments and a project. 02-250 is a suggested pre-requisite.

Prerequisites: 10-315 or (10-701 and 10-601 and 10-401)

Course Website: <https://sites.google.com/view/computationalmedicine/>**02-719 Genomics and Epigenetics of the Brain**

Fall: 12 units

This course will provide an introduction to genomics, epigenetics, and their application to problems in neuroscience. The rapid advances in single cell sequencing and other genomic technologies are revolutionizing how neuroscience research is conducted, providing tools to study how different cell types in the brain produce behavior and contribute to neurological disorders. Analyzing these powerful new datasets requires a foundation in molecular neuroscience as well as key computational biology techniques. In this course, we will cover the biology of epigenetics, how proteins sitting on DNA orchestrate the regulation of genes. In parallel, programming assignments and a project focusing on the analysis of a primary genomic dataset will teach principles of computational biology and their applications to neuroscience. The course material will also serve to demonstrate important concepts in neuroscience, including the diversity of neural cell types, neural plasticity, the role that epigenetics plays in behavior, and how the brain is influenced by neurological and psychiatric disorders. Although the course focuses on neuroscience, the material is accessible and applicable to a wide range of topics in biology.

Prerequisites: (03-151 or 03-121) and 03-220 and (02-201 or 15-110 or 15-121)

**02-721 Algorithms for Computational Structural Biology**

Intermittent: 12 units

Some of the most interesting and difficult challenges in computational biology and bioinformatics arise from the determination, manipulation, or exploitation of molecular structures. This course will survey these challenges and present a variety of computational methods for addressing them. Topics will include: molecular dynamics simulations, computer-aided drug design, and computer-aided protein design. The course is appropriate for both students with backgrounds in computer science and those in the life sciences.

**02-725 Computational Methods for Proteogenomics and Metabolomics**

Spring: 12 units

Proteomics and metabolomics are the large scale study of proteins and metabolites, respectively. In contrast to genomes, proteomes and metabolomes vary with time and the specific stress or conditions an organism is under. Applications of proteomics and metabolomics include determination of protein and metabolite functions (including in immunology and neurobiology) and discovery of biomarkers for disease. These applications require advanced computational methods to analyze experimental measurements, create models from them, and integrate with information from diverse sources. This course specifically covers computational mass spectrometry, structural proteomics, proteogenomics, metabolomics, genome mining and metagenomics.

Prerequisites: 02-250 or 02-251 or 02-604

**02-730 Cell and Systems Modeling**

Fall: 12 units

This course will introduce students to the theory and practice of modeling biological systems from the molecular to the organism level with an emphasis on intracellular processes. Topics covered include kinetic and equilibrium descriptions of biological processes, systematic approaches to model building and parameter estimation, analysis of biochemical circuits modeled as differential equations, modeling the effects of noise using stochastic methods, modeling spatial effects, and modeling at higher levels of abstraction or scale using logical or agent-based approaches. A range of biological models and applications will be considered including gene regulatory networks, cell signaling, and cell cycle regulation. Weekly lab sessions will provide students hands-on experience with methods and models presented in class. Course requirements include regular class participation, bi-weekly homework assignments, a take-home exam, and a final project. The course is designed for graduate and upper-level undergraduate students with a wide variety of backgrounds. The course is intended to be self-contained but students may need to do some additional work to gain fluency in core concepts. Students should have a basic knowledge of calculus, differential equations, and chemistry as well as some previous exposure to molecular biology and biochemistry. Experience with programming and numerical computation is useful but not mandatory. Laboratory exercises will use MATLAB as the primary modeling and computational tool augmented by additional software as needed. \*THIS COURSE WILL BE AT PITT

Prerequisites: (33-121 or 03-121 or 03-151) and (03-232 or 03-231) and 21-112 and 09-105

Course Website: <https://sites.google.com/site/cellandsystemsmodeling/>

**02-731 Modeling Evolution**

Spring: 12 units

Some of the most serious public health problems we face today, from drug-resistant bacteria, to cancer, all arise from a fundamental property of living systems and #8212; their ability to evolve. Since Darwin's theory of natural selection was first proposed, we have begun to understand how heritable differences in reproductive success drive the adaptation of living systems. This makes it intuitive and tempting to view evolution from an optimization perspective. However, genetic drift, phenotypic trade-offs, constraints, and changing environments, are among the many factors that may limit the optimizing force of natural selection. This tug-of-war between selection and drift, between the forces that produce variation in a population, and the forces suppressing this variation, make evolutionary processes much more complex to model and understand than previously thought. The aim of this class is to provide an introduction into the theoretical formalism necessary to understand how biological systems are shaped by the forces and constraints driving evolutionary dynamics.

Prerequisites: 15-112 and 21-241 and (21-325 or 36-218 or 15-259 or 36-225)

**02-740 Bioimage Informatics**

Intermittent: 12 units

With the rapid advance of bioimaging techniques and fast accumulation of bioimage data, computational bioimage analysis and modeling are playing an increasingly important role in understanding of complex biological systems. The goals of this course are to provide students with the ability to understand a broad set of practical and cutting-edge computational techniques to extract knowledge from bioimages.

**02-750 Automation of Scientific Research**

Spring: 12 units

Automated scientific instruments are used widely in research and engineering. Robots dramatically increase the reproducibility of scientific experiments, and are often cheaper and faster than humans, but are most often used to execute brute-force sweeps over experimental conditions. The result is that many experiments are "wasted" on conditions where the effect could have been predicted. Thus, there is a need for computational techniques capable of selecting the most informative experiments. This course will introduce students to techniques from Artificial Intelligence and Machine Learning for automatically selecting experiments to accelerate the pace of discovery and to reduce the overall cost of research. Real-world applications from Biology, Bioengineering, and Medicine will be studied. Grading will be based on homeworks and two exams. The course is intended to be self-contained, but students should have a basic knowledge of biology, programming, statistics, and machine learning.

Prerequisites: 10-601 or 10-701 or 02-620

**02-760 Laboratory Methods for Computational Biologists**

Fall and Spring: 9 units

Computational biologists frequently focus on analyzing and modeling large amounts of biological data, often from high-throughput assays or diverse sources. It is therefore critical that students training in computational biology be familiar with the paradigms and methods of experimentation and measurement that lead to the production of these data. This one-semester laboratory course gives students a deeper appreciation of the principles and challenges of biological experimentation. Students learn a range of topics, including experimental design, structural biology, next generation sequencing, genomics, proteomics, bioimaging, and high-content screening. Class sessions are primarily devoted to designing and performing experiments in the lab using the above techniques. Students are required to keep a detailed laboratory notebook of their experiments and summarize their resulting data in written abstracts and oral presentations given in class-hosted lab meetings. With an emphasis on the basics of experimentation and broad views of multiple cutting-edge and high-throughput techniques, this course is appropriate for students who have never taken a traditional undergraduate biology lab course, as well as those who have and are looking for introductory training in more advanced approaches. Grading: Letter grade based on class participation, laboratory notebooks, experimental design assignments, and written and oral presentations. 02-250 is a suggested pre-requisite.

**02-761 Laboratory Methods for Automated Biology I**

Fall: 12 units

In order to rapidly generate reproducible experimental data, many modern biology labs leverage some form of laboratory automation to execute experiments. In the not so distant future, the use of laboratory automation will continue to increase in the biological lab to the point where many labs will be fully automated. Therefore, it is critical for automation scientists to be familiar with the principles, experimental paradigms, and techniques for automating biological experimentation with an eye toward the fully automated laboratory. In this laboratory course, students will learn about various automatable experimental methods, design of experiments, hardware for preparing samples and executing automated experiments, and software for controlling that hardware. These topics will be taught in lectures as well as through laboratory experience using multi-purpose laboratory robotics. During weekly laboratory time, students will complete and integrate parts of two larger projects. The first project will be focused on liquid handling, plate control, plate reading, and remote control of the automated system based on experimental data. The second project will be focused on the design, implementation, and analysis of a high content screening campaign using fluorescence microscopy, image analysis, and tissue culture methods.

**02-762 Laboratory Methods for Automated Biology II**

Spring: 12 units

This laboratory course provides a continuation and extension of experiences in 02-761. Instruction will consist of lectures and laboratory experience using multi-purpose laboratory robotics. During weekly laboratory time, students will complete and integrate parts of two larger projects. The first project will be focused on the execution of a molecular biology experiment requiring nucleic acid extraction, library preparation for sequencing, and quality control. The second project will be focused on the implementation and execution of automated methods using active learning techniques to direct the learning of a predictive model for a large experimental space (such as learning the effects of many possible drugs on many possible targets). Grading will be based on lab and project completion and quality. Prerequisite: 02-761

**02-764 Automated Science Capstone II**

Spring: 12 units

This course consists of small group projects on development, implementation and/or execution of automated science campaigns in collaboration with industry and/or academic partners. This course may only be taken as part of a continuous sequence with 02-763. Enrollment is only open to M.S. in Automated Science students.

## Human-Computer Interaction Courses

**05-090 Human-Computer Interaction Practicum**

All Semesters: 3 units

This course is for HCII students who wish to have an internship experience as part of their curriculum. Students are required to write a one-page summary statement prior to registration that explains how their internship connects with their HCI curriculum, specifically on how it uses material they have learned as well as prepares them for future courses. Near the end of the internship, students will be required to submit a reflection paper that describes the work they did in more detail, including lessons learned about the work experience and how they utilized their HCI education to work effectively. International students should consult with the Office of International Education for appropriate paperwork and additional requirements before registration. Units earned count toward the total required units necessary for degree completion; students should speak with an academic advisor for details. This course may be taken at most 3 times for a total of 9 units maximum. Students normally register for this course for use during the summer semester.

**05-120 Introduction to Human-Computer Interaction**

Spring: 5 units

This course is only for freshman SCS students that are interested in learning more about HCI.

**05-200 Ethics and Policy Issues in Computing**

Intermittent: 9 units

Should autonomous robots make life and death decisions on their own? Should we allow them to select a target and launch weapons? To diagnose injuries and perform surgery when human doctors are not around? Who should be permitted to observe you, find out who your friends are, what you do and say with them, what you buy, and where you go? Do social media and personalized search restrict our intellectual horizons? Do we live in polarizing information bubbles, just hearing echoes of what we already know and believe? As computing technology becomes ever more pervasive and sophisticated, we are presented with an escalating barrage of decisions about who, how, when, and for what purposes technology should be used. This course will provide an intellectual framework for discussing these pressing issues of our time, as we shape the technologies that in turn shape us. We will seek insight through reading, discussion, guest lectures, and debates. Students will also undertake an analysis of a relevant issue of their choice, developing their own position, and acquiring the research skills needed to lend depth to their thinking. The course will enhance students' ability to think clearly about contentious technology choices, formulate smart positions, and support their views with winning arguments.

**05-291 Learning Media Design**

Fall: 12 units

[IDeATe collaborative course] Learning is a complex human phenomenon with cognitive, social and personal dimensions that need to be accounted for in the design of technology enhanced learning experiences. In this studio course students will apply learning science concepts to critique existing forms of learning media, establish a set of design precedents to guide project work and produce a series of design concepts that support learning interactions in a real-world context. Collaborating in small interdisciplinary teams, students will partner with a local informal learning organization (e.g. museum, after school program provider, maker space) to conduct learning design research studies, synthesize findings, establish learning goals and iteratively prototype and assess design concepts. As final deliverables, students will present their design research findings, design concepts, and prototypes to stakeholders, and draft a media-rich proposal for their learning media concept to pitch to a local funder. Please note that there may be usage/materials fees associated with this course. Please note that there may be usage/materials fees associated with this course.

**05-292 IDeATe: Learning in Museums**

Spring: 12 units

Learning in Museums brings together students from across the disciplines to consider the design of mediated learning experiences through a project-based inquiry course. Students will be introduced to a range of design research methods and associated frameworks that explore the cognitive, social and affective dimensions of learning in everyday contexts through readings, invited lectures, in-class activities and assignments. Students will conduct a series of short design research studies to define learning goals and develop supporting design concepts that improve learning outcomes for diverse participants in informal learning settings (e.g. museums, after school programs, maker spaces or online). In concept development, we will look at how to position technology and question its role in the setting to engage and foster positive learning interactions. This course will culminate in a media-rich presentation of design concepts and a prototype to a stakeholder audience, and include an evaluation plan describing how learning outcomes for the project would be assessed.

**05-300 HCI Undergraduate Pro Seminar**

All Semesters: 2 units

HCI is a broad field that brings together approaches from design, computer science, and psychology. This course provides an introduction to the field of HCI and to the HCI community at CMU. Guest speakers from around campus will provide a general introduction to these approaches and how they are pursued at CMU, and will describe research opportunities that are available to undergraduates. The course will also discuss career options in both industry and academia for students of HCI, and will include presentations from HCI alumni and sessions on preparing resumes, creating portfolios, and interviewing for jobs. The course is designed for current or potential HCI majors and minors but is open to anyone with an interest in applying for the HCI major/minor.

Course Website: <https://hcii.cmu.edu/academics/courses> (<https://hcii.cmu.edu/academics/courses/>)

**05-315 Persuasive Design**

Fall: 12 units

This project-based course focuses on the ethical, human-centered design and evaluation of persuasive technologies that aim to change users' attitudes, emotions, or behaviors in ways that benefit the self and/or society. In addition to exposing students to an array of psychological theories and strategies for implicit and explicit persuasion, the course will cover a variety of topics illustrating both the pitfalls and possibilities in designing for positive impact in HCI. The focal point of the class will be the semester project, for which student teams will iteratively conceptualize, prototype, implement, and evaluate a tool, system, or change to a ubiquitous computing environment that intends to stimulate and sustain belief or behavior change (such as reducing cognitive or social biases, building healthy or prosocial habits, or resisting other persuasive forces one encounters on a daily basis).

**05-317 Design of Artificial Intelligence Products**

Intermittent: 12 units

This course teaches students how to design new products and services that leverage the capabilities of AI and machine learning to improve the quality of peoples lives. Students will learn to follow a matchmaking design, user-centered design, and service design process. Students will learn to ideate; reframing problematic situations by envisioning many possible products and services. Students will learn to iteratively refine and assess their ideas with real users/customers. Class projects will focus on the challenges of deploying systems that generate errors and the challenges of situating intelligent systems such that they harmonize the best qualities of human and machine intelligence.

Course Website: <https://hcii.cmu.edu/academics/courses> (<https://hcii.cmu.edu/academics/courses/>)

**05-318 Human AI Interaction**

Intermittent: 12 units

Artificial Intelligence is inspired by human intelligence, made powerful by human data, and ultimately only useful in how it positively affects the human experience. This course is an introduction to harnessing the power of AI so that it is beneficial and useful to people. We will cover a number of general topics: agency and initiative, AI and ethics, bias and transparency, confidence and errors, human augmentation and amplification, trust and explainability, mixed-initiative systems, and programming by example. These topics will be explored via projects in dialog and speech-controlled systems, automatic speech recognition, computer vision, data science, recommender systems, text summarization, learning science, UI personalization, and visualization. Students will complete individual weekly mini-projects in which they will design and build AI systems across a wide variety of domains. Students should be comfortable with programming; assignments will be primarily in Python and Javascript. Prior experience with AI/machine learning will be useful but is not required. Students will also be responsible for weekly readings and occasional presentations to the class.

Course Website: <http://www.hcii.cmu.edu/academics/courses> (<http://www.hcii.cmu.edu/academics/courses/>)

**05-319 Data Visualization**

Fall: 12 units

This course is an introduction to key design principles and techniques for interactively visualizing data. The major goals of this course are to understand how visual representations can help in the analysis and understanding of complex data, how to design effective visualizations, and how to create your own interactive visualizations using modern web-based frameworks.

Course Website: <https://dig.cmu.edu/courses/2022-fall-datavis.html>

**05-320 Social Web**

Intermittent: 12 units

With the growth of online environments like MySpace, Second Life, World of Warcraft, Wikipedia, blogs, online support groups, and open source development communities, the web is no longer just about information. This course, jointly taught by a computer scientist and a behavioral scientist, will examine a sampling of the social, technical and business challenges social web sites must solve to be successful, teach students how to use high-level tools to analyze, design or build online communities, and help them understand the social impact of spending at least part of their lives online. This class is open to advanced undergraduates and graduate students with either technical or non-technical backgrounds. Course work will include lectures and class discussion, homework, class presentations, and a group research or design project.

**05-321 Transformational Game Design Studio**

Fall: 12 units

TBA



**05-333 Gadgets, Sensors and Activity Recognition in HCI**

Fall: 12 units

Recent advances in HCI have been driven by new capabilities to deliver inexpensive devices to users, to display information in mobile and other contexts, to sense the user and their environment, and use these sensors to create models of a user's context and actions. This course will consider both concepts surrounding these new technological opportunities through discussion of current literature - and practical considerations the skills needed to actually build devices. About 1/3 of this class will review current advances in this area. The remainder will be devoted to development of individual skills so that students leaving the class will have an ability to actually build small devices for human interaction (in short: "HCI gadgets"). In particular, the course will concentrate on the basics of building simple microcontroller-based devices and will also provide very basic coverage of the machine learning techniques needed for simple sensor-driven statistical models. The course is designed to be accessible to students with a wide range of backgrounds including both technically-oriented and non-technical students (especially Designers) interested in HCI. The class will be project oriented with 4-5 electronic prototype building projects during the semester. At least two of these projects will be self-defined in nature and can be adapted to the existing skills and interests of each student. There are no formal prerequisites for this class. However, the class will involve programming and debugging of micro-controllers. Some coverage of the language used to do this will be provided, and if required by your background, the programming component of the projects can be made comparatively small (but, in that case some other aspect of the projects will need to be expanded). However, you should not take this course if you have no programming background. This course assumes no background in electronics.

Course Website: <http://www.hcii.cmu.edu/courses/applied-gadgets-sensors-and-activity-recognition-hci> (<http://www.hcii.cmu.edu/courses/applied-gadgets-sensors-and-activity-recognition-hci/>)

**05-341 Organizational Communication**

All Semesters: 9 units

Most of management is communication. You communicate to get information that will be the basis of decisions, coordinate activity, to provide a vision for the people who work for and with you, to and to sell yourself and your work. The goal of this course is to identify communication challenges within work groups and organizations and ways to overcome them. To do this requires that we know how communication normally works, what parts are difficult, and how to fix it when it goes wrong. The focus of this course is on providing you with a broad understanding of the way communication operates within dyads, work groups, and organizations. The intent is to give you theoretical and empirical underpinnings for the communication you will undoubtedly participate in when you move to a work environment, and strategies for improving communication within your groups. Because technology is changing communication patterns and outcomes both in organizations and more broadly in society, the course examines these technological changes. Readings come primarily from the empirical research literature.

Course Website: <http://www.hcii.cmu.edu/courses/organizational-communication> (<http://www.hcii.cmu.edu/courses/organizational-communication/>)

**05-360 Interaction Design Fundamentals**

Fall and Spring: 12 units

IXD Fundamentals introduces the human-centered design process as well as fundamental interaction design principles, methods, and practices. The course is for both students who may only enroll in one interaction design course and those who intend to build upon their HCI learning by taking advanced interaction design courses. Students must work effectively as individuals and in small teams to learn interaction design concepts and apply them to real-world problems. By the end of this course students should be able to; -Apply appropriate interaction design methods in a human-centered design process. -Create persuasive interim and final design artifacts that demonstrate communication design fundamentals. -Facilitate productive and structured critique across the class and with instructors. - Explain and apply fundamental interaction design principles. -Create clarity and readability in artifacts, including GUIs and deliverables, through the disciplined application of visual design principles such as typography, color and composition. -Practice reframing a given problem in order to create opportunities that drive generating multiple solutions. -Demonstrate habits that foster the creative process, including drawing, divergent thinking, and creative experimentation. -Identify and explore with interaction design materials. This course serves as a prerequisite for Advanced Interaction Design Studio (number TBD). Students who are required to take this course have priority and will be enrolled first. No coding is required.

**05-361 Advanced Interaction Design**

Spring: 12 units

Advanced Interaction Design (05-361/05-661) follows Interaction Design for Human-Computer Interaction (05-360/05-660). Students are expected to build on the basic interaction design principles they learned in Interaction Design for Human-Computer Interaction by applying advanced methods to solve more complex problems using emerging technologies in user experiences that cross devices, modalities and contexts. Students learn how to design with advanced technologies that predict, assist and automate, and make through a design system. Systems thinking, data as a design material, and UI design are emphasized in projects which are designed to give students experience solving complex problems that they are likely to encounter as practitioners. Advanced Interaction Design prepares students to become interaction designers that take a rigorous and principled approach to solving enterprise-scale problems where many systems and applications serve many stakeholders.

**05-380 Prototyping Algorithmic Experiences**

Intermittent: 15 units

This project-based course provides an overview and hands-on introduction to iterative prototyping methods in HCI, with an emphasis on current and emerging technologies such as data-driven algorithmic systems, AI and machine learning, spatial computing, and IoT. Students will learn and implement approaches for creating and using prototypes to iteratively inform the creation of new technologies. The course will help students learn to strategically evaluate whether a given prototyping approach is a good fit for a given design or research question. In addition to HCI undergraduate majors, the course is open to undergraduate and graduate level students with proficiency in programming and prior courses or experience in user-centered research, design, and/or evaluation. Some exceptions to the course prerequisites will be granted with permission of the instructor. Prerequisites: 15-112 and (15-122 or 15-121 or 15-150 or 15-210) and (05-410 or 05-391) and (05-452 or 05-650 or 05-651 or 05-470 or 05-317)

**05-391 Designing Human Centered Software**

All Semesters: 12 units

"Why are things so hard to use these days? Why doesn't this thing I just bought work? Why is this web site so hard to use? These are frustrations that we have all faced from systems not designed with people in mind. The question this course will focus on is: how can we design human-centered systems that people find useful and usable? This course is a broad introduction to designing, prototyping, and evaluating user interfaces. If you take only one course in Human-Computer Interaction, this is the course for you. We will cover theory as well as practical application of ideas from Human-Computer Interaction. Coursework includes lectures, class discussion, homework, class presentations, and group projects. This class is open to all undergrads and grad students, with either technical or non-technical majors. However, there is a programming prerequisite. " Prerequisites: 15-110 or 15-104 or 15-122 or 15-112

Course Website: <http://www.hcii.cmu.edu/courses/designing-human-centered-software> (<http://www.hcii.cmu.edu/courses/designing-human-centered-software/>)

**05-392 Interaction Design Overview**

Fall: 9 units

This studio course offers a broad overview of communication and interaction design. Students will learn design methodologies such as brainstorming, sketching, storyboarding, wire framing, and prototyping. Students learn to take a human-centered design approach to their work. Assignments include short in-class exercises as well as individual and team-based projects. Students take part in studio critiques, engaging in critical discussions about the strengths and weaknesses of their own work and the work of others. No coding is required.

**05-395 Applications of Cognitive Science**

Spring: 9 units

The goal of this course is to examine cases where basic research on cognitive science, including cognitive neuroscience, has made its way into application, in order to understand how science gets applied more generally. The course focuses on applications that are sufficiently advanced as to have made an impact outside of the research field per se; for example, as a product, a change in practice, or a legal statute. Examples are virtual reality (in vision, hearing, and touch), cognitive tutors, phonologically based reading programs, latent semantic analysis applications to writing assessment, and measures of consumers' implicit attitudes. The course will use a case-study approach that considers a set of applications in detail, while building a general understanding of what it means to move research into the applied setting. The questions to be considered include: What makes a body of theoretically based research applicable? What is the pathway from laboratory to practice? What are the barriers - economic, legal, entrenched belief or practice? The format will emphasize analysis and discussion by students. They should bring to the course an interest in application; extensive prior experience in cognitive science is not necessary. The course will include tutorials on basic topics in cognitive science such as perception, memory, and spatial cognition. These should provide sufficient grounding to discuss the applications.

Course Website: <http://www.hcii.cmu.edu/courses/applications-cognitive-science> (<http://www.hcii.cmu.edu/courses/applications-cognitive-science/>)

**05-410 User-Centered Research and Evaluation**

Fall: 12 units

This course provides an overview and introduction to the field of human-computer interaction (HCI). It introduces students to tools, techniques, and sources of information about HCI and provides a systematic approach to design. The course increases awareness of good and bad design through observation of existing technology and teaches the basic skills of generative and evaluative research methods. This is a companion course to courses in visual design (51-422) and software implementation (05-430, 05-431). When registering for this course, undergraduate students are automatically placed the wait list. Students will be then moved into the class, based on if they are in the BHCI second major and year in school e.g. seniors, juniors, etc In the Fall, this course is NOT open to students outside the HCI major. The Spring offering is open to all students. This course is a core requirement for students in the HCI additional major.

**05-413 Human Factors**

Fall: 9 units

This course uses theory and research from human factors, cognitive science, and social science to understand and design the interactions of humans with the built world, tools, and technology. The course emphasizes current work in applied domains such as automotive design, house construction, medical human factors, and design of information devices. The course also will emphasize not only individual human factors (e.g., visual response, anthropometry) but also the organizational arrangements that can amplify or correct human factors problems. Through reading, discussion, and projects, you will learn about human perceptual, cognitive, and physical processes that affect how people interact with, and use, technology and tools. You will learn why we have so many automobile accidents, voting irregularities, and injuries from prescription medication. You will learn some tried and true solutions for human factors problems, and some of the many problems in human factors that remain. You will also have gained experience in research in this field.

Course Website: <http://www.hcii.cs.cmu.edu>

**05-417 Computer-mediated Communication**

Spring: 6 units

This course examines fundamental aspects of interpersonal communication and considers how different types of computer-mediated communications (CMC) technologies affect communication processes. Among the topics we will consider are: conversational structure and CMC, tools to support nonverbal and paralinguistic aspects of communication such as gesture and eye gaze, and social and cultural dimensions of CMC. Students will be expected to post to weekly discussion lists, to write a paper on a specific aspect of CMC, and to present a talk on their final project to the class. The course should be appropriate for graduate students in all areas and for advanced undergraduates.

**05-418 Design Educational Games**

Spring: 12 units

The potential of digital games to improve education is enormous. However, it is a significant challenge to create a game that is both fun and educational. In this course, students will learn to meet this challenge by combining processes and principles from game design and instructional design. Students will also learn to evaluate their games for fun, learning, and the integration of the two. They will be guided by the EDGE framework for the analysis and design educational games. The course will involve a significant hands-on portion, in which students learn a design process to create educational games digital or non-digital. They will also read about existing educational games and discuss game design, instructional design, learning and transfer, and the educational effectiveness of digital games. They will analyze an educational game and present their analysis to the class.

Course Website: <https://www.hcii.cmu.edu/course/design-of-educational-games> (<https://www.hcii.cmu.edu/course/design-of-educational-games/>)

**05-430 Programming Usable Interfaces**

Spring: 15 units

This course is combines lecture, and an intensive programming lab and design studio. It is for those who want to express their interactive ideas in working prototypes. It will cover the importance of human-computer interaction/interface design, iterative design, input/output techniques, how to design and evaluate interfaces, and research topics that will impact user interfaces in the future. In lab, you will learn how to design and program effective graphical user interfaces, and how to perform user tests. We will cover a number of prototyping tools and require prototypes to be constructed in each, ranging from animated mock-ups to fully functional programs. Assignments will require implementing UIs, testing that interface with users, and then modifying the interface based on findings. Some class sessions will feature design reviews of student work. This course is for HCII Masters students and HCI dual majors with a minimal programming background. Students will often not be professional programmers, but will need to interact with programmers. RECITATION SELECTION: Students taking this course can sign up for either Prototyping Lab recitation. PREREQUISITES: Proficiency in a programming language, program structure, algorithm analysis, and data abstraction. Normally met through an introductory programming course using C, C++, Pascal or Java, such as 15100, 15112, 15127 or equivalent. Students entering this course should be able to independently write a 300-line program in 48 hours. This course is NOT open to students outside of the BHCI program. Prerequisites: 15-127 or 15-112 or 15-110 or 15-104 or 15-100

**05-431 Software Structures for User Interfaces**

Fall: 12 units

This course considers the basic and detailed concepts for building software to implement user interfaces (UIs). It considers factors of input, output, application interface, and related infrastructure as well as the typical patterns used to implement them. It considers how these aspects are organized and managed within a well-structured object oriented system. We will cover a variety of "front-end" programming contexts, including conventional graphical user interface (GUI) programming for mobile apps (phones, watches), web apps, and regular desktop applications, across a variety of frameworks. We will also cover programming for data-driven and conversational (AI) user interfaces. We will briefly touch on front-end programming for visualizations, games, 3D, and virtual and artificial reality (VR and AR), along with interactive UI tools such as prototypers and resource editors. The homeworks and project in this course will involve extensive object-oriented programming, likely in both Java and JavaScript, so this course is only appropriate for students with a strong programming background. Note that this is not an HCI methods course and #8212; we do not cover user-centered design or evaluation methods. This course is designed for students in the SCS HCI undergrad Major, but it also available to any undergrad or graduate student with an interest in the topic and solid prior programming experience who wish to understand the structures needed for professional development of interactive systems. Note that all students who register for this class will initially be placed on a waitlist. Priority for getting into the class are students in the HCII programs (more senior students first), and then others. The graduate (05-631) and undergraduate (05-431) numbers are for the same course with the same work.

Prerequisites: 17-514 or 17-437 or 17-214 or 14-513 or 15-513 or 18-213 or 15-213 or 15-214

**05-432 Personalized Online Learning**

Fall: 12 units

Online learning has become widespread (e.g., MOOCs, online and blended courses, and Khan Academy) and many claim it will revolutionize higher education and K-12. How can we make sure online learning is maximally effective? Learners differ along many dimensions and they change over time. Therefore, advanced learning technologies must adapt to learners to provide individualized learning experiences. This course covers a number of proven personalization techniques used in advanced learning technologies. One of the techniques is the use of cognitive modeling to personalize practice of complex cognitive skills in intelligent tutoring systems. This approach, developed at CMU, may well be the most significant application of cognitive science in education and is commercially successful. We will also survey newer techniques, such as personalizing based on student meta-cognition, affect, and motivation. Finally, we will look at personalization approaches that are widely believed to be effective but have not proven to be so. The course involves readings and discussion of different ways of personalizing instruction, with an emphasis on cognitive modeling approaches. Students will learn to use the Cognitive Tutor Authoring Tools (CTAT) to implement tutor prototypes that rely on computer-executable models of human problem solving to personalize instruction. The course is meant for graduate or advanced undergraduate students in Human-Computer Interaction, Psychology, Computer Science, Design, or related fields, who are interested in educational applications. Students should either have some programming skills or experience in the cognitive psychology of human problem solving, or experience with instructional design.

Course Website: <http://www.hcii.cmu.edu/courses/personalized-online-learning> (<http://www.hcii.cmu.edu/courses/personalized-online-learning/>)

**05-433 Programming Usable Interfaces OR Software Structures for Usable Interfaces**

Fall: 6 units

Section A: Programming Usable Interfaces Section B: Software Structures for Usable Interfaces This is a lecture-only course (see 05-430/05-630 or 05-431/631 for the lecture + lab version of these courses) that is intended for those who want to learn how to design and evaluate user interfaces. We will cover the importance of human-computer interaction and interface design, the iterative design cycle used in HCI, an overview of input and output techniques, how to design and evaluate interaction techniques, and end with a discussion of hot topics in research that will impact user interfaces in the coming years. This course is only intended for HCII Masters students or HCI undergraduate majors who have already taken an associated User Interface lab, or non-MHCI/BHCI students interested in the design of user interfaces. WAITLIST LOGISTICS: Note that ALL students who register for this class will initially be placed on a waitlist. Your position on the waitlist is not an indication of whether you will be accepted into the class. Contacting the instructor will not move you off the waitlist. Priority for getting off the waitlist are MHCI students, BHCI students (more senior students first), and then others.

**05-434 Machine Learning in Practice**

Fall and Spring: 12 units

Machine Learning is concerned with computer programs that enable the behavior of a computer to be learned from examples or experience rather than dictated through rules written by hand. It has practical value in many application areas of computer science such as on-line communities and digital libraries. This class is meant to teach the practical side of machine learning for applications, such as mining newsgroup data or building adaptive user interfaces. The emphasis will be on learning the process of applying machine learning effectively to a variety of problems rather than emphasizing an understanding of the theory behind what makes machine learning work. This course does not assume any prior exposure to machine learning theory or practice. In the first 2/3 of the course, we will cover a wide range of learning algorithms that can be applied to a variety of problems. In particular, we will cover topics such as decision trees, rule based classification, support vector machines, Bayesian networks, and clustering. In the final third of the class, we will go into more depth on one application area, namely the application of machine learning to problems involving text processing, such as information retrieval or text categorization. 05-834 is the HCII graduate section. If you are an LTI student, please sign up for the LTI graduate course number (11-663) ONLY to count properly towards your degree requirements. 05-434 is the HCII undergraduate section. If you are an LTI student, please sign up for the LTI undergraduate course number (11-344) ONLY to count properly towards your degree requirements.

Course Website: <http://www.hcii.cmu.edu/courses/applied-machine-learning> (<http://www.hcii.cmu.edu/courses/applied-machine-learning/>)

**05-435 Applied Fabrication for HCI**

Fall: 12 units

This course will consider how new fabrication techniques such as 3D printing, laser cutting, CNC machining and related computer controlled technologies can be applied to problems in Human-Computer Interaction. Each offering will concentrate on a particular application domain for its projects. This year the course will consider assistive technology. This course will be very hands-on and skills-oriented, with the goal of teaching students the skills necessary to apply these technologies to HCI problems such as rapid prototyping of new device concepts. To this end? Every student in this course will build and take home a 3D printer. (There will be \$400-\$500 cost associated with this course to make that possible. Details on this are still to be determined.)

**05-436 Usable Privacy and Security**

Spring: 9 units

There is growing recognition that technology alone will not provide all of the solutions to security and privacy problems. Human factors play an important role in these areas, and it is important for security and privacy experts to have an understanding of how people will interact with the systems they develop. This course is designed to introduce students to a variety of usability and user interface problems related to privacy and security and to give them experience in designing studies aimed at helping to evaluate usability issues in security and privacy systems. The course is suitable both for students interested in privacy and security who would like to learn more about usability, as well as for students interested in usability who would like to learn more about security and privacy. Much of the course will be taught in a graduate seminar style in which all students will be expected to do a weekly reading assignment and each week different students will prepare a presentation for the class. Students will also work on a group project throughout the semester. The course is open to all graduate students who have technical backgrounds. The 12-unit course numbers (08-734 and 5-836) are for PhD students and masters students. Students enrolled in these course numbers will be expected to play a leadership role in a group project that produces a paper suitable for publication. The 9-unit 500-level course numbers (08-534 and 05-436) are for juniors, seniors, and masters students. Students enrolled in these course numbers will have less demanding project and presentation requirements.

Course Website: <http://www.hcii.cmu.edu/courses/usable-privacy-and-security> (<http://www.hcii.cmu.edu/courses/usable-privacy-and-security/>)

**05-439 The Big Data Pipeline: Collecting and Using Big Data for Interactive Systems**

Spring: 12 units

This course covers techniques and technologies for creating data driven interfaces. You will learn about the entire data pipeline from sensing to cleaning data to different forms of analysis and computation.

Course Website: <http://data.cmu.edu>

**05-440 Interaction Techniques**

Intermittent: 12 units

This course will provide a comprehensive study of the many ways to interact with computers and computerized devices. An "interaction technique" starts when the user does something that causes an electronic device to respond and includes the direct feedback from the device to the user. Examples include physical buttons and switches, on-screen menus and scroll bars operated by a mouse, touch screen widgets and gestures such as flick-to-scroll, text entry on computers or touch screens, game controllers, interactions in 3D and virtual/augmented reality, consumer electronic controls such as remote controls, and adaptations of all of these for people with disabilities. We will start with a history of the invention and development of these techniques, discuss the various options used today, and continue on to the future with the latest research on interaction techniques presented at conferences such as ACM CHI and UIST. Appropriate design and evaluation methods for interaction techniques will also be covered. Guest lectures from inventors of interaction techniques are planned. Students will have a choice for final projects that can focus on historical or novel interaction techniques.

Course Website: <http://www.cs.cmu.edu/~bam/uicourse/05440inter/>

**05-452 Service Design**

Fall: 12 units

In this course, we will collectively define and study services and product service systems, and learn the basics of designing them. We will do this through lectures, studio projects, and verbal and written exposition. Classwork will be done individually and in teams.

**05-470 Digital Service Innovation**

Intermittent: 12 units

Attention entrepreneurs, designers, and engineers! This course teaches you to invent digital services. You will learn about value-creation in the service sector and a human-centered design process including brainstorming, storyboarding, interviewing, video sketches, and pitching. Students work in small, interdisciplinary teams to discover unmet needs of users. They conceive of a digital service and assess its technical feasibility, financial viability, and desirability. Then they produce a plan with a business model and a video sketch and pitch it to industry professionals. Grades will be determined primarily by the quality of the team's products.

**05-499 Special Topics in HCI**

Fall and Spring: 12 units

Special Topics in HCI is an opportunity for students interested in HCI to gain a deeper understanding of a specific area in this field. Each class is designed to cover an emerging research area within HCI, from designing large-scale peer learning systems to designing games around audience agency. All sections will help students: (1) build a more comprehensive understanding of an area of study within HCI, (2) work closely with faculty and peers to create mini-projects or team assignments that help students master the course material, (3) explore evidence-based research methods and techniques in HCI. Sections will vary in topic and often change from semester to semester. Because of this, students can take multiple sections, as they are individual classes. Undergraduate sections are listed as 499 and graduate sections are listed as 899. For descriptions of specific sections for this academic year, visit the "Courses" section on the Human-Computer Interaction Institute website.

Course Website: <http://www.hcii.cmu.edu/academics/courses> (<http://www.hcii.cmu.edu/academics/courses/>)

**05-540 Rapid Prototyping of Computer Systems**

Spring: 12 units

This is a project-oriented course, which will deal with all four aspects of project development: the application, the artifact, the computer-aided design environment, and the physical prototyping facilities. The class consists of students from different disciplines who must synthesize and implement a system in a short period of time. Upon completion of this course the student will be able to: generate systems specifications from a perceived need; partition functionality between hardware and software; produce interface specifications for a system composed of numerous subsystems; use computer-aided development tools; fabricate, integrate, and debug a hardware/software system; and evaluate the system in the context of an end user application. The class consists of students from different disciplines who must synthesize and implement a system in a short period of time.

Course Website: <http://www.hcii.cmu.edu/courses/rapid-prototyping-computer-systems> (<http://www.hcii.cmu.edu/courses/rapid-prototyping-computer-systems/>)

**05-571 Undergraduate Project in HCI**

Spring: 12 units

Experiential learning is a key component of the MHCI program. Through a substantial team project, students apply classroom knowledge in analysis and evaluation, implementation and design, and develop skills working in multidisciplinary teams. Student teams work with Carnegie Mellon University-based clients or external clients to iteratively design, build and test a software application which people directly use.  
Prerequisites: 05-610 Min. grade B or 05-431 Min. grade B or 05-630 Min. grade B or 05-631 Min. grade B or 05-410 Min. grade B or 05-430 Min. grade B

Course Website: <http://www.hcii.cmu.edu/courses/undergraduate-project-hci> (<http://www.hcii.cmu.edu/courses/undergraduate-project-hci/>)

**05-589 Independent Study in HCI-UG**

All Semesters

In collaboration with and with the permission of the professor, undergraduate students may engage in independent project work on any number of research projects sponsored by faculty. Students must complete an Independent Study Proposal, negotiate the number of units to be earned, complete a contract, and present a tangible deliverable. The Undergraduate Program Advisor's signature is required for HCI undergraduate-level Independent Study courses. Registration is through the HCII Undergraduate Programs Manager only.

**05-600 HCI Pro Seminar**

Fall: 6 units

This course is only for MHCI students. This course is specifically built to expose students to the world of HCI through research and industry talks, as well as strengthening HCI communication skills for work in industry. Seminar Component: To expose students to the world of HCI through research and industry expert talks with written assignments. Conflict Management Component: To educate students on conflict management, teamwork, active listening skills, and communication skills in order to give them tools to collaborate and work more efficiently on multi-disciplinary teams. Professional Series Component: To expose students to the world of HCI through guest speakers, prepare students to navigate job hunting through resume and portfolio workshops and to provide industry insights into HCI and the profession through guest speakers and panel discussions.

Course Website: <http://www.hcii.cs.cmu.edu>

**05-602 IDEATe: Learning in Museums**

Spring: 12 units

Learning in Museums brings together students from across the disciplines to consider the design of mediated learning experiences in a project-based inquiry course. Students will be introduced to a range of design research methods and associated frameworks that explore the cognitive, social and affective dimensions of learning in everyday contexts through readings, invited lectures, in-class activities and assignments. Students will conduct a series of short design research studies to define learning goals and develop supporting design concepts intended to improve learning outcomes for diverse participants in informal learning settings (e.g. museums, after-school programs, maker spaces or online). In concept development, we will look at how to position technology and question its role in the setting to engage and foster positive learning interactions and conversation. This semester we will be working with the Carnegie Museum of Natural History as our primary stakeholder. The course will culminate in a media-rich presentation of design concepts and a fielded prototype to a review panel and include a piloted evaluation plan describing how learning outcomes for the project would be assessed. In consultation with the instructor, students in the graduate section of the course will be assigned an HCI/learning research literature review and presentation related to their project topic.

**05-610 User-Centered Research and Evaluation**

Fall: 12 units

This course provides an overview and introduction to the field of human-computer interaction (HCI). It introduces students to tools, techniques, and sources of information about HCI and provides a systematic approach to design. The course increases awareness of good and bad design through observation of existing technology, and teaches the basic skills of generative and evaluative research methods. This is a companion course to software implementation (05-430, 05-431 05-380). When registering for this course, undergraduate students are automatically placed the wait list. Students will be then moved into the class, based on if they are in the BHCI primary or second major and year in school e.g. seniors, juniors, etc. Freshman are not permitted to register in this course. In the Fall, this course is NOT open to students outside the HCI major or MHCI. The Spring offering is open to all students. This course is a core requirement for students in the HCI additional major and the MHCI program.

Course Website: <http://www.hcii.cs.cmu.edu>

**05-615 Persuasive Design**

Fall: 12 units

This project-based course focuses on the ethical, human-centered design and evaluation of persuasive technologies that aim to change users' attitudes, emotions, or behaviors in ways that benefit the self and/or society. In addition to exposing students to an array of psychological theories and strategies for implicit and explicit persuasion, the course will cover a variety of topics illustrating both the pitfalls and possibilities in designing for positive impact in HCI. The focal point of the class will be the semester project, for which student teams will iteratively conceptualize, prototype, implement, and evaluate a tool, system, or change to a ubiquitous computing environment that intends to stimulate and sustain belief or behavior change (such as reducing cognitive or social biases, building healthy or prosocial habits, or resisting other persuasive forces one encounters on a daily basis).

**05-618 Human AI Interaction**

Intermittent: 12 units

Artificial Intelligence is inspired by human intelligence, made powerful by human data, and ultimately only useful in how it positively affects the human experience. This course is an introduction to harnessing the power of AI so that it is beneficial and useful to people. We will cover a number of general topics: agency and initiative, AI and ethics, bias and transparency, confidence and errors, human augmentation and amplification, trust and explainability, mixed-initiative systems, and programming by example. These topics will be explored via projects in dialog and speech-controlled systems, automatic speech recognition, computer vision, data science, recommender systems, text summarization, learning science, UI personalization, and visualization. Students will complete individual weekly mini-projects in which they will design and build AI systems across a wide variety of domains. Students should be comfortable with programming; assignments will be primarily in Python and Javascript. Prior experience with AI/machine learning will be useful but is not required. Students will also be responsible for weekly readings and occasional presentations to the class.

Course Website: <https://www.hcii.cmu.edu/academics/courses> (<https://www.hcii.cmu.edu/academics/courses/>)

**05-619 Data Visualization**

Fall: 12 units

This course is an introduction to key design principles and techniques for interactively visualizing data. The major goals of this course are to understand how visual representations can help in the analysis and understanding of complex data, how to design effective visualizations, and how to create your own interactive visualizations using modern web-based frameworks.

Course Website: <https://dig.cmu.edu/courses/2022-fall-datavis.html>

**05-650 Interaction Design Studio II**

Spring: 12 units

This course follows Interaction Design Fundamentals (05-651). Students are expected to apply what they have learned about design thinking and methodologies as a starting point for all assignments. Students will work in teams to perform guerrilla research, synthesize data, and consider the needs of multiple stakeholders in their design of mobile services and other intelligent systems. Design concepts go beyond user interfaces to include sensors, controls, and ubiquitous computing. Emphasis is placed on the quality of the students ideas and their ability to give form to their design concepts. By completing and presenting their work, students will gain skills related to professional UX design practice.  
Prerequisite: 05-651

Course Website: <http://www.hcii.cmu.edu/courses/interaction-design-studio> (<http://www.hcii.cmu.edu/courses/interaction-design-studio/>)

**05-651 Interaction Design Studio I**

Fall: 12 units

This studio course introduces students to design thinking and the basic practices of interaction design. We follow a human-centered design process that includes research, concept generation, prototyping, and refinement. Students must work effectively as individuals and in small teams to design mobile information systems and other interactive experiences. Assignments approach design on three levels: specific user interactions, contexts of use, and larger systems. Students will become familiar with design methodologies such as sketching, storyboarding, wire framing, prototyping, etc. No coding is required. This course serves as a prerequisite for Interaction Design Studio (05-650). Students who are required to take this course have priority and will be enrolled first.

**05-660 Interaction Design Fundamentals**

Fall and Spring: 12 units

IXD Fundamentals introduces the human-centered design process as well as fundamental interaction design principles, methods, and practices. The course is for both students who may only enroll in one interaction design course and those who intend to build upon their HCI learning by taking advanced interaction design courses. Students must work effectively as individuals and in small teams to learn interaction design concepts and apply them to real-world problems. By the end of this course students should be able to; -Apply appropriate interaction design methods in a human-centered design process. -Create persuasive interim and final design artifacts that demonstrate communication design fundamentals. -Facilitate productive and structured critique across the class and with instructors. - Explain and apply fundamental interaction design principles. -Create clarity and readability in artifacts, including GUIs and deliverables, through the disciplined application of visual design principles such as typography, color and composition. -Practice reframing a given problem in order to create opportunities that drive generating multiple solutions. -Demonstrate habits that foster the creative process, including drawing, divergent thinking, and creative experimentation. -Identify and explore with interaction design materials. This course serves as a prerequisite for Advanced Interaction Design Studio (number TBD). Students who are required to take this course have priority and will be enrolled first. No coding is required.

**05-661 Advanced Interaction Design**

Spring: 12 units

Advanced Interaction Design (05-361/05-661) follows Interaction Design for Human-Computer Interaction (05-360/05-660). Students are expected to build on the basic interaction design principles they learned in Interaction Design for Human-Computer Interaction by applying advanced methods to solve more complex problems using emerging technologies in user experiences that cross devices, modalities and contexts. Students learn how to design with advanced technologies that predict, assist and automate, and make through a design system. Systems thinking, data as a design material, and UI design are emphasized in projects which are designed to give students experience solving complex problems that they are likely to encounter as practitioners. Advanced Interaction Design prepares students to become interaction designers that take a rigorous and principled approach to solving enterprise-scale problems where many systems and applications serve many stakeholders.

**05-670 Digital Service Innovation**

Fall: 12 units

Attention entrepreneurs, designers, and engineers! This course teaches you to invent digital services. You will learn about value-creation in the service sector and a human-centered design process including brainstorming, storyboarding, interviewing, video sketches, and pitching. Students work in small, interdisciplinary teams to discover unmet needs of users. They conceive of a digital service and assess its technical feasibility, financial viability, and desirability. Then they produce a plan with a business model and a video sketch and pitch it to industry professionals. Grades will be determined primarily by the quality of the team's products.

**05-674 Ethics and Policy Issues in Computing**

Intermittent: 9 units

Should autonomous robots make life and death decisions on their own? Should we allow them to select a target and launch weapons? To diagnose injuries and perform surgery when human doctors are not around? Who should be permitted to observe you, find out who your friends are, what you do and say with them, what you buy, and where you go? Do social media and personalized search restrict our intellectual horizons? Do we live in polarizing information bubbles, just hearing echoes of what we already know and believe? As computing technology becomes ever more pervasive and sophisticated, we are presented with an escalating barrage of decisions about who, how, when, and for what purposes technology should be used. This course will provide an intellectual framework for discussing these pressing issues of our time, as we shape the technologies that in turn shape us. We will seek insight through reading, discussion, guest lectures, and debates. Students will also undertake an analysis of a relevant issue of their choice, developing their own position, and acquiring the research skills needed to lend depth to their thinking. The course will enhance students' ability to think clearly about contentious technology choices, formulate smart positions, and support their views with winning arguments.

**05-680 Independent Study in HCI - METALS**

All Semesters

With the permission of the professor, METALS students may engage in independent project work on any number of innovative research projects sponsored by faculty. Students must complete an Independent Study Proposal, negotiate the number of units to be earned, submit a contract, and present a tangible deliverable. The Program Advisor's signature is required for the METALS Independent Study course.

**05-685 Prototyping Algorithmic Experiences**

Intermittent: 15 units

This project-based course provides an overview and hands-on introduction to iterative prototyping methods in HCI, with an emphasis on current and emerging technologies such as data-driven algorithmic systems, AI and machine learning, spatial computing, and IoT. Students will learn and implement approaches for creating and using prototypes to iteratively inform the creation of new technologies. The course will help students learn to strategically evaluate whether a given prototyping approach is a good fit for a given design or research question. In addition to HCI undergraduate majors, the course is open to undergraduate and graduate level students with proficiency in programming and prior courses or experience in user-centered research, design, and/or evaluation. Some exceptions to the course prerequisites will be granted with the permission of the instructor. Prerequisites: 15-112 and (15-121 or 15-150 or 15-210 or 15-122) and (05-891 or 05-610) and (05-651 or 05-650 or 05-652 or 05-617 or 05-670)

**05-738 Evidence-Based Educational Design**

Fall: 12 units

The aim of this course is to teach students how to develop educational goals based on a detailed task analysis of the knowledge, skills, and dispositions required for mastery of a particular aspect of a domain. Goals for early childhood, elementary, middle school, high school, postsecondary, and adult education will be discussed and related to relevant state, national, and international standards. A comprehensive understanding of student achievement will be developed. The importance of matching the instructional program and its assessment to goals will be discussed and demonstrated. Assessment that focuses on covering the full range of specified goals will be studied along with diverse approaches for valid assessment. Other topics include making instructional material choices, funding, classroom management, ethics, schools, and social systems. Assignments will emphasize linking goals - instruction - assessment. A term project will consist of an in-depth study of one central unit in a discipline or grade level. Undergraduates are invited to take this course with special permission from Dr. Lauren Hercik.

**05-823 E-Learning Design Principles and Methods**

Fall: 12 units

This course is about e-learning design principles, the evidence and theory behind them, and how to apply these principles to develop effective educational technologies. It is organized around the book "e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning" by Clark and Mayer with further readings drawn from cognitive science, educational psychology, and human-computer interaction. You will learn design principles 1) for combining words, audio, and graphics in multimedia instruction, 2) for combining examples, explanations, practice and feedback in online support for learning by doing, and 3) for balancing learner versus system control and supporting student metacognition. You will read about the experiments that support these design principles, see examples of how to design such experiments, and practice applying the principles in educational technology development.

Course Website: [http://www.learnlab.org/research/wiki/index.php/E-learning\\_Design\\_Principles\\_2013#Course\\_Details](http://www.learnlab.org/research/wiki/index.php/E-learning_Design_Principles_2013#Course_Details) ([http://www.learnlab.org/research/wiki/index.php/E-learning\\_Design\\_Principles\\_2013/#Course\\_Details](http://www.learnlab.org/research/wiki/index.php/E-learning_Design_Principles_2013/#Course_Details))

**05-839 Interactive Data Science**

Spring: 12 units

This course covers techniques and technologies for creating data driven interfaces. You will learn about the entire data pipeline from sensing to cleaning data to different forms of analysis and computation.

Course Website: <https://hcii.cmu.edu/academics/courses> (<https://hcii.cmu.edu/academics/courses/>)

**05-840 Tools for Online Learning**

Fall: 12 units

In this course, we will explore issues that pertain to interaction and interface design. The class will focus on elements of the larger interaction design process including basic design principles, information architecture and navigation, planning and brainstorming methods, and techniques for developing rapid sketches and prototypes. Course Requirements: This class will not focus on learning specific software tools. Students are expected to have prior experience using a variety of design and programming tools. Please speak with the instructor if you have questions regarding these prerequisites. This course was design for students in the METALS program.

**Software Societal Systems Courses****08-200 Ethics and Policy Issues in Computing**

Spring: 9 units

In this course, students will study the social impacts of computing technology and systems. The course will provide a brief introduction to ethics and to the new and difficult ethical questions modern computing technology presents us with. It will focus on a number of areas in which computers and information technology are having an impact on society including data privacy, social media, and autonomous technologies.

**08-722 Data Structures for Application Programmers**

Fall and Spring: 6 units

This course is an introduction to Data Structures and a few fundamental algorithms for students with some prior programming experience (functions, loops and arrays mainly in Java). It covers the conceptual and implementation views of some common data structures and algorithms. It also goes over the Java Collections (such as List, ArrayList, LinkedList, Set, HashSet, TreeSet, Map, HashMap, TreeMap, PriorityQueue) to solidify the understanding of the data structures. There is an introduction to the analysis of algorithms that operate on them. Following learning-by-doing methodology, there will be many repetitions of writing code and reviews of the items covered in lectures. Students are required to be familiar with Java Programming before taking this course. Those who are not are encouraged to take 08-671 in mini 1 before taking this course. Students are required to have a reasonably modern laptop computer on which install the Java software used for this course.

**Language Technologies Institute Courses****11-291 Applied Computational Intelligence Lab**

Intermittent: 9 units

What would an "intelligent" picture on the wall do? What if it could see and hear you? What should it say if it could talk? What if your pantry, wardrobe or medicine cabinet could sense, think and act? What should they do and say? What should your cell phone be saying to you? These are not whimsical or theoretical questions...they inevitably arise as ordinary everyday objects around us acquire the ability to sense changes in their environment, think about their implications, and act in pursuit of their goals. These objects are connected to the web and become conduits for services, erasing the distinction between products and services. The ability to invent and build smart products/services is becoming a key skill in the new technology-driven services economy. The focus of the course will be on building "ordinary" objects that can sense, think and act in the real world and on exploring the implications of these capabilities. Students will select their own project and by the end of the semester will create a working prototype that will be exhibited in a public place. Prizes will be offered for the most creative projects. In the course of their projects, students will learn how to use state-of-the-art tools for: Object detection using video cameras, microphones and other sensors Movement and gesture detection Speech recognition and generation Reasoning and planning: While the course organizers have many ideas for specific projects, students will be encouraged to design their own projects. Students are expected to work in small groups on their own time and receive faculty advice as needed. There will be weekly meetings of the whole class.

Prerequisites: 15-122 Min. grade C and 21-127 Min. grade C

**11-324 Human Language for Artificial Intelligence**

Fall: 12 units

An enduring aspect of the quest to build intelligent machines is the challenge of human language. This course introduces students with a background in computer science and a research interest in artificial intelligence fields to the structure of natural language, from sound to society. It covers phonetics (the physical aspects of speech), phonology (the sound-structure of language), morphology (the structure of words), morphosyntax (the use of word and phrase structure to encode meaning), syntactic formalisms (using finite sets of production rules to characterize infinite configurations of structure), discourse analysis and pragmatics (language in discourse and communicative context), and sociolinguistics (language in social context and social meaning). Evaluation is based on seven homework assignments, a midterm examination, and a final examination.

**11-344 Machine Learning in Practice**

Fall and Spring: 12 units

Machine Learning is concerned with computer programs that enable the behavior of a computer to be learned from examples or experience rather than dictated through rules written by hand. It has practical value in many application areas of computer science such as on-line communities and digital libraries. This class is meant to teach the practical side of machine learning for applications, such as mining newsgroup data or building adaptive user interfaces. The emphasis will be on learning the process of applying machine learning effectively to a variety of problems rather than emphasizing an understanding of the theory behind what makes machine learning work. This course does not assume any prior exposure to machine learning theory or practice. In the first 2/3 of the course, we will cover a wide range of learning algorithms that can be applied to a variety of problems. In particular, we will cover topics such as decision trees, rule based classification, support vector machines, Bayesian networks, and clustering. In the final third of the class, we will go into more depth on one application area, namely the application of machine learning to problems involving text processing, such as information retrieval or text categorization.

**11-345 Undergrad Independent Study**

All Semesters

No course description provided.

**11-364 An Introduction to Knowledge-Based Deep Learning and Socratic Coaches**

Spring: 12 units

The subject of this course will be deep learning, one of the most dynamic and exciting emerging areas of computer science. Deep learning deals with and is conquering the problems resulting from the enormous quantity of data that now surrounds us. Furthermore, the course will explore knowledge-based deep learning, a new methodology invented by the instructor that offers many potential advantages over conventional deep learning. This is a learn-by-doing, team-project based course, which will be divided into four phases. In phase one, each student will read and present a number of papers describing state-of-the-art deep learning systems and successful applications. In phase two, each team will implement the system described in one of the papers. In phase three, each team will scale that implementation to one of the large benchmark datasets. In phase four, each team will do a special research project implementing a knowledge-based deep learning system based on pending patent applications of Professor Baker. As a potential follow-on for successful projects, students may participate in a summer course on entrepreneurial applications of deep learning or work as interns in a bootstrap startup based on the knowledge-based deep learning projects. Prerequisite: Strong quantitative aptitude, programming skill, ability to quickly absorb new ideas, teamwork skills.

**11-390 LTI Minor Project - Juniors**

All Semesters: 12 units

No course description provided.

**11-411 Natural Language Processing**

Intermittent: 12 units

This course is about a variety of ways to represent human languages (like English and Chinese) as computational systems, and how to exploit those representations to write programs that do neat stuff with text and speech data, like translation, summarization, extracting information, question answering, natural interfaces to databases, and conversational agents. This field is called Natural Language Processing or Computational Linguistics, and it is extremely multidisciplinary. This course will therefore include some ideas central to Machine Learning and to Linguistics. We'll cover computational treatments of words, sounds, sentences, meanings, and conversations. We'll see how probabilities and real-world text data can help. We'll see how different levels interact in state-of-the-art approaches to applications like translation and information extraction. From a software engineering perspective, there will be an emphasis on rapid prototyping, a useful skill in many other areas of Computer Science.

Prerequisite: 15-122

**11-422 Grammar Formalisms**

Spring: 12 units

TBA

**11-423 ConLanging: Lrng. Ling. & Lang Tech via Constr. Artif. Lang.**

Spring: 12 units

Students will work individually or in small groups to create artificial human(oid) languages for fictional human cultures or SciFi worlds. Students will implement language technologies for their languages. In the course of creating the languages, students will learn about the building blocks of human language such as phones, phonemes, morphemes, and morpho-syntactic constructions including their semantics and pragmatics. Class instruction will focus specifically on variation among human languages so that the students can make conlangs that are not just naively English-like. We will also touch on philosophical issues in philosophy of language and on real-world socio-political issues related to language policy. Students will be required to use at least one of the following technologies: language documentation tools that are used for field linguistics and corpus annotation, automatic speech recognition, speech synthesis, morphological analysis, parsing, or machine translation. Learning Objectives: 1. The building blocks (phonemes, morphemes, etc.) of language, how languages are built from them, and how they interact 2. Metalinguistic awareness and knowledge about variation in human language 3. Language, thought, and culture: how does language reflect thought and culture, and vice versa. Why wouldn't Elvish be a good language for Klingons? 4. Language policy in the real world: For students who want to manipulate real languages. 5. Historical linguistics and language change: for students who want to manipulate real languages or make families of related conlangs for fictional worlds. 6. Practical experience with a language technology. <http://tts.speech.cs.cmu.edu/11-823/> Course Website: <http://tts.speech.cs.cmu.edu/11-823> (<http://tts.speech.cs.cmu.edu/11-823/>)

**11-439 Designing Around Patents on Machine Learning and NLP Technology**

Spring: 9 units

This course uses Machine Learning and Natural Language Processing as vehicles to teach principles in designing software to avoid patents. After introducing students to the basics of patents, we investigate how to use students software skills to design around patents that is, create new technology that avoids a patent while maintaining some, or even all, of the performance and value of the patented technology. Designing around a patent can be viewed as a puzzle that requires technical skill, understanding the business value of technology, knowledge of patents, and creativity. Students will also be able to help design patents that cannot be easily designed-around. Not only does this add a valuable new dimension to the student's skill set, the course material is organized as a vehicle for refining knowledge of ML and NLP techniques. We will practice by designing around patents on well-known ML and NLP algorithms. Students study the basic algorithm, learn the patents that cover those algorithms, and then experiment with software modifications that avoid the patent while preserving as much of the algorithm's performance as possible. In essence, students will view the presence of particular patents as a design constraints, akin to designing for limited memory, bandwidth, or processor speed. Students must have already taken courses in Natural Language Processing (e.g., 11-411) and Machine Learning (e.g., 10-315). Law of Computer Technology 17-562, 17-662, 17-762 or Patents, Licensing, and Innovation 19-473, 19-673 are helpful but not required.

**11-441 Machine Learning for Text and Graph-based Mining**

Fall and Spring: 9 units

This course provides a comprehensive introduction to the theory and implementation of algorithms for organizing and searching large text collections. The first half of the course studies text search engines for enterprise and Web environments; the open-source Indri search engine is used as a working example. The second half studies text mining techniques such as clustering, categorization, and information extraction. Programming assignments give hands-on experience with document ranking algorithms, categorizing documents into browsing hierarchies, and related topics.

**11-442 Search Engines**

Fall: 9 units

This course studies the theory, design, and implementation of text-based search engines. The core components include statistical characteristics of text, representation of information needs and documents, several important retrieval models, and experimental evaluation. The course also covers common elements of commercial search engines, for example, integration of diverse search engines into a single search service ("federated search", "vertical search"), personalized search results, diverse search results, and sponsored search. The software architecture components include design and implementation of large-scale, distributed search engines.

Course Website: <http://boston.lti.cs.cmu.edu/classes/11-642/>

**11-485 Introduction to Deep Learning**

Intermittent: 9 units

Neural networks have increasingly taken over various AI tasks, and currently produce the state of the art in many AI tasks ranging from computer vision and planning for self-driving cars to playing computer games. Basic knowledge of NNs, known currently in the popular literature as "deep learning", familiarity with various formalisms, and knowledge of tools, is now an essential requirement for any researcher or developer in most AI and NLP fields. This course is a broad introduction to the field of neural networks and their "deep" learning formalisms. The course traces some of the development of neural network theory and design through time, leading quickly to a discussion of various network formalisms, including simple feedforward, convolutional, recurrent, and probabilistic formalisms, the rationale behind their development, and challenges behind learning such networks and various proposed solutions. We subsequently cover various extensions and models that enable their application to various tasks such as computer vision, speech recognition, machine translation and playing games. Instruction Unlike prior editions of 11-785, the instruction will primarily be through instructor lectures, and the occasional guest lecture. Evaluation Students will be evaluated based on weekly continuous-evaluation tests, and their performance in assignments and a final course project. There will be six hands-on assignments, requiring both low-level coding and toolkit-based implementation of neural networks, covering basic MLP, convolutional and recurrent formalisms, as well as one or more advanced tasks, in addition to the final project.

Prerequisites: 15-112 and 21-120 and 21-241

**11-488 Concepts in Digital Multimedia and Cyber Forensics**

Spring: 12 units

This course covers the use of computational methods in crime investigation (forensics) and prevention (intelligence). In almost all areas of forensics and intelligence, computational methods continue to aid, and sometimes entirely replace, human expertise in tracking crime. This is desirable since automation can address the problems associated with scale and global crime linkage through diverse data computational tools can potentially overcome and surpass human capabilities for crime investigation. This course is of a cross-disciplinary nature. It amalgamates knowledge from criminology, forensic sciences, computer science, statistics, signal processing, machine learning, AI, psychology, medicine and many other fields. Students from all departments and schools are welcome to take this course.

Course Website: <https://forensics-ai.github.io/gh-syllabus/>**11-490 LTI Minor Project - Seniors**

All Semesters: 12 units

No course description provided.

**11-492 Speech Processing**

Spring: 12 units

Speech Processing offers a practical and theoretical understanding of how human speech can be processed by computers. It covers speech recognition, speech synthesis and spoken dialog systems. The course involves practicals where the student will build working speech recognition systems, speech synthesis systems and integrate them to create a speech interface. This work will be based on existing toolkits. Details of algorithms, techniques and limitations of state of the art speech systems will also be presented. This course is designed for students wishing understand how to process real data for real applications, applying statistical and machine learning techniques as well as working with limitations in the technology.

Prerequisite: 15-210

**11-546 Applied Legal Analytics & Artificial Intelligence**

Spring: 12 units

Technological advances are affecting the legal profession and enable innovation by experts proficient in both law and AI technology. This joint course, co-taught by instructors from the University of Pittsburgh School of Law and Carnegie Mellon University's Language Technologies Institute, provides a hands-on practical introduction to the fields of artificial intelligence and law, machine learning, and natural language processing as they are being applied to support the work of legal professionals, researchers, and administrators, such as extracting semantic information from legal documents and using it to solve legal problems. Meanwhile, LegalTech companies and startups have been tapping into the industry's need to make large-scale document analysis tasks more efficient, and to use predictive analytics for better decision making. This course is intended to bring students of law and technical disciplines together into a collaborative classroom setting to learn about the technologies at the intersection of law and AI through lectures and programming exercises, as well as gain practical experience through collaborative project work. Topics in focus include machine learning and natural language applied to legal data, computational models of legal reasoning, and selected legal issues that relate to AI technologies. Students should come from either a (pre-) law background with a strong interest in gaining practical experience with legal analytics, or from a technical discipline with an equally strong interest in tackling the challenges posed by legal analytics tasks and data.

Course Website: <https://luimagroup.github.io/appliedlegalanalytics/>**11-590 LTI Minor Project - Advanced**

All Semesters: 12 units

No course description provided.

**11-603 Python for Data Science**

Summer: 12 units

Students learn the concepts, techniques, skills, and tools needed for developing programs in Python. Core topics include types, variables, functions, iteration, conditionals, data structures, classes, objects, modules, and I/O operations. Students get an introductory experience with several development environments, including Jupyter Notebook, as well as selected software development practices, such as test-driven development, debugging, and style. Course projects include real-life applications on enterprise data and document manipulation, web scraping, and data analysis.

Course Website: <https://canvas.andrew.cmu.edu>**11-624 Human Language for Artificial Intelligence**

Fall: 12 units

An enduring aspect of the quest to build intelligent machines is the challenge of human language. This course introduces students with a background in computer science and a research interest in artificial intelligence fields to the structure of natural language, from sound to society. It covers phonetics (the physical aspects of speech), phonology (the sound-structure of language), morphology (the structure of words), morphosyntax (the use of word and phrase structure to encode meaning), syntactic formalisms (using finite sets of production rules to characterize infinite configurations of structure), discourse analysis and pragmatics (language in discourse and communicative context), and sociolinguistics (language in social context and social meaning). Evaluation is based on seven homework assignments, a midterm examination, and a final examination.

**11-630 MCDS Practicum Internship**

Summer

The MCDS Practicum course is used for recording CDS students summer internships for the MCDS Program.



**11-639 Designing Around Patents on Machine Learning and NLP Technology**

Spring: 12 units

This course uses Machine Learning and Natural Language Processing as vehicles to teach principles in designing software to avoid patents. After introducing students to the basics of patents, we investigate how to use students software skills to design around patents that is, create new technology that avoids a patent while maintaining some, or even all, of the performance and value of the patented technology. Designing around a patent can be viewed as a puzzle that requires technical skill, understanding the business value of technology, knowledge of patents, and creativity. Students will also be able to help design patents that cannot be easily designed-around. Not only does this add a valuable new dimension to the students skill set, the course material is organized as a vehicle for refining knowledge of ML and NLP techniques. We will practice by designing around patents on well-known ML and NLP algorithms. Students study the basic algorithm, learn the patents that cover those algorithms, and then experiment with software modifications that avoid the patent while preserving as much of the algorithms performance as possible. In essence, students will view the presence of particular patents as a design constraints, akin to designing for limited memory, bandwidth, or processor speed. Students must have already taken courses in Natural Language Processing (e.g., 11-411) and Machine Learning (e.g., 10-315). Law of Computer Technology 17-562, 17-662, 17-762 or Patents, Licensing, and Innovation 19-473, 19-673 are helpful but not required.

**11-646 Applied Legal Analytics & Artificial Intelligence**

Spring: 12 units

Technological advances are affecting the legal profession and enable innovation by experts proficient in both law and AI technology. This joint course, co-taught by instructors from the University of Pittsburgh School of Law and Carnegie Mellon Universitys Language Technologies Institute, provides a hands-on practical introduction to the fields of artificial intelligence and law, machine learning, and natural language processing as they are being applied to support the work of legal professionals, researchers, and administrators, such as extracting semantic information from legal documents and using it to solve legal problems. Meanwhile, LegalTech companies and startups have been tapping into the industrys need to make large-scale document analysis tasks more efficient, and to use predictive analytics for better decision making. This course is intended to bring students of law and technical disciplines together into a collaborative classroom setting to learn about the technologies at the intersection of law and AI through lectures and programming exercises, as well as gain practical experience through collaborative project work. Topics in focus include machine learning and natural language applied to legal data, computational models of legal reasoning, and selected legal issues that relate to AI technologies. Students should come from either a (pre-) law background with a strong interest in gaining practical experience with legal analytics, or from a technical discipline with an equally strong interest in tackling the challenges posed by legal analytics tasks and data.

Course Website: <https://luimagroup.github.io/appliedlegalanalytics/>**11-661 Language and Statistics**

Fall: 12 units

Language technologies (search, text mining, information retrieval, speech recognition, machine translation, question answering, biological sequence analysis...) are at the forefront of this century's information revolution. In addition to their use of machine learning, these technologies rely centrally on classic statistical estimation techniques. Yet most CS and engineering undergraduate programs do not prepare students in this area beyond an introductory prob and stats course. This course is designed to plug this hole. The goal of "Language and Statistics" is to ground the data-driven techniques used in language technologies in sound statistical methodology. We start by formulating various language technology problems in both an information theoretic framework (the source-channel paradigm) and a Bayesian framework (the Bayes classifier). We then discuss the statistical properties of words, sentences, documents and whole languages, and the computational formalisms used to represent language. These discussions naturally lead to specific concepts in statistical estimation. Topics include: Zipf's distribution and type-token curves; point estimators, Maximum Likelihood estimation, bias and variance, sparseness, smoothing and clustering; interpolation, shrinkage, and backoff; entropy, cross entropy and mutual information; decision tree models applied to language; latent variable models and the EM algorithm; hidden Markov models; exponential models and maximum entropy; semantic modeling and dimensionality reduction; probabilistic context-free grammars and syntactic language models. The course is designed for LTI and SCS graduate students, but others are welcome. CS UG upperclassmen who've taken it have done well, though they found it challenging. The 11-661 version does not require the course project. Prerequisites: Strong quantitative aptitude. Comfort with basic UG-level probability. Some programming skill.

Course Website: <http://www.cs.cmu.edu/~roni/11661/>**11-667 Large Language Models Methods and Application**

Fall: 12 units

This course provides a broad foundation for understanding, working with, and adapting existing tools and technologies in the area of Large Language Models like BERT, T5, GPT, and others. It begins with a short history of the area of language models and quickly transitions to a broad survey of the area, offering exposure to the gamut of topics including systems, data, data filtering, training objectives, RLHF/instruction tuning, ethics, policy, evaluation, and other human facing issues. Students will delve into Transformer architectures more broadly and how they work, as well as exploring the reasons why they are better than LSTM-based seq2seq, decoding strategies, etc. Students will learn through readings and hands-on assignments where they will explore techniques for pretraining, attention, prompting, etc. They will then apply these skills in a semester-long course project, making use of locally sourced model instances that offer the opportunity to explore behind the curtain of commercial APIs. Prerequisites: 11-711 or 11-785 or 11-685 or 10-601 or 10-701

**11-696 MIIIS Capstone Planning Seminar**

Spring: 6 units

The MIIIS Capstone Planning Seminar prepares students to complete the MIIIS Capstone Project in the following semester. Students are organized into teams that will work together to complete the capstone project. They define project goals, requirements, success metrics, and deliverables; and they identify and acquires data, software, and other resources required for successful completion of the project. The planning seminar must be completed in the semester prior to taking the capstone project.

**11-697 Introduction to Question Answering**

Fall: 12 units

The Introduction to Question Answering course provides a chance for hands-on, in-depth exploration of core algorithmic approaches to question answering (QA) for students who haven't worked on a QA system before.

**11-711 Advanced Natural Language Processing**

Fall: 12 units

Advanced natural language processing is an introductory graduate-level course on natural language processing aimed at students who are interested in doing cutting-edge research in the field. In it, we describe fundamental tasks in natural language processing such as syntactic, semantic, and discourse analysis, as well as methods to solve these tasks. The course focuses on modern methods using neural networks, and covers the basic modeling and learning algorithms required therefore. The class culminates in a project in which students attempt to reimplement and improve upon a research paper in a topic of their choosing.

**11-716 Graduate Seminar on Dialog Processing**

All Semesters: 6 units

Dialog systems and processes are becoming an increasingly vital area of interest both in research and in practical applications. The purpose of this course will be to examine, in a structured way, the literature in this area as well as learn about ongoing work. The course will cover traditional approaches to the problem, as exemplified by the work of Grosz and Sidner, as well as more recent work in dialog, discourse and evaluation, including statistical approaches to problems in the field. We will select several papers on a particular topic to read each week. While everyone will do all readings, a presenter will be assigned to overview the paper and lead the discussion. On occasion, a researcher may be invited to present their own work in detail and discuss it with the group. A student or researcher taking part in the seminar will come away with a solid knowledge of classic work on dialog, as well as familiarity with ongoing trends.

**11-721 Grammars and Lexicons**

All Semesters: 12 units

Grammars and Lexicons is an introductory graduate course on linguistic data analysis and theory, focusing on methodologies that are suitable for computational implementations. The course covers major syntactic and morphological phenomena in a variety of languages. The emphasis will be on examining both the diversity of linguistic structures and the constraints on variation across languages. Students will be expected to develop and defend analyses of data, capturing linguistic generalizations and making correct predictions within and across languages. The goal is for students to become familiar with the range of phenomena that occur in human languages so that they can generalize the insights into the design of computational systems. The theoretical framework for syntactic and lexical analysis will be Lexical Functional Grammar. Grades will be based on problem sets and take-home exams.

**11-722 Grammar Formalisms**

Intermittent: 12 units

The goal of this course is to familiarize students with grammar formalisms that are commonly used for research in computational linguistics, language technologies, and linguistics. We hope to have students from a variety of disciplines (linguistics, computer science, psychology, modern languages, philosophy) in order to cover a broad perspective in class discussions. Comparison of formalisms will lead to a deeper understanding of human language and natural language processing algorithms. The formalisms will include: Head Driven Phrase Structure Grammar, Lexical Functional Grammar, Tree Adjoining Grammar and Categorical Grammar. If time permits, we will cover Penn Treebank, dependency grammar, and Construction Grammar. We will cover the treatment of basic syntactic and semantic phenomena in each formalism, and will also discuss algorithms for parsing and generating sentences for each formalism. If time permits, we may discuss formal language theory and generative capacity. The course is taught jointly by the following faculty of the Language Technologies Institute: Alan Black Alon Lavie Lori Levin (main coordinator)

**11-724 Human Language for Artificial Intelligence**

Fall: 12 units

An enduring aspect of the quest to build intelligent machines is the challenge of human language. This course introduces students with a background in computer science and a research interest in artificial intelligence fields to the structure of natural language, from sound to society. It covers phonetics (the physical aspects of speech), phonology (the sound-structure of language), morphology (the structure of words), morphosyntax (the use of word and phrase structure to encode meaning), syntactic formalisms (using finite sets of production rules to characterize infinite configurations of structure), discourse analysis and pragmatics (language in discourse and communicative context), and sociolinguistics (language in social context and social meaning). Evaluation is based on seven homework assignments, a midterm examination, and a final examination.

Course Website: <http://www.lti.cs.cmu.edu/Courses/11-724-desc.htm>**11-731 Machine Translation and Sequence-to-Sequence Models**

Spring: 12 units

Instructors: Graham Neubig. Prerequisites: This course has no official pre-requisites, although 11-711 "Algorithms for NLP" or 10-701 "Machine Learning" would be helpful. Course Description: Machine Translation and Sequence-to-Sequence Models is an introductory graduate-level course surveying the primary approaches and methods for developing systems to translate between human languages, or other sequential data. The main objective of the course is to obtain basic understanding and implementation skills for modern methods for MT and sequence transduction, including how to design models, how to learn the model parameters, how to search for the best output, and how to create training data. The course will focus on machine translation, but also briefly cover tasks such as dialog response generation, image caption generation, and others.

**11-737 Multilingual Natural Language Processing.**

Fall: 12 units

11737 Multilingual Natural Language Processing is an advanced graduate-level course on natural language processing techniques applicable to many languages. Students who take this course should be able to develop linguistically motivated solutions to core and applied NLP tasks for any language. This includes understanding and mitigating the difficulties posed by lack of data in low-resourced languages or language varieties, and the necessity to model particular properties of the language of interest such as complex morphology or syntax. The course will introduce modeling solutions to these issues such as multilingual or cross-lingual methods, linguistically informed NLP models, and methods for effectively bootstrapping systems with limited data or human intervention. The project work will involve building an end-to-end NLP pipeline in a language you don't know.

Course Website: <https://www.cs.cmu.edu/~leili/course/11737mnlp23fa/>**11-739 Designing Around Patents on Machine Learning and NLP Technology**

Spring: 12 units

This course uses Machine Learning and Natural Language Processing as vehicles to teach principles in designing software to avoid patents. After introducing students to the basics of patents, we investigate how to use students' software skills to design around patents that is, create new technology that avoids a patent while maintaining some, or even all, of the performance and value of the patented technology. Designing around a patent can be viewed as a puzzle that requires technical skill, understanding the business value of technology, knowledge of patents, and creativity. Students will also be able to help design patents that cannot be easily designed-around. Not only does this add a valuable new dimension to the student's skill set, the course material is organized as a vehicle for refining knowledge of ML and NLP techniques. We will practice by designing around patents on well-known ML and NLP algorithms. Students study the basic algorithm, learn the patents that cover those algorithms, and then experiment with software modifications that avoid the patent while preserving as much of the algorithm's performance as possible. In essence, students will view the presence of particular patents as a design constraint, akin to designing for limited memory, bandwidth, or processor speed. Students must have already taken courses in Natural Language Processing (e.g., 11-411) and Machine Learning (e.g., 10-315). Law of Computer Technology 17-562, 17-662, 17-762 or Patents, Licensing, and Innovation 19-473, 19-673 are helpful but not required.

**11-741 Machine Learning for Text and Graph-based Mining**

Fall and Spring: 12 units

This course studies the theory, design, and implementation of text-based information systems. The Information Retrieval core components of the course include statistical characteristics of text, representation of information needs and documents, several important retrieval models (Boolean, vector space, probabilistic, inference net, language modeling), clustering algorithms, automatic text categorization, and experimental evaluation. The software architecture components include design and implementation of high-capacity text retrieval and text filtering systems. A variety of current research topics are also covered, including cross-lingual retrieval, document summarization, machine learning, topic detection and tracking, and multi-media retrieval. Prerequisites: Programming and data-structures at the level of 15-212 or higher. Algorithms comparable to the undergraduate CS algorithms course (15-451) or higher. Basic linear algebra (21-241 or 21-341). Basic statistics (36-202) or higher.

**11-747 Neural Networks for NLP**

All Semesters: 12 units

Neural networks provide powerful new tools for modeling language, and have been used both to improve the state-of-the-art in a number of tasks and to tackle new problems that were not easy in the past. This class will start with a brief overview of neural networks, then spend the majority of the class demonstrating how to apply neural networks to natural language problems. Each section class will introduce a particular problem or phenomenon in natural language, describe why it is difficult to model, and demonstrate several models that were designed to tackle this problem. In the process of doing so, the class will cover different techniques that are useful in creating models, including handling variably sized and structured sentences, efficient handling of large data, semi-supervised and unsupervised learning, structured prediction, and multilingual modeling. There are no official pre-requisites, but a natural language processing course such as 11-411, 11-611, or 11-711, or other experience with implementing natural language processing models is highly recommended.

**11-751 Speech Recognition and Understanding**

All Semesters: 12 units

The technology to allow humans to communicate by speech with machines or by which machines can understand when humans communicate with each other is rapidly maturing. This course provides an introduction to the theoretical tools as well as the experimental practice that has made the field what it is today. We will cover theoretical foundations, essential algorithms, major approaches, experimental strategies and current state-of-the-art systems and will introduce the participants to ongoing work in representation, algorithms and interface design. This course is suitable for graduate students with some background in computer science and electrical engineering, as well as for advanced undergraduates. Prerequisites: Sound mathematical background, knowledge of basic statistics, good computing skills. No prior experience with speech recognition is necessary. This course is primarily for graduate students in LTI, CS, Robotics, ECE, Psychology, or Computational Linguistics. Others by prior permission of instructor.

**11-752 Speech II: Phonetics, Prosody, Perception and Synthesis**

Spring: 12 units

The goal of the course is to give the student basic knowledge from several fields that is necessary in order to pursue research in automatic speech processing. The course will begin with a study of the acoustic content of the speech signal. The students will use the spectrographic display to examine the signal and discover its variable properties. Phones in increasingly larger contexts will be studied with the goal of understanding coarticulation. Phonological rules will be studied as a contextual aid in understanding the spectrographic display. The spectrogram will then serve as a first introduction to the basic elements of prosody. Other displays will then be used to study the three parts of prosody: amplitude, duration, and pitch. Building on these three elements, the student will then examine how the three interact in careful and spontaneous speech. Next, the students will explore perception. Topics covered will be: physical aspects of perception, psychological aspects of perception, testing perception processes, practical applications of knowledge about perception. The second part of this course will cover all aspects of speech synthesis. Students need only have a basic knowledge of speech and language processing. Some degree of programming and statistical modelling will be beneficial, but not required. Taught every other year

**11-755 Machine Learning for Signal Processing**

Fall: 12 units

Signal Processing is the science that deals with extraction of information from signals of various kinds. This has two distinct aspects and #8212; characterization and categorization. Traditionally, signal characterization has been performed with mathematically-driven transforms, while categorization and classification are achieved using statistical tools. Machine learning aims to design algorithms that learn about the state of the world directly from data. A increasingly popular trend has been to develop and apply machine learning techniques to both aspects of signal processing, often blurring the distinction between the two. This course discusses the use of machine learning techniques to process signals. We cover a variety of topics, from data driven approaches for characterization of signals such as audio including speech, images and video, and machine learning methods for a variety of speech and image processing problems.

**11-761 Language and Statistics**

Fall: 12 units

Language technologies (search, text mining, information retrieval, speech recognition, machine translation, question answering, biological sequence analysis...) are at the forefront of this century's information revolution. In addition to their use of machine learning, these technologies rely centrally on classic statistical estimation techniques. Yet most CS and engineering undergraduate programs do not prepare students in this area beyond an introductory prob and amp;stats course. This course is designed to plug this hole. The goal of "Language and Statistics" is to ground the data-driven techniques used in language technologies in sound statistical methodology. We start by formulating various language technology problems in both an information theoretic framework (the source-channel paradigm) and a Bayesian framework (the Bayes classifier). We then discuss the statistical properties of words, sentences, documents and whole languages, and the computational formalisms used to represent language. These discussions naturally lead to specific concepts in statistical estimation. Topics include: Zipf's distribution and type-token curves; point estimators, Maximum Likelihood estimation, bias and variance, sparseness, smoothing and clustering; interpolation, shrinkage, and backoff; entropy, cross entropy and mutual information; decision tree models applied to language; latent variable models and the EM algorithm; hidden Markov models; exponential models and maximum entropy; semantic modeling and dimensionality reduction; probabilistic context-free grammars and syntactic language models. The course is designed for LTI and amp; SCS graduate students, but others are welcome. CS UG upperclassmen who've taken it have done well, though they found it challenging. The 11-661 version does not require the course project. Prerequisites: Strong quantitative aptitude. Comfort with basic UG-level probability. Some programming skill.

Course Website: <http://www.cs.cmu.edu/~roni/11761/>**11-762 Language and Statistics II**

Fall: 12 units

This course will cover modern empirical methods in natural language processing. It is designed for language technologies students who want to understand statistical methodology in the language domain, and for machine learning students who want to know about current problems and solutions in text processing. Students will, upon completion, understand how statistical modeling and learning can be applied to text, be able to develop and apply new statistical models for problems in their own research, and be able to critically read papers from the major related conferences (EMNLP and ACL). A recurring theme will be the tradeoffs between computational cost, mathematical elegance, and applicability to real problems. The course will be organized around methods, with concrete tasks introduced throughout. The course is designed for SCS graduate students. Prerequisite: Language and Statistics (11-761) or permission of the instructor. Recommended: Algorithms for Natural Language Processing (11-711), Machine Learning (15-681, 15-781, or 11-746). Prerequisite: 11-761

**11-763 Structured Prediction for Language and other Discrete Data**

Fall: 12 units

This course seeks to cover statistical modeling techniques for discrete, structured data such as text. It brings together content previously covered in Language and Statistics 2 (11-762) and Information Extraction (10-707 and 11-748), and aims to define a canonical set of models and techniques applicable to problems in natural language processing, information extraction, and other application areas. Upon completion, students will have a broad understanding of machine learning techniques for structured outputs, will be able to develop appropriate algorithms for use in new research, and will be able to critically read related literature. The course is organized around methods, with example tasks introduced throughout.

Course Website: <http://www.cs.cmu.edu/~nasmith/SPFLODD/>**11-776 Multimodal Affective Computing**

Fall: 12 units

Humans are highly social creatures and have evolved complex mechanisms for signaling information about their thoughts, feelings, and intentions (both deliberately and reflexively). In turn, humans have also evolved complex mechanisms for receiving these signals and inferring the thoughts, feelings, and intentions of others. Proper understanding of human behavior, in all its nuance, requires careful consideration and integration of verbal, vocal, and visual information. These communication dynamics have long been studied in psychology and other social sciences. More recently, the field of multimodal affective computing has sought to enhance these studies using techniques from computer science and artificial intelligence. Common topics of study in this field include affective states, cognitive states, personality, psychopathology, social processes, and communication. As such, multimodal affective computing has broad applicability in both scientific and applied settings ranging from medicine and education to robotics and marketing. The objectives of this course are: (1) To give an overview of the components of human behavior (verbal, vocal, and visual) and the computer science areas that measure them (NLP, speech processing, and computer vision) (2) To provide foundational knowledge of psychological constructs commonly studied in multimodal affective computing (e.g., emotion, personality, and psychopathology) (3) To provide practical instruction on using statistical tools to study research hypotheses (4) To provide information about computational predictive models that integrate multimodal information from the verbal, vocal, and visual modalities (5) To give students practical experience in the computational study of human behavior and psychological constructs through an in-depth course project

**11-777 Multimodal Machine Learning**

Fall: 12 units

Multimodal machine learning (MMML) is a vibrant multi-disciplinary research field which addresses some of the original goals of artificial intelligence by integrating and modeling multiple communicative modalities, including linguistic, acoustic and visual messages. With the initial research on audio-visual speech recognition and more recently with language vision projects such as image and video captioning, this research field brings some unique challenges for multimodal researchers given the heterogeneity of the data and the contingency often found between modalities. The course will present the fundamental mathematical concepts in machine learning and deep learning relevant to the five main challenges in multimodal machine learning: (1) multimodal representation learning, (2) translation and mapping, (3) modality alignment, (4) multimodal fusion and (5) co-learning. These include, but not limited to, multimodal auto-encoder, deep canonical correlation analysis, multi-kernel learning, attention models and multimodal recurrent neural networks. We will also review recent papers describing state-of-the-art probabilistic models and computational algorithms for MMML and discuss the current and upcoming challenges. The course will discuss many of the recent applications of MMML including multimodal affect recognition, image and video captioning and cross-modal multimedia retrieval. This is a graduate course designed primarily for PhD and research master students at LTI, MLD, CSD, HCL and RI; others, for example (undergraduate) students of CS or from professional master programs, are advised to seek prior permission of the instructor. It is required for students to have taken an introduction machine learning course such as 10-401, 10-601, 10-701, 11-663, 11-441, 11-641 or 11-741. Prior knowledge of deep learning is recommended."

Course Website: <https://piazza.com/cmu/fall2018/11777/home> (<https://piazza.com/cmu/fall2018/11777/home/>)

**11-792 Intelligent Information Systems Project - HEINZ STUDENTS ONLY**

Spring: 12 units

The Software Engineering for IS sequence combines classroom material and assignments in the fundamentals of software engineering (11-791) with a self-paced, faculty-supervised directed project (11-792). The two courses cover all elements of project design, implementation, evaluation, and documentation. Students may elect to take only 11-791; however, if both parts are taken, they should be taken in proper sequence. Prerequisite: 11-791. The course is required for VLIS students. Prerequisites: 15-393 or 11-791

**11-851 Talking to Robots**

Fall

Household robots need to move beyond simple programmed tasks like those a roomba, and become full-fledged digital assistants. A robotic agent that exists (physically) in the world, gains access to rich and personalized knowledge of its environment. How much do things weigh? What's fragile? Where you store the extra chocolates that you don't want anyone to find because they are just for you. Building an agent that can accomplish tasks requires the integration of a diverse set of technologies and engineering. Language models, SLAM, semantic mapping, task planning, understanding affordances, and end effector control. This course will cover both foundational works in grounding language to action and analyze (or reimplement) state-of-the-art Large Language Model based task planners.

Course Website: <https://talkingtorobots.com/11-851/>

**11-877 Advanced Topics in Multimodal Machine Learning**

Spring

This course is designed to be a graduate-level course covering recent research papers in multimodal machine learning, a vibrant multi-disciplinary research field which studies modeling, alignment, and fusion of heterogeneous data from multiple modalities, including language, vision, and acoustic. The course will focus on discussions and understanding of recent research papers in this field. Students are expected to have already taken 11-777 Multimodal Machine Learning course or have equivalent research experience (instructor approval required). The course is planned for 6 credit units. Optionally, students can register for 12 credit units, with the expectation to do a comprehensive research project as part of the semester. These course projects are expected to be done in teams, with the research topic to be in the realm of multimodal machine learning and pre-approved by the course instructors.

Prerequisite: 11-777 Min. grade C

**11-927 MIIS Capstone Project**

Fall: 36 units

The capstone project course is a group-oriented demonstration of student skill in one or more areas covered by the degree. Typically the result of the capstone project is a major software application. The capstone project course consists of two components. The classroom component guides students in project planning, team management, development of requirements and design specifications, and software tools for managing group-oriented projects. The lab component provides project-specific technical guidance and expertise, for example in the development of a question answering system, dialog, or sentiment analysis application. Thus, each project receives two types of supervision, often from two separate members of the faculty.

**Machine Learning Courses****10-301 Introduction to Machine Learning (Undergrad)**

Fall and Spring: 12 units

Machine Learning (ML) develops computer programs that automatically improve their performance through experience. This includes learning many types of tasks based on many types of experience, e.g. spotting high-risk medical patients, recognizing speech, classifying text documents, detecting credit card fraud, or driving autonomous vehicles. 10301 covers all or most of: concept learning, decision trees, neural networks, linear learning, active learning, estimation and amp; the bias-variance tradeoff, hypothesis testing, Bayesian learning, the MDL principle, the Gibbs classifier, Naive Bayes, Bayes Nets and amp; Graphical Models, the EM algorithm, Hidden Markov Models, K-Nearest-Neighbors and nonparametric learning, reinforcement learning, bagging, boosting and discriminative training. Grading will be based on weekly or biweekly assignments (written and/or programming), a midterm, a final exam. 10301 is recommended for undergraduates who are not SCS majors. (SCS majors should instead take 10315.) Prerequisites (strictly enforced): strong quantitative aptitude, college probability and amp; statistics course, and programming proficiency. For learning to apply ML practically and amp; effectively, without the above prerequisites, consider 11344/05834 instead. You can find a link to the Intro to ML course comparison page, which includes a self-assessment exam to help you choose which Intro to ML course to take, in the Course URL field. Prerequisites: 15-122 Min. grade C and (21-128 Min. grade C or 15-151 Min. grade C or 21-127 Min. grade C) and (36-217 Min. grade C or 36-218 Min. grade C or 15-359 Min. grade C or 36-219 Min. grade C or 36-225 Min. grade C or 36-235 Min. grade C or 21-325 Min. grade C)

Course Website: <http://mlcourse.org>

**10-315 Introduction to Machine Learning (SCS Majors)**

Fall and Spring: 12 units

Machine learning is subfield of computer science with the goal of exploring, studying, and developing learning systems, methods, and algorithms that can improve their performance with learning from data. This course is designed to give undergraduate students a one-semester-long introduction to the main principles, algorithms, and applications of machine learning and is specifically designed for the SCS undergrad majors. The topics of this course will be in part parallel with those covered in the graduate machine learning courses (10-715, 10-701, 10-601), but with a greater emphasis on applications and case studies in machine learning. After completing the course, students will be able to: \*select and apply an appropriate supervised learning algorithm for classification problems (e.g., naive Bayes, perceptron, support vector machine, logistic regression). \*select and apply an appropriate supervised learning algorithm for regression problems (e.g., linear regression, ridge regression). \*recognize different types of unsupervised learning problems, and select and apply appropriate algorithms (e.g., clustering, linear and nonlinear dimensionality reduction). \*work with probabilities (Bayes rule, conditioning, expectations, independence), linear algebra (vector and matrix operations, eigenvectors, SVD), and calculus (gradients, Jacobians) to derive machine learning methods such as linear regression, naive Bayes, and principal components analysis. \*understand machine learning principles such as model selection, overfitting, and underfitting, and techniques such as cross-validation and regularization. \*implement machine learning algorithms such as logistic regression via stochastic gradient descent, linear regression (using a linear algebra toolbox), perceptron, or k-means clustering. \*run appropriate supervised and unsupervised learning algorithms on real and synthetic data sets and interpret the results. Prerequisites: 15-122 Min. grade C and (15-151 Min. grade C or 21-128 Min. grade C or 21-127 Min. grade C) and (36-218 Min. grade C or 15-359 Min. grade C or 21-325 Min. grade C or 36-219 Min. grade C or 36-235 Min. grade C or 36-217 Min. grade C or 36-225 Min. grade C or 15-259 Min. grade C) and (21-242 Min. grade C or 21-241 Min. grade C or 21-240 Min. grade C)

Course Website: <https://goo.gl/mmR2eL> (<https://goo.gl/mmR2eL/>)

**10-335 Art and Machine Learning**

Intermittent: 12 units

Ars, the Latin origin of the word art, means Art and Science. These two fields, which have been separated for a long time, are joining back together in many areas. One of those junctions is where Art and Machine Learning meet. Art in recent years has been moving forward along with the rise of new technologies and scientific discoveries. Machine Learning (ML) is one of the most cutting edge advancements in Computer Science. The popularity and accessibility of frameworks such as Google's Deep Dream system, Pikazo the neural style transfer, Kulitta AI Music Generation Framework, Deep Mind's WaveNet, Sony's Flow Machines, and recurrent neural network based language models brought great attention to the marriage of Art and ML methods. The number of ML applications that mimic famous artworks, e.g. The Next Rembrandt project, or even create original artworks such as the robot artist TAIIDA's paintings, is rapidly growing. Increasing number of artists are also attempting to use ML methods in their artworks. This course is project-based and aims to introduce the crossroad of Art and Machine Learning to the broad range of students including both Art and Computer Science majors. We will offer the knowledge of examples, technologies, and issues that connect Art and Machine Learning to the students. Students will study example codes and produce creative applications/artworks using ML methods. Students do not need to have pre-existing knowledge of Machine Learning or experience of Art practice. Students are required to have basic understanding of Python and be open-minded, for example, open to learn the necessary mathematical background and open to discussions on conceptual development and artistic value of their projects.

**10-403 Deep Reinforcement Learning & Control**

Spring: 12 units

This course brings together many disciplines of Artificial Intelligence (including computer vision, robot control, reinforcement learning, language understanding) to show how to develop intelligent agents that can learn to sense the world and learn to act by imitating others, maximizing sparse rewards, and/or satisfying their curiosity.

Prerequisites: 10-401 Min. grade C or 10-601 Min. grade C or 10-701 Min. grade C or 10-315 Min. grade C or 10-301 Min. grade C

Course Website: <https://cmudeeprl.github.io/Spring202010403website/>

**10-405 Machine Learning with Large Datasets (Undergraduate)**

Spring: 12 units

Large datasets pose difficulties across the machine learning pipeline. They are difficult to visualize and introduce computational, storage, and communication bottlenecks during data pre-processing and model training. Moreover, high capacity models often used in conjunction with large datasets introduce additional computational and storage hurdles during model training and inference. This course is intended to provide a student with the mathematical, algorithmic, and practical knowledge of issues involving learning with large datasets. Among the topics considered are: data cleaning, visualization, and pre-processing at scale; principles of parallel and distributed computing for machine learning; techniques for scalable deep learning; analysis of programs in terms of memory, computation, and (for parallel methods) communication complexity; and methods for low-latency inference. The class will include programming and written assignments to provide hands-on experience applying machine learning at scale. An introductory machine learning course is a prerequisite. A strong background in programming will also be necessary; suggested prerequisites include 15-210, 15-214, or equivalent. Students are expected to be familiar with Python or learn it during the course.

Prerequisites: (15-214 or 15-210 or 15-211 or 17-214) and (10-715 or 10-701 or 10-601 or 10-401 or 10-315 or 10-301)

Course Website: <https://10605.github.io/>

**10-414 Deep Learning Systems: Algorithms and Implementation**

Intermittent: 12 units

The goal of this course is to provide students an understanding and overview of the "full stack" of deep learning systems, ranging from the high-level modeling design of modern deep learning systems, to the basic implementation of automatic differentiation tools, to the underlying device-level implementation of efficient algorithms. Throughout the course, students will design and build from scratch a complete deep learning library, capable of efficient GPU-based operations, automatic differentiation of all implemented functions, and the necessary modules to support parameterized layers, loss functions, data loaders, and optimizers. Using these tools, students will then build several state-of-the-art modeling methods, including convolutional networks for image classification and segmentation, recurrent networks and self-attention models for sequential tasks such as language modeling, and generative models for image generation.

Prerequisites: (15-513 Min. grade C or 15-213 Min. grade C) and (21-241 Min. grade C or 21-240 Min. grade C) and (21-127 Min. grade C or 15-151 Min. grade C or 21-128 Min. grade C) and (10-715 Min. grade C or 10-601 Min. grade C or 10-301 Min. grade C or 10-701 Min. grade C or 10-315 Min. grade C)

**10-417 Intermediate Deep Learning**

Fall: 12 units

Building intelligent machines that are capable of extracting meaningful representations from data lies at the core of solving many AI related tasks. In the past decade, researchers across many communities, from applied statistics to engineering, computer science and neuroscience, have developed deep models that are composed of several layers of nonlinear processing. An important property of these models is that they can learn useful representations by re-using and combining intermediate concepts, allowing these models to be successfully applied in a wide variety of domains, including visual object recognition, information retrieval, natural language processing, and speech perception. The goal of this course is to introduce students to both the foundational ideas and the recent advances in deep learning. The first part of the course will focus on supervised learning, including neural networks, back-propagation algorithm, convolutional models, recurrent neural networks, and their extensions with applications to image recognition, video analysis, and language modelling. The second part of the course will cover unsupervised learning, including variational autoencoders, sparse-coding, Boltzmann machines, and generative adversarial networks. This course will assume a reasonable degree of mathematical maturity and will require strong programming skills.

Prerequisites: 10-701 Min. grade C or 10-601 Min. grade C or 10-715 Min. grade C or 10-301 Min. grade C or 10-315 Min. grade C

Course Website: <https://deeplearning-cmu-10417.github.io>

**10-418 Machine Learning for Structured Data**

Intermittent: 12 units

A key challenge in machine learning is that of structured prediction: taking unstructured data as input and producing a structured output. Structured prediction problems abound throughout application areas such as natural language processing, speech processing, computational biology, computer vision, healthcare, and many others. In this course, we will study modern approaches to structured prediction building on probabilistic graphical models, deep learning, and search. The course will focus on three key aspects: models, inference, and learning. The models we consider will focus on both generative and discriminative models such as Bayesian networks, Markov random fields (MRFs), conditional random fields (CRFs), and deep neural networks including convolutional neural networks (CNNs) and recurrent neural networks (RNNs) and #8212; as well as hybrids of graphical models and neural networks. The course will explore approaches to exact and approximate inference: junction tree algorithm, approximate marginal inference by Markov chain Monte Carlo (MCMC) and variational methods, approximate MAP inference by integer linear programming (ILP) and search. We will explore unsupervised, semi-supervised, and supervised learning using different formulations of the learning problem: MLE, Bayesian inference, structured perceptron, M3Ns, learning to search, and autoencoders. Covered applications will include machine translation, speech recognition, DNA sequence analysis, scene understanding, medical diagnosis. This course is cross-listed as 10-418 and 10-618; students registered for 10-618 will do a course project.

Prerequisites: 10-301 Min. grade C or 10-315 Min. grade C or 10-401 Min. grade C or 10-701 Min. grade C or 10-601 Min. grade C or 10-715 Min. grade C

**10-422 Foundations of Learning, Game Theory, and Their Connections**

Intermittent: 12 units

In the past decades, researchers have discovered a number of important and deep connections between machine learning theory and algorithmic game theory. This course will explore these connections, both introducing fundamental topics in each area and describing how ideas from each area can shed light on the other.

Prerequisites: (36-217 Min. grade C or 15-259 Min. grade C or 21-325 Min. grade C or 36-219 Min. grade C or 36-218 Min. grade C or 36-225 Min. grade C) and 15-251 Min. grade C and (21-241 Min. grade C or 21-242 Min. grade C or 21-240 Min. grade C)

**10-425 Introduction to Convex Optimization**

Intermittent: 12 units

As machine learning grows in prominence, so also has optimization become a mainstay for machine learning, particularly techniques for convex optimization. Most learning problems are formulated as optimization of some objective function, sometimes subject to constraints. This course explores the optimization algorithms used to solve these machine learning problems. We characterize the properties of the optimization problems that enable these techniques to be efficient (e.g. convexity, smoothness, linearity, separability) as well as properties that inhibit efficient optimization (e.g. nonconvexity). Core topics include first order methods (gradient descent, subgradient methods, proximal and stochastic gradient descent), duality and linear programming, and second-order/quasi-Newton methods. We also consider advanced techniques ranging from those that have spurred the growth of deep learning (e.g. adaptive gradient methods, momentum) and those that enable large-scale distributed optimization. The course will focus both on theory and practical applications, frequently drawing motivation from examples in machine learning. The course is designed so that a machine learning (ML) course could be taken after or before this one; ML is not a prerequisite. Students will gain the tools to both implement and analyze modern optimization techniques.

Prerequisites: (21-241 Min. grade C or 21-240 Min. grade C) and (21-325 Min. grade C or 36-219 Min. grade C or 36-217 Min. grade C or 15-359 Min. grade C) and (21-254 Min. grade C or 11-485 Min. grade C or 21-256 Min. grade C or 21-259 Min. grade C or 10-315 Min. grade C or 10-301 Min. grade C) and (21-127 Min. grade C or 15-151 Min. grade C) and 15-122 Min. grade C

Course Website: <https://www.cs.cmu.edu/~mgormley/courses/10425>  
(<https://www.cs.cmu.edu/~mgormley/courses/10425/>)

**10-500 Senior Research Project**

All Semesters

Register for this course if you are minoring in Machine Learning. This course is intended for research with a faculty member that would count towards the minor.

**10-520 Independent Study**

All Semesters

Independent Study intended to work on research with a Machine Learning faculty member.

**10-600 Mathematical background for Machine Learning**

Fall and Spring: 12 units

This course provides a place for students to practice the necessary mathematical background for further study in machine learning and #8212; particularly for taking 10-601 and 10-701. Topics covered include probability, linear algebra (inner product spaces, linear operators), multivariate differential calculus, optimization, and likelihood functions. The course assumes some background in each of the above, but will review and give practice in each. (It does not provide from-scratch coverage of all of the above, which would be impossible in a course of this length.) Some coding will be required: the course will provide practice with translating the above mathematical concepts into concrete programs. This course supersedes the two mini-courses 10-606 and 10-607.

**10-601 Introduction to Machine Learning (Master's)**

Fall and Spring: 12 units

Machine Learning (ML) develops computer programs that automatically improve their performance through experience. This includes learning many types of tasks based on many types of experience, e.g. spotting high-risk medical patients, recognizing speech, classifying text documents, detecting credit card fraud, or driving autonomous vehicles. 10601 covers all or most of: concept learning, decision trees, neural networks, linear learning, active learning, estimation and amp; the bias-variance tradeoff, hypothesis testing, Bayesian learning, the MDL principle, the Gibbs classifier, Naive Bayes, Bayes Nets and amp; Graphical Models, the EM algorithm, Hidden Markov Models, K-Nearest-Neighbors and nonparametric learning, reinforcement learning, bagging, boosting and discriminative training. Grading will be based on weekly or biweekly assignments (written and/or programming), a midterm, a final exam, and possibly a project (details may vary depending on the section). 10601 is recommended for quantitative master's and amp; PhD students outside MLD. Prerequisites (strictly enforced): strong quantitative aptitude, college prob and amp; stats course, and programming proficiency. For learning to apply ML practically and amp; effectively, without the above prerequisites, consider 11344/05834 instead. If you are unsure whether you have sufficient mathematical background to do well in this course, you should consider taking the minis 10-606 and 10-607 Mathematical and Computational Foundations for Machine Learning. You can find a link to the Intro to ML course comparison page, which includes a self-assessment exam to help you choose which Intro to ML course to take, in the Course URL field.

Prerequisites: 15-122 Min. grade C and (21-127 Min. grade C or 15-151 Min. grade C or 21-128 Min. grade C) and (15-259 Min. grade C or 36-217 Min. grade C or 36-218 Min. grade C or 36-235 Min. grade C or 36-225 Min. grade C or 21-325 Min. grade C or 36-219 Min. grade C or 15-359 Min. grade C)

Course Website: <http://mlcourse.org>

**10-605 Machine Learning with Large Datasets**

Fall and Spring: 12 units

Large datasets are difficult to work with for several reasons. They are difficult to visualize, and it is difficult to understand what sort of errors and biases are present in them. They are computationally expensive to process, and often the cost of learning is hard to predict - for instance, an algorithm that runs quickly in a dataset that fits in memory may be exorbitantly expensive when the dataset is too large for memory. Large datasets may also display qualitatively different behavior in terms of which learning methods produce the most accurate predictions. This course is intended to provide a student practical knowledge of, and experience with, the issues involving large datasets. Among the issues considered are: scalable learning techniques, such as streaming machine learning techniques; parallel infrastructures such as map-reduce; practical techniques for reducing the memory requirements for learning methods, such as feature hashing and Bloom filters; and techniques for analysis of programs in terms of memory, disk usage, and (for parallel methods) communication complexity. The class will include programming assignments, and a one-month short project chosen by the student. The project will be designed to compare the scalability of variant learning algorithms on datasets. An introductory course in machine learning, like 10-601 or 10-701, is a prerequisite or a co-requisite. If you plan to take this course and 10-601 concurrently please tell the instructor. The course will include several substantial programming assignments, so an additional prerequisite is 15-211, or 15-214, or comparable familiarity with Java and good programming skills. Prerequisites: (15-214 or 15-210 or 17-214 or 15-211) and (10-715 or 10-301 or 10-315 or 10-401 or 10-701 or 10-601)

Course Website: <https://10605.github.io/>

**10-606 Mathematical Foundations for Machine Learning**

Fall: 6 units

This course provides a place for students to practice the necessary mathematical background for further study in machine learning. Topics covered include probability (random variables, modeling with continuous and discrete distributions), linear algebra (inner product spaces, linear operators), and multivariate differential calculus (partial derivatives, matrix differentials). The course assumes some background in each of the above, but will review and give practice in each. (It does not provide from-scratch coverage of all of the above, which would be impossible in a course of this length.) Some coding will be required: the course will provide practice with translating the above mathematical concepts into concrete programs. This course is one of two minis intended to prepare students for further study in machine learning and #8212; particularly for taking 10-601 and 10-701. One of the courses 10-606 focuses on mathematical background, and the other course 10-607 focuses on computational background. Most students take both mini courses, but this is not required. 10-606 is not a prerequisite of 10-607.

**10-607 Computational Foundations for Machine Learning**

Fall: 6 units

This course provides a place for students to practice the necessary computational background for further study in machine learning. Topics covered include computational complexity, analysis of algorithms, proof techniques, optimization, dynamic programming, recursion, and data structures. The course assumes some background in each of the above, but will review and give practice in each. (It does not provide from-scratch coverage of all of the above, which would be impossible in a course of this length.) Some coding will be required: the course will provide practice with translating the above computational concepts into concrete programs. This course is one of two minis intended to prepare students for further study in machine learning and #8212; particularly for taking 10-601 and 10-701. One of the courses 10-606 focuses on mathematical background, and the other course 10-607 focuses on computational background. Most students take both mini courses, but this is not required. 10-606 is not a prerequisite of 10-607.

**10-608 Conversational Machine Learning**

Intermittent: 12 units

Machine Learning today is largely about finding patterns in large amounts of data. But as personal devices that interact with us in natural language become ubiquitous (e.g., Siri, Google Now), they open an amazing possibility of letting users teach machines in natural language, similar to how we teach each other. Conversation, as an interface to machine learning systems, opens a new paradigm that both unifies several existing machine learning paradigms (e.g., active learning, supervised learning), but also brings a unique set of advantages and challenges that lie at the intersection of machine learning and natural language processing. This course will be structured as a well-defined mini-challenge (project) course. We will present you with several well-defined open problems and provide you with recently collected datasets that can get you started immediately! But you will be free to define your own problem using that data as well, or come up with your own problem entirely. There are no other constraints, and since this is a new area of research, you can (and should) be creative and as crazy in coming up with methods to tackle them. At the same time, we will provide guidance via readings and class-based hacking sessions. This course is a great way to get introduced to open problems in a collaborative and structured environment. Challenges Building a classifier with zero examples. Telling sequence to sequence models about their mistakes Letting machine learning models ask questions

Prerequisites: 10-715 Min. grade C or 10-401 Min. grade C or 10-701 Min. grade C or 10-601 Min. grade C

**10-613 Machine Learning Ethics and Society**

Intermittent: 12 units

The practice of Machine Learning (ML) increasingly involves making choices that impact real people and society at large. This course covers an array of ethical, societal, and policy considerations in applying ML tools to high-stakes domains, such as employment, education, lending, criminal justice, medicine, and beyond. We will discuss: (1) the pathways through which ML can lead to or amplify problematic decision-making practices (e.g., those exhibiting discrimination, inscrutability, invasion of privacy, and beyond); (2) recent technological methods and remedies to capture and alleviate these concerns; and (3) the scope of applicability and limitations of technological remedies in the context of several contemporary application domains. The course's primary goals are: (a) to raise awareness about the social, ethical, and policy implications of ML, and (b) to prepare students to critically analyze these issues as they emerge in the ever-expanding use of ML in socially consequential domains.

Prerequisites: 10-301 or 10-315 or 10-715 or 10-601 or 10-701

Course Website: <http://www.cs.cmu.edu/~hheidari/mles-spring-23.html>

**10-617 Intermediate Deep Learning**

Fall: 12 units

Building intelligent machines that are capable of extracting meaningful representations from data lies at the core of solving many AI related tasks. In the past decade, researchers across many communities, from applied statistics to engineering, computer science and neuroscience, have developed deep models that are composed of several layers of nonlinear processing. An important property of these models is that they can learn useful representations by re-using and combining intermediate concepts, allowing these models to be successfully applied in a wide variety of domains, including visual object recognition, information retrieval, natural language processing, and speech perception. The goal of this course is to introduce students to both the foundational ideas and the recent advances in deep learning. The first part of the course will focus on supervised learning, including neural networks, back-propagation algorithm, convolutional models, recurrent neural networks, and their extensions with applications to image recognition, video analysis, and language modelling. The second part of the course will cover unsupervised learning, including variational autoencoders, sparse-coding, Boltzmann machines, and generative adversarial networks. This course will assume a reasonable degree of mathematical maturity and will require strong programming skills. Prerequisites: 10-601 Min. grade C or 10-701 Min. grade C or 10-315 Min. grade C or 10-301 Min. grade C or 10-715 Min. grade C

Course Website: <https://deeplearning-cmu-10417.github.io>

**10-618 Machine Learning for Structured Data**

Intermittent: 12 units

A key challenge in machine learning is that of structured prediction: taking unstructured data as input and producing a structured output. Structured prediction problems abound throughout application areas such as natural language processing, speech processing, computational biology, computer vision, healthcare, and many others. In this course, we will study modern approaches to structured prediction building on probabilistic graphical models, deep learning, and search. The course will focus on three key aspects: models, inference, and learning. The models we consider will focus on both generative and discriminative models such as Bayesian networks, Markov random fields (MRFs), conditional random fields (CRFs), and deep neural networks including convolutional neural networks (CNNs) and recurrent neural networks (RNNs) and #8212; as well as hybrids of graphical models and neural networks. The course will explore approaches to exact and approximate inference: junction tree algorithm, approximate marginal inference by Markov chain Monte Carlo (MCMC) and variational methods, approximate MAP inference by integer linear programming (ILP) and search. We will explore unsupervised, semi-supervised, and supervised learning using different formulations of the learning problem: MLE, Bayesian inference, structured perceptron, M3Ns, learning to search, and autoencoders. Covered applications will include machine translation, speech recognition, DNA sequence analysis, scene understanding, medical diagnosis. This course is cross-listed as 10-418 and 10-618; students registered for 10-618 will do a course project.

Prerequisites: 10-601 Min. grade C or 10-401 Min. grade C or 10-315 Min. grade C or 10-301 Min. grade C or 10-715 Min. grade C or 10-701 Min. grade C

**10-701 Introduction to Machine Learning (PhD)**

Fall and Spring: 12 units

Machine learning studies the question "How can we build computer programs that automatically improve their performance through experience?" This includes learning to perform many types of tasks based on many types of experience. For example, it includes robots learning to better navigate based on experience gained by roaming their environments, medical decision aids that learn to predict which therapies work best for which diseases based on data mining of historical health records, and speech recognition systems that learn to better understand your speech based on experience listening to you. This course is designed to give PhD students a thorough grounding in the methods, mathematics and algorithms needed to do research and applications in machine learning. Students entering the class with a pre-existing working knowledge of probability, statistics and algorithms will be at an advantage, but the class has been designed so that anyone with a strong numerate background can catch up and fully participate. You can evaluate your ability to take the course via a self-assessment exam that will be made available to you after you register. If you are interested in this topic, but are not a PhD student, or are a PhD student not specializing in machine learning, you might consider the master's level course on Machine Learning, 10-601." This class may be appropriate for MS and undergrad students who are interested in the theory and algorithms behind ML. If you are unsure whether you have sufficient mathematical background to do well in this course, you should consider taking the minis 10-606/10-607 Mathematical Background for Machine Learning. You can find a link to the Intro to ML course comparison page, which includes a self-assessment exam to help you choose which Intro to ML course to take, in the Course URL field.

Prerequisites: 15-122 Min. grade C and (15-151 Min. grade C or 21-128 Min. grade C or 21-127 Min. grade C) and (15-259 Min. grade C or 15-359 Min. grade C or 36-218 Min. grade C or 36-225 Min. grade C or 21-325 Min. grade C or 36-217 Min. grade C or 36-219 Min. grade C)

**10-702 Statistical Machine Learning**

Spring: 12 units

Statistical Machine Learning is a second graduate level course in advanced machine learning, assuming that students have taken Machine Learning (10-701) or Advanced Machine Learning (10-715), and Intermediate Statistics (36-705). The term "statistical" in the title reflects the emphasis on statistical theory and methodology. This course is mostly focused on methodology and theoretical foundations. It treats both the "art" of designing good learning algorithms and the "science" of analyzing an algorithm's statistical properties and performance guarantees. Theorems are presented together with practical aspects of methodology and intuition to help students develop tools for selecting appropriate methods and approaches to problems in their own research. Though computation is certainly a critical component of what makes a method successful, it will not receive the same central focus as methodology and theory. We will cover topics in statistical theory that are important for researchers in machine learning, including consistency, minimax estimation, and concentration of measure. We will also cover statistical topics that may not be covered in as much depth in other machine learning courses, such as nonparametric density estimation, nonparametric regression, and Bayesian estimation. Prerequisites: (10-705 or 36-705) and (10-701 or 10-715)

Course Website: <http://www.stat.cmu.edu/~larry/=sml/>

**10-703 Deep Reinforcement Learning & Control**

Spring: 12 units

This course will cover latest advances in Reinforcement Learning and Imitation learning. This is a fast developing research field and an official textbook is available only for about one fourth of the course material. The rest will be taught from recent research papers. This course brings together many disciplines of Artificial Intelligence to show how to develop intelligent agent that can learn to sense the world and learn to act imitating others or maximizing sparse rewards Particular focus will be given in incorporating visual sensory input and learning suitable visual state representations.

Prerequisites: 10-601 Min. grade B or 10-701 Min. grade B or 10-715 Min. grade B or 10-301 Min. grade B or 10-315 Min. grade B or 10-401 Min. grade B

Course Website: <https://cmudeeprl.github.io/703website/>

**10-707 Advanced Deep Learning**

Fall and Spring: 12 units

Models that are capable of extracting complex, hierarchical representations from high-dimensional data lie at the core of solving many ML and AI domains, such as visual object recognition, information retrieval, natural language processing, and speech perception. While the usefulness of such deep learning techniques is undisputed, our understanding of them is still in many ways nascent. The goal of this course is to introduce students to recent and exciting developments (both theoretical and practical) in these methods. This is an advanced graduate course, designed for Masters and Ph.D. level students, and will assume a substantial degree of mathematical maturity. Prerequisite: ML: 10-701 or 10-715, and strong programming skills.

Prerequisites: 10-401 Min. grade C or 10-315 Min. grade C or 10-715 Min. grade C or 10-701 Min. grade C or 10-601 Min. grade C

**10-708 Probabilistic Graphical Models**

Spring: 12 units

Many of the problems in artificial intelligence, statistics, computer systems, computer vision, natural language processing, and computational biology, among many other fields, can be viewed as the search for a coherent global conclusion from local information. The probabilistic graphical models framework provides an unified view for this wide range of problems, enabling efficient inference, decision-making and learning in problems with a very large number of attributes and huge datasets. This graduate-level course will provide you with a strong foundation for both applying graphical models to complex problems and for addressing core research topics in graphical models. The class will cover three aspects: The core representation, including Bayesian and Markov networks, and dynamic Bayesian networks; probabilistic inference algorithms, both exact and approximate; and, learning methods for both the parameters and the structure of graphical models. Students entering the class should have a pre-existing working knowledge of probability, statistics, and algorithms, though the class has been designed to allow students with a strong numerate background to catch up and fully participate. It is expected that after taking this class, the students should have obtain sufficient working knowledge of multi-variate probabilistic modeling and inference for practical applications, should be able to formulate and solve a wide range of problems in their own domain using GM, and can advance into more specialized technical literature by themselves. Students are required to have successfully completed 10701 or 10715, or an equivalent class.

Prerequisites: 10-601 Min. grade C or 10-315 Min. grade C or 10-301 Min. grade C or 10-715 Min. grade C or 10-701 Min. grade C

Course Website: <https://andrejristeski.github.io/10708-2/>

**10-715 Advanced Introduction to Machine Learning**

Fall: 12 units

Machine Learning is the primary pillar that Artificial Intelligence is built upon. This course is designed for Ph.D. students whose primary field of study is machine learning, and who intend to make machine learning methodological research a main focus of their thesis. It will give students a thorough grounding in the algorithms, mathematics, theories, and insights needed to do in-depth research and applications in machine learning. The topics of this course will in part parallel those covered in the general PhD-level machine learning course (10-701), but with a greater emphasis on depth in theory. Students entering the class are expected to have a pre-existing strong working knowledge of linear algebra, probability, statistics, and algorithms. The course will also involve programming in Python. If you are interested in this topic, but do not have the required background or are not planning to work on a PhD thesis with machine learning as the main focus, you might consider the general PhD-level Machine Learning course (10-701) or the Masters-level Machine Learning course (10-601). You can find a webpage to the Intro to ML course comparison page, which includes a self-assessment exam to help you choose which Intro to ML course to take, in the Course URL field.

Prerequisites: 15-122 Min. grade C and (21-127 Min. grade C or 15-151 Min. grade C or 21-128 Min. grade C) and (15-359 Min. grade C or 36-225 Min. grade C or 36-217 Min. grade C or 21-325 Min. grade C or 36-218 Min. grade C or 15-259 Min. grade C)

Course Website: <https://www.cs.cmu.edu/~nihars/teaching/10715-Fa23/index.html> (<https://www.cs.cmu.edu/~nihars/teaching/10715-Fa23/>)



**10-716 Advanced Machine Learning: Theory and Methods**

Spring: 12 units

Advanced Machine Learning is a graduate level course introducing the theoretical foundations of modern machine learning, as well as advanced methods and frameworks used in modern machine learning. The course assumes that students have taken graduate level introductory courses in machine learning (Introduction to Machine Learning, 10-701 or 10-715), as well as Statistics (Intermediate Statistics, 36-700 or 36-705). The course treats both the art of designing good learning algorithms, as well as the science of analyzing an algorithm's computational and statistical properties and performance guarantees. Theorems are presented together with practical aspects of methodology and intuition to help students develop tools for selecting appropriate methods and approaches to problems in their own research. We will cover advanced machine learning methods such as nonparametric and deep compositional approaches to density estimation and regression; advanced theory such as fundamentals of clustering, classification, boosting; theory and methods at the intersection of statistical and computational efficiency; as well as vignettes of theoretical results on some hot topics such as robustness and explainability.

Prerequisites: (10-701 Min. grade C or 10-715 Min. grade C) and (36-705 Min. grade C or 36-700 Min. grade C)

Course Website: <http://www.cs.cmu.edu/~pradeep/716/>

**10-725 Convex Optimization**

Intermittent: 12 units

Nearly every problem in machine learning can be formulated as the optimization of some function, possibly under some set of constraints. This universal reduction may seem to suggest that such optimization tasks are intractable. Fortunately, many real world problems have special structure, such as convexity, smoothness, separability, etc., which allow us to formulate optimization problems that can often be solved efficiently. This course is designed to give a graduate-level student a thorough grounding in the formulation of optimization problems that exploit such structure, and in efficient solution methods for these problems. The main focus is on the formulation and solution of convex optimization problems, though we will discuss some recent advances in nonconvex optimization. These general concepts will also be illustrated through applications in machine learning and statistics. Students entering the class should have a pre-existing working knowledge of algorithms, though the class has been designed to allow students with a strong numerate background to catch up and fully participate. Though not required, having taken 10-701 or an equivalent machine learning or statistical modeling class is strongly encouraged, as we will use applications in machine learning and statistics to demonstrate the concepts we cover in class. Students will work on an extensive optimization-based project throughout the semester.

Prerequisites: (21-341 Min. grade C or 21-242 Min. grade C or 10-606 Min. grade C or 21-671 Min. grade C or 21-240 Min. grade C) and (21-268 Min. grade C or 21-259 Min. grade C or 21-254 Min. grade C) and (21-325 Min. grade C or 36-219 Min. grade C or 36-225 Min. grade C or 36-218 Min. grade C or 15-259 Min. grade C or 15-359 Min. grade C or 36-217 Min. grade C)

Course Website: <https://www.stat.cmu.edu/~siva/teaching/725/>

**10-737 Creative AI**

Intermittent

Artificial intelligence (AI) systems now generate authentic paintings, compose music pieces, and find out-of-box solutions to real-life problems in our world. Creativity, which was considered to be a moon shot for AI, does not seem to be too far any more. Is that true? Are we close to see creative AI? The answer is yes and no. We are moving closer with meaningful developments in Machine Learning, however there are several questions to be explored further to achieve the creative AI. What kind of creativity we want to represent? How do we translate creativity into what machines can understand? How do we design ML algorithms to be more creative? This course is where we explore these questions through seminars and projects. Our goal is to design computational models that present the very possibility of the creative AI. The instructors who are specialized in Machine Learning Art and Robotics lead this course together. We introduce related examples and possible methods including multi-modal data-driven learning, learning from demonstration, and combined learning from data and human demonstrations. Students are welcome to bring in their expertise and passion from diverse backgrounds to explore this topic together.

Course Website: <http://kangeunsu.com/creativeai19f/>

**10-745 Scalability in Machine Learning**

Fall: 12 units

The goal of this course is to provide a survey into some of the recent advances in the theory and practice of dealing with scalability issues in machine learning. We will investigate scalability issues along the following dimensions: Challenges with i) large datasets, ii) high-dimensions, and iii) complex data structure. The course is intended to prepare students to write research papers about scalability issues in machine learning. This is an advanced-level, fast-paced course that requires students to already have a solid understanding of machine learning (e.g. by taking an intro to ML class), good programming skills in Python, and being comfortable with dealing with abstract mathematical concepts and reading research papers. The course will have significant overlap with 10-405/605/805, but 10-745 will be faster-paced and go deeper into the theoretical investigations of the methods. Some of the classes will be flipped that will require students to watch a video lecture or read a research paper before the class, and the content will be discussed during the class time. The class will include a course project, HW assignments, and two-in class exams.

Prerequisites: 10-301 Min. grade B or 10-315 Min. grade B or 10-401 Min. grade B or 10-701 Min. grade B or 10-715 Min. grade B or 10-601 Min. grade B

**10-777 Historical Advances in Machine Learning**

Intermittent: 12 units

We will read (before class) and discuss (in class) a variety of historically important papers in ML (and to some extent AI). Not all of these were initially published in the ML/AI literature (eg: Bellman in math, VC in probability, bandits in statistics, fuzzy sets in control, optimization work in OR, etc, but now play central roles in ML and/or AI). Since "historical" is always ambiguous, we're going to go with "presented/published before the instructor was born" as a definition (pre-1988). While the content of the paper will be the primary focus, we will also attempt to understand the research context in which the paper was written. For example, what questions were other researchers asking at the time? Was the paper immediately recognized as a breakthrough or did it take a long time? Do we view the contents of the paper today as "obvious in hindsight" or is there still a lot of material in the paper that is nontrivial and even surprising or underappreciated? Who was the author, were they already relatively well known when they wrote the paper, or was it the paper itself that made them famous? What else did these authors work on before/after the paper?

Prerequisites: 10-715 Min. grade C or 10-701 Min. grade C or 10-601 Min. grade C or 10-301 Min. grade C or 10-315 Min. grade C

**10-805 Machine Learning with Large Datasets**

Spring: 12 units

Large datasets pose difficulties across the machine learning pipeline. They are difficult to visualize and introduce computational, storage, and communication bottlenecks during data preprocessing and model training. Moreover, high capacity models often used in conjunction with large datasets introduce additional computational and storage hurdles during model training and inference. This course is intended to provide a student with the mathematical, algorithmic, and practical knowledge of issues involving learning with large datasets. Among the topics considered are: data cleaning, visualization, and pre-processing at scale; principles of parallel and distributed computing for machine learning; techniques for scalable deep learning; analysis of programs in terms of memory, computation, and (for parallel methods) communication complexity; and methods for low-latency inference. The class will include programming and written assignments to provide hands-on experience applying machine learning at scale. An introductory machine learning course (10-301, 10-315, 10-601, 10-701, or 10-715) is a prerequisite. A strong background in programming will also be necessary; suggested prerequisites include 15-210, 15-214, or equivalent. Students are expected to be familiar with Python or learn it during the course.

Prerequisites: (17-214 or 15-211 or 15-210 or 15-214) and (10-301 or 10-315 or 10-701 or 10-401 or 10-601 or 10-715)

Course Website: <https://10605.github.io/>

**10-806 Foundations of Machine Learning and Data Science**

Fall: 12 units

This course will cover fundamental topics in Machine Learning and Data Science, including powerful algorithms with provable guarantees for making sense of and generalizing from large amounts of data. The course will start by providing a basic arsenal of useful statistical and computational tools, including generalization guarantees, core algorithmic methods, and fundamental analysis models. We will examine questions such as: Under what conditions can we hope to meaningfully generalize from limited data? How can we best combine different kinds of information such as labeled and unlabeled data, leverage multiple related learning tasks, or leverage multiple types of features? What can we prove about methods for summarizing and making sense of massive datasets, especially under limited memory? We will also examine other important constraints and resources in data science including privacy, communication, and taking advantage of limited interaction. In addressing these and related questions we will make connections to statistics, algorithms, linear algebra, complexity theory, information theory, optimization, game theory, and empirical machine learning research. Topics to be covered will include: - Fundamental measures of complexity for generalization, including VC-dimension and Rademacher complexity. - Core algorithmic tools including boosting, regularization, and online optimization with connections to game theory. - Spectral methods, streaming algorithms and other approaches for handling massive data. - Foundations and algorithms for addressing important constraints or externalities such as privacy, limited memory, and communication constraints. - Foundations for modern learning paradigms including semi-supervised learning, never-ending learning, interactive learning, and deep learning.

Course Website: <http://www.cs.cmu.edu/~ninamf/courses/806/10-806-index.html> (<http://www.cs.cmu.edu/~ninamf/courses/806/10-806->)

**10-807 Topics in Deep Learning**

Fall: 12 units

Building intelligent machines that are capable of extracting meaningful representations from high-dimensional data lies at the core of solving many AI related tasks. In the past few years, researchers across many different communities, from applied statistics to engineering, computer science and neuroscience, have developed deep (hierarchical) models and #8212; models that are composed of several layers of nonlinear processing. An important property of these models is that they can learn useful representations by re-using and combining intermediate concepts, allowing these models to be successfully applied in a wide variety of domains, including visual object recognition, information retrieval, natural language processing, and speech perception. This is an advanced graduate course, designed for Master's and Ph.D. level students, and will assume a reasonable degree of mathematical maturity. The goal of this course is to introduce students to the recent and exciting developments of various deep learning methods. Some topics to be covered include: restricted Boltzmann machines (RBMs) and their multi-layer extensions Deep Belief Networks and Deep Boltzmann machines; sparse coding, autoencoders, variational autoencoders, convolutional neural networks, recurrent neural networks, generative adversarial networks, and attention-based models with applications in vision, NLP, and multimodal learning. We will also address mathematical issues, focusing on efficient large-scale optimization methods for inference and learning, as well as training density models with intractable partition functions. Prerequisite: ML: 10-701 or 10-715, and strong programming skills.

Prerequisites: 10-715 Min. grade C or 10-701 Min. grade C

**10-822 Presentation Skills**

Fall and Spring: 6 units

This course provides a forum for students to learn and refine public speaking and technical reading skills. The course will include brief workshops embedded throughout the semester to cover such things as effective structure of presentations and papers, how to give a short talk (think NIPS spotlights), "elevator" talks, structure of a research paper, conference presentations, proposal writing (think thesis and beyond), slide crafting, posters, critical evaluation, and public communications for research. Students will be expected to prepare and present a number of practice talks throughout the semester.

**10-830 Machine Learning in Policy**

Spring: 12 units

Machine learning, a field derived primarily from computer science and statistics, has matured and gained wide adoption over past decades. Alongside exponential increases in data measurement and availability, the ability to develop appropriate and tailored analyses is in demand. As practitioners in the social sciences consider machine learning methods, however, we are identifying limitations and externalities of the applications of machine learning techniques, such as overconfidence in settings with concept drift, lack of generalizability due to selection bias, and magnification of inequities. Machine Learning and Policy seeks to (1) demonstrate motivations and successes of machine learning, to (2) contrast them with more classical methods, and to (3) investigate the promise and cautions of machine learning for public policy. The course will cover variety of topics, including: Basics of machine learning; probability/Bayes/likelihood/conjugacy, terminology, code/algorithm design, evaluation, mathematical formulations Popular and well-performing methods; random forests/trees/ensembles, neural networks/backpropagation/embeddings/generalized adversarial networks, generalized linear models/shrinkage/convexity/basis functions, support vector machines/kernels/optimization/Lagrangian Leveraging other data sources; natural language processing/topic modeling/relational (non-i.i.d.)/relational (Markov logic networks)/temporal data Additional topics: causality/confounding/pr propensity scoring/inverse weighting/causal directed acyclic graphs, fairness/ethics, interpretation/explanation/visualization, anomaly detection, semi-supervised and active learning, reinforcement learning.

Course Website: <https://www.andrew.cmu.edu/user/jweiss2/mlp/>

**10-831 Special Topics in Machine Learning and Policy**

Spring: 6 units

Special Topics in Machine Learning and Policy (90-921/10-831) is intended for Ph.D. students in Heinz College, MLD, and other university departments who wish to engage in detailed exploration of a specific topic at the intersection of machine learning and public policy. Qualified master's students may also enroll with permission of the instructor; all students are expected to have some prior background in machine learning and data mining (10-601, 10-701, 90-866, 90-904/10-830, or a similar course). We will explore state-of-the-art methods for detection of emerging events and other relevant patterns in massive, high-dimensional datasets, and discuss how such methods can be applied usefully for the public good in medicine, public health, law enforcement, security, and other domains. The course will consist of lectures, discussions on current research articles and future directions, and course projects. Specific topics to be covered may include: anomaly detection, change-point detection, time series monitoring, spatial and space-time scan statistics, pattern detection in graph data, submodularity and LTSS properties for efficient pattern detection, combining multiple data sources, scaling up pattern detection to massive datasets, applications to public health, law enforcement, homeland security, and health care. A sample syllabus is available at: <http://www.cs.cmu.edu/~neill/courses/90921-S10.html>

Course Website: <http://www.cs.cmu.edu/~neill/courses/90921-S10.html>

**Robotics Courses****16-161 ROB Freshman Seminar: Artificial Intelligence and Humanity**

Fall and Spring: 9 units

In 1965 British mathematician I.J. Good wrote, An ultraintelligent machine could design even better machines; there would then unquestionably be an intelligence explosion, and the intelligence of man would be left far behind. As we enter an age where companies like Uber are testing driverless cars in Pittsburgh and innovative interfaces like IBMs Watson can play jeopardy and learn techniques for medical diagnoses, how are we to negotiate an intelligence explosion that for many individuals might threaten the very notions of what it means to be human? The future of human-to-machine relationships will likely define our historical epoch and yet, many young technologists and humanists underestimate the downstream impact of technological innovations on human society. Presently, we have little choice but to attend to this rapidly anxiety-ridden question. This seminar will attend to the challenge of present existential questions on what it means to be human (read not machine) in the context of a rapidly advancing technological age. We will consider human narratives throughout history that exam how governments and individual citizens defined humanity in the context of slavery and colonialism as a framework for exploring and projecting what it means to be human in the age of rapidly advancing intelligent machines. We will trace the technological advancements of the recent five decades and identify historical precedents and speculative narratives that help us to consider issues like labor, economic disparity, negotiations of power, human dignity and ethical responsibility within the context of human relations with advancing technological tools that are now coined, artificial intelligence.

**16-211 Foundational Mathematics of Robotics**

Fall: 12 units

This course will cover core mathematics concepts used in many advanced robotics courses at the RI. Perhaps unlike prior courses in math, the focus of this class will be to ground concepts in robotics algorithms or applications. For example: How to move and manipulate objects in 3D space (coordinate transforms, rotations). How to move an articulated robot's end-effector in Cartesian space (Jacobians, gradient optimization). How to have a robot learn to recognize a vision input (neural networks, back propagation). How to plan to navigate a robot optimally (dynamic programming, A\* Search). Prerequisites: 21-122 and 21-241 and (21-325 or 36-225 or 36-218 or 15-259)

**16-223 IDEaTe Portal: Creative Kinetic Systems**

Fall: 10 units

The art and science of machines which evoke human delight through physical movement is founded on a balance of form and computation. This introductory physical computing course addresses the practical design and fabrication of robots, interactive gadgets, and kinetic sculptures. The emphasis is on creating experiences for human audiences through the physical behavior of devices which embody computation with mechanism, sensing, and actuation. Specific topics include basic electronics, elementary mechanical design, embedded programming, and parametric CAD. A key objective is gaining an intuitive understanding of how information and energy move between the physical, electronic, and computational domains to create a compelling behavior. The final projects are tested in the field on children and adults. This interdisciplinary course is an IDEaTe Portal Course open to students from all colleges. For students choosing to follow an IDEaTe program it is an entry into either Physical Computing or Intelligent Environments. The structure of the class revolves around collaborative exercises and projects which introduce core physical computing and system engineering techniques in a human-centric context. Students apply system and design thinking across multiple domains, work together to make and test several devices, and participate in wide-ranging critique which considers both technical and artistic success.

Course Website: <https://courses.ideate.cmu.edu/16-223> (<https://courses.ideate.cmu.edu/16-223/>)

**16-224 IDEaTe: Re-Crafting Computational Thinking with Soft Technologies**

Fall: 6 units

This course focuses on teaching introductory concepts of Robotics, Mechatronics, and Computer Science using an arts-based approach. During the course, students will build their own weaving robot, program it, and learn how weaving art is connected to computer programming and matrix mathematics. Students will also learn the history of weaving, how to design beautiful patterns, and how to extract the features of those patterns into mathematical equations and computer programs.

**16-235 Fantastic Robots and How to Fold Them**

Spring: 9 units

This course will focus on the basics of robotics through a hands-on approach. Students will build their own robots by designing a mechanical structure and embedding actuators, sensors, and controllers. They will then use these robots to solve a simple maze with obstacles. The course content will be delivered through lectures, workshops, and a course-long team project. In classical robotics, we explore the three main behaviors of robots through the work frame of "sense-plan-act". Robots are more than just these behaviors, and students will learn about how to make the physical embodiments of robots through an overview of design and manufacturing techniques for robot mechanisms. Students will be able to make their own mechanisms, improve the system through hardware or software, and learn how to analyze the kinematics and dynamics of these mechanisms to understand and control the motion.

Prerequisites: 15-112 Min. grade C or 15-110 Min. grade C or 15-104 Min. grade C

**16-264 Humanoids**

Spring: 12 units

This course surveys perception, cognition, and movement in humans, humanoid robots, and humanoid graphical characters. Application areas include more human-like robots, video game characters, and interactive movie characters.

Course Website: <http://www.cs.cmu.edu/~cga/humanoids-ugrad/>

**16-299 Introduction to Feedback Control Systems**

Spring: 12 units

This course is designed as a first course in feedback control systems for computer science majors. Course topics include classical linear control theory (differential equations, Laplace transforms, feedback control), linear state-space methods (controllability/observability, pole placement, LQR), nonlinear systems theory, and an introduction to control using computer learning techniques. Priorities will be given to computer science majors with robotics minor.

Prerequisites: 15-122 and 21-122

Course Website: <http://www.cs.cmu.edu/~cga/controls-intro/>

**16-311 Introduction to Robotics**

Spring: 12 units

This course presents an overview of robotics in practice and research with topics including vision, machine learning, motion planning, mobile mechanisms, kinematics, inverse kinematics, and sensors. In course projects, students construct LEGO robots which are driven by a microcontroller, with each project reinforcing the basic principles developed in lectures. Students usually work in teams of three: an electrical engineer, a mechanical engineer, and a computer scientist. Groups are typically self-formed except for the first lab. This course will also expose students to some of the contemporary happenings in robotics, including current robotics research, applications, robot contests and robots in the news. Students registering for this course must register for both Mon/Wed mornings and Tuesday afternoon sections.

Prerequisites: 21-240 Min. grade C or 24-311 Min. grade C or 18-202 Min. grade C or 21-260 Min. grade C or 21-241 Min. grade C

Course Website: <http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/16311/www/current/>

**16-322 Modern Sensors for Intelligent Systems**

Spring: 12 units

The class aims at introducing sensing technologies for robots and other intelligent systems. The course will cover the physical principles of traditional sensors, sensor calibration and evaluation, signal processing algorithms for different sensors, and examples of sensor applications for robots or other intelligent systems. On the sensing system design part, the course will cover the common sensor fusion design and algorithms, and provide examples of sensing systems for different robots or intelligent systems, such as wearable sensors, self-driving cars, autonomous vehicles, assistant robots, and field robots in extreme conditions. The class will contain lectures, two lab sessions, and a course project.

**16-350 Planning Techniques for Robotics**

Spring: 12 units

Planning is one of the core components that enable robots to be autonomous. Robot planning is responsible for deciding in real-time what should the robot do next, how to do it, where should the robot move next and how to move there. This class does an in-depth study of popular planning techniques in robotics and examines their use in ground and aerial robots, humanoids, mobile manipulation platforms and multi-robot systems. The students learn the theory of these methods and also implement them in a series of programming-based projects. To take the class students should have taken an Intro to Robotics class and have a good knowledge of programming and data structures.

Course Website: <http://www.cs.cmu.edu/~maxim/classes/robotplanning/>

**16-362 Mobile Robot Algorithms Laboratory**

Fall: 12 units

This course is an introduction to the theory and algorithms of multirotor vehicle autonomy. Students will work individually to develop a multirotor simulator in Python and C++, add sensors, plan, and perform exploration. Lectures will cover topics to advance the capabilities of the simulator. Homeworks will be designed to increase the autonomy capabilities of the multirotor vehicle. The class will culminate in an individual project that pushes the autonomy capabilities developed throughout the course and may cover multi-robot aerial autonomy, dynamic environment modeling, or advanced planning and control. In order to succeed in the course, students must have a 2nd year science/engineering level background in mathematics (matrices, vectors, coordinate systems) and have already mastered at least one object-oriented programming language like C++ or Python. When the course is over, students will have written a single software system that has been incrementally extended in functionality and regularly debugged throughout the semester.

**16-371 Personalized Responsive Environments**

Spring: 9 units

[IDeATe collaborative course]. Environmental factors have a significant impact on mood and productivity. Creating responsive environments necessitates the design of surroundings that are able to metamorphose in order to optimize user strengths and available resources and evolve in stride with user needs. This course will investigate the development of spaces that adapt to user preferences, moods, and task specific demands. Both the design and engineering of such personalized environments will be explored. Central course concepts will include, understanding the user, integrating various modalities (e.g., light, heat, sound) to support the changing needs of task and user, and the creation of adaptive environments that learn user preferences over time. Please note that there may be usage/materials fees associated with this course.

Prerequisites: 60-223 Min. grade C or 18-090 Min. grade C or 15-104 Min. grade C or 62-150 Min. grade C

**16-374 IDeATe: Art of Robotic Special Effects**

Spring: 12 units

Inspired by the early "trick" films of George Melies, this project-oriented course brings together robotics and film production technique to infuse cinema with the wonder of live magic. Students will learn the basics of film production using animatronics, camera motion control, and compositing. The projects apply these techniques to create innovative physical effects for short films, all the way from concept to post-production. The course emphasizes real-time practical effects to explore the immediacy and interactivity of improvisation and rehearsal. The robotics topics include animatronic rapid prototyping and programming human-robot collaborative performance. The course includes a brief overview of the history of special effects and robotics to set the work in context.

Course Website: <https://courses.ideate.cmu.edu/16-374> (<https://courses.ideate.cmu.edu/16-374/>)

**16-375 IDeATe: Robotics for Creative Practice**

Fall: 10 units

Robots come in all shapes and sizes: it is the integration of software and hardware that can make any machine surprisingly animate. This project-oriented course brings art and engineering together to build performance systems using embodied behavior as a creative medium. Students learn skills for designing, constructing and programming automated systems for storytelling and human interaction, then explore the results through exhibition and performance. Technical topics include closed-loop motion control, expressive physical and computational behavior, machine choreography, and performance conceptualization. Discussion topics include both contemporary kinetic sculpture and robotics research. This interdisciplinary course is part of IDeATe Physical Computing but is open to any student.

Prerequisites: 60-212 or 99-361 or 15-112 or 15-110 or 60-210 or 15-104

Course Website: <https://courses.ideate.cmu.edu/16-375> (<https://courses.ideate.cmu.edu/16-375/>)

**16-376 IDeATe: Kinetic Fabrics**

Spring: 10 units

Kinetic Fabrics brings together the fields of robotics and textiles to explore their unified creative and expressive potential. It is a wide-open frontier for kinetic art, wearable art, and architectural installation. In this course students will build a variety of performative systems combining fabrics and robotic technologies. Students will apply modular actuation and sensing to textile artworks, using software designed to facilitate fluid explorations, rapid iterations, and playful experimentation. Students will learn basic textile skills, such as hand and machine sewing, as well as gain facility and familiarity with the characteristics of multiple type of fabrics. Historical precedents as well as contemporary examples of works will support students creative growth and knowledge of the field. Students' course work will include short-term and long-term projects, sampling and prototyping, critique, and documentation. Additionally, students will organize an end-of-semester event where they will perform a developed kinetic fabric work for a public audience.

Course Website: <https://courses.ideate.cmu.edu/16-376> (<https://courses.ideate.cmu.edu/16-376/>)

**16-384 Robot Kinematics and Dynamics**

Fall: 12 units

Foundations and principles of robotic kinematics. Topics include transformations, forward kinematics, inverse kinematics, differential kinematics (Jacobians), manipulability, and basic equations of motion. Course also include programming on robot arms.

Prerequisites: 24-311 or 21-241 or 18-202 or 15-122 Min. grade C or 16-311

**16-385 Computer Vision**

Fall and Spring: 12 units

This course provides a comprehensive introduction to computer vision. Major topics include image processing, detection and recognition, geometry-based and physics-based vision, sensing and perception, and video analysis. Students will learn basic concepts of computer vision as well as hands on experience to solve real-life vision problems. This course is for undergraduate students only.

Prerequisites: (18-202 Min. grade C and 15-122 Min. grade C) or (21-241 Min. grade C and 21-259 Min. grade C and 15-122 Min. grade C) or (21-241 Min. grade C and 21-256 Min. grade C and 15-122 Min. grade C) or (21-241 Min. grade C and 24-282 Min. grade C and 15-122 Min. grade C) or (21-241 Min. grade C and 21-254 Min. grade C and 15-122 Min. grade C)

Course Website: <http://www.cs.cmu.edu/~16385/>

**16-397 Art, Conflict and Technology**

Spring: 12 units

This course considers the period of violence in Northern Ireland from 1968 to 1998 known as The Troubles, and recent issues pertaining to sovereignty and borders caused by Brexit, Britain's proposed exit from the European Union, as a point of comparison between societies rife with strife, division and predilections to violence. We investigate the ways in which visual art to literature to theatrical performance explores and interrogates societal conflict and emergence from conflict, and how evolving technological systems influence political power dynamics and modes of artistic practice. We will use the legacy of societal conflict in Ireland and Northern Ireland to compare concepts and physical manifestations of borders, barriers and bridges in the region and in global contexts. We will examine fluctuating development of democratic processes in Ireland and Northern Ireland, individual and group public performance, and the influence of technologically crude and highly sophisticated tools on communities emerging from strife. We will use our analytical lens to focus on figurative and literal borders, barriers and bridges to explore work produced in Belfast, Derry and Dublin, alongside circumstances and artistic practice in present-day Pittsburgh, Ciudad Juarez, Jerusalem and Soweto. On a visit to Ireland and Northern Ireland over spring break, students will meet with artists, writers, legislators, community organizers, academics and ex-combatants, to learn about their past experience and current motivations. Students will analyze artistic practice, peacekeeping initiatives and performance of identity in relation to the historical framework from which it emerges in Ireland and Northern Ireland. We will use this foundation as a point of comparison to practices throughout the world. Students will process their experience and developing analytical skills by documenting their responses through original creative work.

**16-421 Vision Sensors**

Spring: 12 units

This course covers the fundamentals of vision cameras and other sensors - how they function, how they are built, and how to use them effectively. The course presents a journey through the fascinating five hundred year history of "camera-making" from the early 1500's "camera obscura" through the advent of film and lenses, to today's mirror-based and solid state devices (CCD, CMOS). The course includes a significant hands-on component where students learn how to use the sensors and understand, model and deal with the uncertainty (noise) in their measurements. While the first half of the course deals with conventional "single viewpoint" or "perspective" cameras, the second half of the course covers much more recent "multi-viewpoint" or "multi-perspective" cameras that includes a host of lenses and mirrors.

Prerequisites: 21-241 and 21-111

Course Website: <http://www.cs.cmu.edu/~ILIM/courses/vision-sensors/>

**16-423 Designing Computer Vision Apps**

Fall: 12 units

Computer vision is a discipline that attempts to extract information from images and videos. Nearly every smart device on the planet has a camera, and people are increasingly interested in how to develop apps that use computer vision to perform an ever expanding list of things including: 3D mapping, photo/image search, people/object tracking, augmented reality etc. This course is intended for students who are not familiar with computer vision, but want to come up to speed rapidly with the latest in environments, software tools and best practices for developing computer vision apps. No prior knowledge of computer vision or machine learning is required although a strong programming background is a must (at a minimum good knowledge of C/C++). Topics will include using conventional computer vision software tools (OpenCV, MATLAB toolboxes, VLFeat, CAFFE), and development on iOS devices using mobile vision libraries such as GPUImage and fast math libraries like Armadillo and Eigen. For consistency, all app development will be in iOS and it is expected that all students participating in the class have access to an Intel-based MAC running OS X Mavericks or later. Although the coursework will be focused on a single operating system, the knowledge gained from this class is intended to generalize to other mobile platforms such as Android etc. Prerequisites: (15-213 and 21-240) or (15-213 and 21-241) or (18-213 and 18-202)

Course Website: <http://16423.courses.cs.cmu.edu>**16-425 Medical Image Analysis**

Spring: 12 units

Students will gain theoretical and practical skills in 2D, 3D, and 4D biomedical image analysis, including skills relevant to general image analysis. The fundamentals of computational medical image analysis will be explored, leading to current research in applying geometry and statistics to segmentation, registration, visualization, and image understanding. Additional and related covered topics include de-noising/restoration, morphology, level sets, and shape/feature analysis. Students will develop practical experience through projects using the latest version of the National Library of Medicine Insight Toolkit (ITK) and SimpleITK, a popular open-source software library developed by a consortium of institutions including Carnegie Mellon University and the University of Pittsburgh. In addition to image analysis, the course will include interaction with radiologists and pathologist(s). \*\*\* Lectures are at CMU and students will visit clinicians at UPMC. Some or all of the class lectures may also be videoed for public distribution, but students may request to be excluded from distributed video. 16-725 is a graduate class, and 16-425 is a cross-listed undergraduate section. 16-425 is new this year, and has substantially reduced requirements for the final project and for the larger homework assignments, nor does it require shadowing the clinicians. Prerequisites: Knowledge of vector calculus, basic probability, and either C++ or python, including basic command-line familiarity and how to pass arguments to your own command-line programs. Extensive expertise with C++ and templates is not necessary, but some students may find it helpful.

Course Website: [http://www.cs.cmu.edu/~galeotti/methods\\_course/](http://www.cs.cmu.edu/~galeotti/methods_course/)**16-441 Advanced CP/SIS: Urban Intervention**

Fall and Spring: 12 units

This course introduces students to theories, practices, and communities for critical investigation of urban spaces and play within them. The course unfolds along two parallel trajectories: research (literature review, lectures, readings, demonstrations) and design (three iterated individualized projects and a fourth larger scale final project). The first half of the course will introduce students to a wide range of theories and techniques within urban intervention that draw from fluxus, the situationist international, activism and hacktivism, as well as public policy, philosophy, psychology and economics. Students will study theoretical and practical frameworks for artistic intervention into public urban spaces, while concurrently researching actual sites and communities within Pittsburgh for experimentation. Students are required to conceptualized projects on larger (urban) scales, and find ways to implement their projects safely and legally by pursuing the necessary administrative, social, technical, financial steps required to create meaningful interventions in public spaces. This class will specifically explore three media for urban intervention: Sound Outdoor video projection Robotics, Autonomy and Mobility in the way of remote control vehicles (e.g. cars, quad-copters, etc.). For each theme, students are required to produce one project that is iterated twice or more. The undergraduate (60441) and graduate (60741) sections of the course meet concurrently and follow the same syllabus and assignments. In addition to the coursework documented in the syllabus, Graduate level students are expected to write a research paper suitable for submission to a notable relevant academic conference. This process includes a rough draft, revisions and a completed and formatted paper ready for submission

**16-450 Robotics Systems Engineering**

Fall: 12 units

Systems engineering examines methods of specifying, designing, analyzing and testing complex systems. In this course, principles and processes of systems engineering are introduced and applied to the development of robotic devices. The focus is on robotic system engineered to perform complex behavior. Such systems embed computing elements, integrate sensors and actuators, operate in a reliable and robust fashion, and demand rigorous engineering from conception through production. The course is organized as a progression through the systems engineering process of conceptualization, specification, design, and prototyping with consideration of verification and validation. Students completing this course will engineer a robotic system through its compete design and initial prototype. The project concept and teams can continue into the Spring-semester (16-474 Robotics Capstone) for system refinement, testing and demonstration. Prerequisites: 16-311 Min. grade B and (18-370 Min. grade B or 16-299 Min. grade B or 24-451 Min. grade B)

**16-455 IDeATe: Human-Machine Virtuosity**

Spring: 12 units

[IDeATe course] Human dexterous skill embodies a wealth of physical understanding which complements computer-based design and machine fabrication. This project-oriented course explores the duality between hand and machine through the practical development of innovative design and fabrication systems. These systems fluidly combine the expressivity and intuition of physical tools with the scalability and precision of the digital realm. Students will develop novel hybrid design and production workflows combining analog and digital processes to support the design and fabrication of their chosen projects. Specific skills covered include 3D modeling (CAD), 3D scanning, algorithmic geometric modeling, digital and robotic fabrication (additive and subtractive manufacturing), motion capture and computer based sensing, and human-robot interaction design. Areas of interest include architecture, art, and product design.

Course Website: <https://courses.ideate.cmu.edu/16-455> (<https://courses.ideate.cmu.edu/16-455/>)**16-456 Reality Computing Studio**

Fall: 12 units

[IDeATe collaborative course] Reality computing encompasses a constellation of technologies focused around capturing reality (laser scanning, photogrammetry), working with spatial data (CAD, physical modeling, simulation), and using data to interact with and influence the physical world (augmented / virtual reality, projector systems, 3d printing, robotics). Taught in collaboration with the school of architecture, this studio asks students to apply these technologies to real world problems such as residential design, sustainability, and infrastructure monitoring.

Course Website: <http://ideate.cmu.edu/about-ideate/departments/robotics-institute/reality-computing/>**16-457 Reality Computing II**

Spring: 12 units

[IDeATe collaborative course] Reality computing encompasses a constellation of technologies focused around capturing reality (laser scanning, photogrammetry), working with spatial data (CAD, physical modeling, simulation), and using data to interact with and influence the physical world (augmented / virtual reality, projector systems, 3d printing, robotics). This iteration of the reality computing course will focus on "design realization": the translation from digital design to fully realized tangible artifact. Collaborating with the UDBS design studio, and within the context of a full-scale residential prototype, students will investigate how reality computing technologies can be used to accelerate and advance the process of design realization by using reality computing to understand existing homes, map design data into the real world, and highlight conflicts between design and reality. Topics of special focus within the course are residential design (John Folan) and augmented reality and robotics (Pyry Matikainen).

Course Website: <http://ideate.cmu.edu/about-ideate/departments/robotics-institute/reality-computing/>

**16-461 Experimental Capture**

Fall: 9 units

Performance capture is used in applications as varied as special effects in movies, animation, sports training, physical rehabilitation, and human-robot/human-computer interaction. This course will survey state-of-the-art techniques and emerging ideas, in the industry and in academia, to capture, model, and render human performances. The course will be a mix between lectures and discussion of recent progress in human motion capture and analysis. The course evaluation will be project-based, in which students will capture their own body and face motion, and build projects around the data they collect individually and as a group. We will cover: 1. Capture Techniques: We will describe and use various systems including motion capture, video-based capture, depth sensors, scanners, and eye-gaze trackers; 2. Modeling and Representation: We will cover classic and contemporary representations of face and body pose and motion, including statistical and physics-based techniques; 3. Rendering Applications: As new rendering paradigms emerge, new applications continue to develop. We will study recent progress in animation, synthesis, classification, and rehabilitation on new forms of displays. Please note that there may be usage/materials fees associated with this course.

Prerequisites: 60-422 or 15-365

**16-465 Game Engine Programming**

Spring: 10 units

This course is designed to help students understand, modify, and develop game engines. Game engines consist of reusable runtime and asset pipeline code. They provide game-relevant abstractions of low-level system services and libraries, making it easier to write bug-free games that work across multiple platforms. Game engines also handle artistic content, providing or integrating with authoring tools to ease the process of creating high-fidelity games. In this course, we will discuss the problems game engines attempt to solve, examine how current state-of-the-art engines address these problems, and create our own engines based on what we learn. We will cover both the content authoring and runtime aspects of engines. Coursework will consist of frequent, tightly-scoped programming and system design assignments; expeditions through game engine source code; and two group projects and #8212; one in an engine created from scratch, and one that requires modification of an existing engine. Prerequisites: Students will be expected to be fluent in at least one programming language. We will be working with C++, Javascript, and a smattering of Python. We will be using git for version control and code sharing. The assignments in the course will be designed to be completed on an OSX or Linux workstation (e.g. the IDeATe "virtual cluster"). Working with Windows will be possible, but might require extra effort. We will be building a 3D model pipeline around Blender, but no prior knowledge of the tool will be assumed.

Prerequisites: 62-150 Min. grade C or 15-213 Min. grade C or 15-112 Min. grade C or 15-104 Min. grade C

**16-467 Human Robot Interaction**

Spring: 12 units

The field of human-robot interaction (HRI) is fast becoming a significant area of research in robotics. The basic objective is to create natural and effective interactions between people and robots. HRI is highly interdisciplinary, bringing together methodologies and techniques from robotics, artificial intelligence, human-computer interaction, psychology, education, and other fields. This course is primarily lecture-based, with in-class participatory mini-projects, homework assignments, a group term project that will enable students to put theory to practice, and a final. The topics covered will include technologies that enable human-robot interactions, the psychology of interaction between people and robots, how to design and conduct HRI studies, and real-world applications such as assistive robots. This course has no prerequisites, but some basic familiarity with robots is recommended (programming knowledge is not necessary, but is useful for the term project).

Course Website: <http://harp.ri.cmu.edu/courses> (<http://harp.ri.cmu.edu/courses/>)**16-469 Innovation and Shared Prosperity: Community-engagement for change**

Fall: 12 units

How might we, as a community of learners, utilize our collective talents for innovation and shift our society towards greater justice? In this course we will cover the historical and social context of university-community engagement, discuss best practices in engagement efforts, and operationalize emergent strategy alongside design justice principles. Learnings from this course will foster the growth of lifelong dispositions and habits that can empower learners to chart a course for their personal careers that are consonant with community empowerment and societal equity. This class is for individuals interested in pursuing a career at the intersection of technology and societal equity or for individuals who are interested in issues of justice and equity more broadly. Learning methods for this course will include readings, reflections, and in-class discussions. Students in this class will be asked to draw on their own experiences and to explore case studies. We also anticipate that students will directly engage with local community as facilitated through existing connections with the Center for Shared Prosperity.

**16-474 Robotics Capstone**

Spring: 12 units

In this course students refine the design, build, integrate, test, and demonstrate the performance of the robot they designed in the pre-requisite Systems Engineering Course (16-450). The students are expected to continue to apply the process and methods of Systems Engineering to track requirements, evaluate alternatives, refine the cyberphysical architectures, plan and devise tests, verify the design, and validate system performance. In addition, the students learn and apply Project Management techniques to manage the technical scope, schedule, budget, and risks of their project. The course consists of lectures, class meetings, reviews, and a final demonstration. Lectures cover core topics in Project Management and special topics in Systems Engineering. During class meetings the students and instructor review progress on the project and discuss technical and project-execution challenges. There are three major reviews approximately at the end of each of the first three months of the semester. For each review, the students give a presentation and submit an updated version of the System Design and Development Document. The course culminates in a System Performance Validation Demonstration at the end of the semester. In addition to that the students hold a special demonstration of their robotic system for the broader Robotics community.

Prerequisite: 16-450 Min. grade C

**16-480 IDeATe: Special Topics: Creative Soft Robotics**

Spring: 10 units

This experimental course offers unique topics situated at the intersection of robotics research and the arts, with a specific research focus that varies each semester. In this course, students survey the state of an emerging research area, then design and fabricate experimental systems and artworks on the theme. Students are guided through literature search and technical paper analysis to identify opportunities and techniques. The textual study spans contemporary robotics and arts literature. The project component is research-focused and explores novel techniques in design, fabrication, programming, and control. The project sequence culminates in the collaborative design of expressive robotic systems which match technical innovation with a human need or artistic expression. The initial iteration of the course focuses on soft robotics, an emerging discipline centered on devices constructed from compliant materials that incorporate sensing and actuation. The literature survey spans soft robotics and kinetic sculpture. The projects center on fabricating forms that incorporate actuators and sensors using silicone rubber cast into 3D-printed and laser-cut molds. This course is offered by IDeATe and this iteration will satisfy minor requirements for IDeATe Soft Technologies or IDeATe Physical Computing.

Course Website: <https://courses.ideate.cmu.edu/16-480> (<https://courses.ideate.cmu.edu/16-480/>)**16-595 Undergraduate Independent Study**

All Semesters

For students to pursue an independent study with a Robotics Institute faculty member.

**16-597 Undergraduate Reading and Research**

All Semesters

Undergraduate Reading and Research enables students to gain academic credits for conducting independent studies in robotics. Students must work with a robotics faculty advisor to devise a specific objective, activities (such as reading, evaluating, designing, coding, building, or testing robotic systems) and metrics for evaluation of their performance by their advisor.

**16-621 MSCV Project I**

Spring: 12 units

The MSCV capstone project course is designed to give project teams additional feedback on their capstone project from peers and faculty. Every week, capstone teams will present their project PPFs (Past-Present-Future) reports. For the presenting teams, the capstone course will help develop presentation and communication skills. For the students participating as peer-reviewers, it will help develop critical thinking and the ability to give constructive advice.

Course Website: <https://piazza.com/cmu/spring2019/16621> (<https://piazza.com/cmu/spring2019/16621/>)

**16-622 MSCV Capstone**

Fall: 12 units

The MSCV capstone project course is designed to give project teams additional feedback on their capstone project from peers and faculty. Every week, capstone teams will present their project PPFs (Past-Present-Future) reports. For the presenting teams, the capstone course will help develop presentation and communication skills. For the students participating as peer-reviewers, it will help develop critical thinking and the ability to give constructive advice.

**16-623 Advanced Computer Vision Apps.**

Fall: 12 units

Computer vision is a discipline that attempts to extract information from images and videos. Nearly every smart device on the planet has a camera, and people are increasingly interested in how to develop apps that use computer vision to perform an ever expanding list of things including: 3D mapping, photo/image search, people/object tracking, augmented reality etc. This course is intended for graduate students who are familiar with computer vision, and are keen to learn more about the applying state of the art vision methods on smart devices and embedded systems. A strong programming background is a must (at a minimum good knowledge of C/C++), topics will include using conventional computer vision software tools (OpenCV, MATLAB toolboxes, VLFeat, CAFFE, Torch 7), and development on iOS devices using mobile vision libraries such as GPUImage, Metal and fast math libraries like Armadillo and Eigen. For consistency, all app development will be in iOS and it is expected that all students participating in the class have access to an Intel-based MAC running OS X Mavericks or later. Although the coursework will be focused on a single operating system, the knowledge gained from this class will easily generalize to other mobile platforms such as Android etc.

Prerequisites: 16-720 or 16-385

Course Website: <http://16623.courses.cs.cmu.edu>

**16-627 MSCV Seminar**

Fall

(Only open to MSCV students.) MSCV students will be required to participate in this one-semester seminar course which will prepare them for the MSCV project starting in the Spring semester. The first part of this course will cover talks by computer vision and related faculty about the ongoing research, development programs related to Computer Vision at CMU. The second part of this course will include student/faculty tutorial on topics such as OpenCV, Dataset Creation, Mechanical Turk etc. The goal of this series is to get students acquainted with practical knowledge for a successful project. In the last month of the course, each lecture will cover upto four possible MSCV projects pitched by faculty or industrial sponsors. At the end of the course students will turn in their choices, and a faculty committee will assign them the final projects.

**16-663 F1Tenth Autonomous Racing**

Fall: 12 units

This hands-on, lab-centered course is for senior undergraduates and graduate students interested in the fields of artificial perception, motion planning, control theory, and applied machine learning. It is also for students interested in the burgeoning field of autonomous driving. This course introduces the students to the hardware, software and algorithms involved in building and racing an autonomous race car. Every week, students take two lectures and complete an extensive hands-on lab. By Week 6, the students will have built, programmed and driven a 1/10th scale autonomous race car. By Week 10, the students will have learned fundamental principles in perception, planning and control and will race using map-based approaches. In the last 6 weeks, they develop and implement advanced racing strategies, computer vision and machine learning algorithms that will give their team the edge in the race that concludes the course.

**16-664 Self-Driving Cars: Perception & Control**

Fall: 12 units

This course will teach the theoretical underpinnings of self-driving car algorithms and the practical application of the material in hands-on labs. Topics will include deep learning, computer vision, sensor fusion, localization, trajectory optimization, obstacle avoidance, and vehicle dynamics.

**16-665 Robot Mobility on Air, Land, & Sea**

Fall: 12 units

Required core course for MRSD first-year students. Many robots are designed to move through their environments. Three prevalent environments on earth are land, air, and water. This course will explore the modeling, control, and navigation of ground-based (wheeled and legged), air-based (rotorcraft such as quadcopters), and water-based robots.

**16-675 Manufacturing Futures**

Spring: 12 units

The course will introduce an array of technologies that will contribute to the future of making things and will be organized into 4 logical modules that will culminate in a team-based design project. Module 1 (Manufacturing Visions and Design Methodology): David Bourne. Module 2 (Manufacturing Processes and Process Tradeoffs): Brandon Bodily. Module 3 (Electronic Manufacturing): Rahul Panat. Module 4 (Workforce Development) : David Bourne.

**16-682 Robotic Systems Development Project Course II**

Fall: 15 units

Required core course for MRSD second-year students. This course is the second semester in a two-semester sequence intended to enable student teams to design and implement robot systems from the requirements development phase through implementation, verification, and demonstration of a working prototype. Teams of 4-5 students continue work on a project provided by industrial and academic partners, refine design requirements, refine or create new subsystems, and integrate and demonstrate the full system.

**16-714 Special Topics: Physics-based Simulation in Robotics**

Fall: 12 units

special topics physics-based simulation in robotics - description TBD

**16-715 Advanced Robot Dynamics and Simulation**

Fall: 12 units

This course explores the fundamental mathematics behind modeling the physics of robots, as well as state-of-the-art algorithms for robot simulation. We will review classical topics like Lagrangian mechanics and Hamilton's Principle of Least Action, as well as modern computational methods like discrete mechanics and fast linear-time algorithms for dynamics simulation. A particular focus of the course will be rigorous treatments of 3D rotations and non-smooth contact interactions (impacts and friction) that are so prevalent in robotics applications. We will use numerous case studies to explore these topics, including quadrotors, fixed-wing aircraft, wheeled vehicles, quadrupeds, humanoids, and manipulators. Homework assignments will focus on practical implementation of algorithms and a course project will encourage students to apply simulation methods to their own research.

**16-720 Computer Vision**

Fall and Spring: 12 units

Section A is a required core course for MRSD first-year students, and Section B is a required core course for MSCV students. This course introduces the fundamental techniques used in computer vision, that is, the analysis of patterns in visual images to reconstruct and understand the objects and scenes that generated them. Topics covered include image formation and representation, camera geometry, and calibration, computational imaging, multi-view geometry, stereo, 3D reconstruction from images, motion analysis, physics-based vision, image segmentation and object recognition. The material is based on graduate-level texts augmented with research papers, as appropriate. Evaluation is based on homeworks and a final project. The homeworks involve considerable Python programming exercises. Texts recommended but not required: Title: "Computer Vision Algorithms and Applications" Author: Richard Szeliski Series: Texts in Computer Science Publisher: Springer ISBN: 978-1-84882-934-3 Title: "Computer Vision: A Modern Approach" Authors: David Forsyth and Jean Ponce Publisher: Prentice Hall ISBN: 0-13-085198-1

Course Website: <http://www.andrew.cmu.edu/course/16-720/>

**16-725 (Bio)Medical Image Analysis**

Spring: 12 units

Students will gain theoretical and practical skills in 2D, 3D, and 4D biomedical image analysis, including skills relevant to general image analysis. The fundamentals of computational medical image analysis will be explored, leading to current research in applying geometry and statistics to segmentation, registration, visualization, and image understanding. Additional and related covered topics include de-noising/restoration, morphology, level sets, and shape/feature analysis. Students will develop practical experience through projects using the latest version of the National Library of Medicine Insight Toolkit (ITK) and SimpleITK, a popular open-source software library developed by a consortium of institutions including Carnegie Mellon University and the University of Pittsburgh. In addition to image analysis, the course will include interaction with radiologists and pathologists(s). \*\*\* Lectures are at CMU and students will visit clinicians at UPMC. Some or all of the class lectures may also be videoed for public distribution, but students may request to be excluded from distributed video. 16-725 is a graduate class, and 16-425 is a cross-listed undergraduate section. 16-425 is new this year, and has substantially reduced requirements for the final project and for the larger homework assignments, nor does it require shadowing the clinicians. Prerequisites: Knowledge of vector calculus, basic probability, and either C++ or python, including basic command-line familiarity and how to pass arguments to your own command-line programs. Extensive expertise with C++ and templates is not necessary, but some students may find it helpful.

Course Website: [http://www.cs.cmu.edu/~galeotti/methods\\_course/](http://www.cs.cmu.edu/~galeotti/methods_course/)**16-726 Learning-based Image Synthesis**

Spring: 12 units

This course introduces machine learning methods for image and video synthesis. The objectives of synthesis research vary from modeling statistical distributions of visual data, through realistic picture-perfect recreations of the world in graphics, and all the way to providing interactive tools for artistic expression. Key machine learning algorithms will be presented, ranging from classical learning methods (e.g., nearest neighbor, PCA) to deep learning models (e.g., ConvNets, NeRF, deep generative models, including GANs, VAEs, autoregressive models, and diffusion models). Finally, we will discuss image and video forensics methods for detecting synthetic content. In this class, students will learn to build practical applications and create new visual effects using their own photos and videos.

**16-730 Robotics Business**

Spring: 12 units

This course introduces and develops business concepts that will be useful to new and existing companies, while focusing on robotic technology exemplars. The concepts begin with how to identify a new idea to for a business that can be effectively started. Initial ideas often start as a grandiose plan to change the world and these plans are legitimately the fuel that drive new businesses forward. However, when a company starts (e.g., builds a prototype or writes a first line of code), what is the least product a company can produce that customers still want and need? This kernel and #8212; extracted from the "big plan" and #8212; is a Minimal Viable Product (MVP). Once an MVP business kernel is formulated, we will learn and study how to understand customer needs, how to market a new idea and how raise and manage money for a new business entity. These steps abridge information that can be found in an MBA curriculum, but engineers and scientists focused on the technical side will need this information to participate in the process of building companies. In parallel, we will investigate the marketplace through the stock market. The stock market is a powerful window into the world of business. In other words, when a new business is built it has to live inside the competitive environment of every other business. To understand this eco-system, we will follow several companies in-situ as they go through their own ups-and-downs within the business world. The course is project based. Each student will either build their own business concept, or they will build an improvement plan that would be targeted to improve an existing business. Professor Bourne is a founding member of the Robotics Institute(1979) and has taught business concepts within the Tepper Business School and the Robotics Institute since 1988. In addition, he is the President of his own company Design One Software.

**16-735 Ethics and Robotics**

Intermittent: 12 units

This course contextualizes robotics, AI, and machine learning within cultural conversation, ethics, and power relationships in society. It will draw upon "AI and Humanity" as well as numerous other texts, including Mindless by Simon Head, Drone Theory by Gr and #233;goire Chamayou, and news articles. The course will culminate in team-based design and futuring project addressing the ways in which robotic technologies will influence society and values in the near future. Our target audience is students who will participate in computer science and robotics research and can use this course to inform future research and career decisions.

Course Website: <https://vdean.github.io/16-735-ethics-robotics.html>**16-740 AI for Manipulation**

Spring: 12 units

Manipulation is the process of changing the state of objects through direct physical interactions. To perform manipulation tasks in unstructured environments, autonomous robots will need to learn about the objects in their surroundings as well as the skills required to manipulate and change the state of these objects. In this course, we explore the use of machine learning and data-driven algorithms for robot manipulation. The course introduces students to the wide variety of challenges posed by manipulation tasks, and how these challenges can be formulated as learning problems. Students are taught how these problems can be solved using machine learning techniques. The types of machine learning methods covered in this course include supervised, unsupervised, active, and reinforcement learning methods. The course includes both lectures and guided paper discussions.

**16-741 Mechanics of Manipulation**

Fall: 12 units

Mechanics of Manipulation is a graduate level course that dives into the fundamentals of robotic manipulation. Through this course you will learn the kinematics, statics, and dynamics of robotic manipulators as they interact with the world to accomplish tasks. You will gain experience with the intelligent use of kinematic constraint, gravity, and frictional forces. Additional topics include rigid body mechanics, automatic planning based on mechanics, deformable manipulation, and simulation of dynamic manipulation. Applications of robotic manipulation are drawn from physical human-robot interaction, manufacturing, and other domains.

Course Website: <http://www.cs.cmu.edu/afs/cs/academic/class/16741-s07/www/index.html> (<http://www.cs.cmu.edu/afs/cs/academic/class/16741-s07/www/>)**16-742 Geometry of Locomotion**

Fall: 12 units

This course introduces geometric methods for the analysis of locomoting systems. Focusing on the kinematics of locomoting systems, the course covers topics from differential geometry, geometric mechanics, and motion planning. Specific topics include configuration spaces, manifolds, groups, Lie groups, representations of velocity, holonomic and nonholonomic constraints, constraint curvature, response to cyclic inputs and distance metrics. The primary goal of this class is to develop an intuitive understanding of these concepts and how they are used in locomoting systems, rather than working through a set of formal proofs and derivations. We do, however, incorporate enough mathematical formalism for this class to serve as a starting point for further investigation into this topic area. We also call upon biological data, when available, and relate to the mathematical formalisms in the class.

**16-745 Optimal Control and Reinforcement Learning**

Spring: 12 units

This is a course about how to make robots move through and interact with their environment with speed, efficiency, and robustness. We will survey a broad range of topics from nonlinear dynamics, linear systems theory, classical optimal control, numerical optimization, state estimation, system identification, and reinforcement learning. The goal is to provide students with hands-on experience applying each of these ideas to a variety of robotic systems so that they can use them in their own research.

Course Website: <http://www.cs.cmu.edu/~cga/dynopt/>



**16-748 Underactuated Robots**

Fall: 12 units

People and animals move through and interact with the world in a fundamentally dynamic way. In the vast majority of cases the same cannot be said for robots. In fact, many conventional approaches to motion planning and robot control attempt to explicitly cancel out the dynamics associated with different tasks. This class will consider underactuated robots, systems that do not have full control over their state and therefore cannot be planned for or controlled via conventional methods. Our goal will be to make novel locomoting robots act more "naturally." This class will highlight the relationship between conventional ideas from deterministic motion planning and control design (e.g., dynamic programming and linear-quadratic regulators) and their contemporary counterparts, many of which help form the analytical basis for the probabilistic reasoning that underlies contemporary AI systems (e.g., POMDPs). Note that this course is inspired by and, for the most part, will follow the format of "Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines" created by Prof. Russ Tedrake at MIT. We will take several tangents, but the course materials provided by Prof. Tedrake through MIT Open Courseware are an incredible resource for this course (and really just in general).

**16-761 Mobile Robots**

Spring: 12 units

The course is targeted to senior undergraduates and graduate level students. The lectures will develop the fundamentals of this emerging sub-field of robotics by calling on the experience of practitioners, the common themes of the literature, and relevant material from more basic fields such as computer vision, mathematics, and physics.

Course Website: <http://www.frc.ri.cmu.edu/~alonzo/teaching/16-761/16-761.html>

**16-765 Robotics & AI for Agriculture**

Spring: 12 units

Robotics and artificial intelligence technologies have the potential to increase the efficiency, long-term sustainability, and profitability of agricultural production methods. This class will introduce common aspects of agricultural systems, the AI/Robotics tools that are being used to address them, and key research challenges looking forward. Technical topics include IoT sensor networks, in-field computer vision, 3D crop mapping and modeling, mobile robot navigation, and robotic manipulation of plants. Course sessions will be split evenly between lectures by the instructor and student-led discussion of relevant papers from the contemporary research literature.

**16-778 Mechatronic Design**

Spring: 12 units

Mechatronics is the synergistic integration of mechanism, electronics, and computer control to achieve a functional system. This course is a semester-long multidisciplinary capstone hardware project design experience in which small (typically four-person) teams of electrical and computer engineering, mechanical engineering and robotics students deliver an end-of-course demonstration of a final integrated system capable of performing a mechatronic task. Throughout the semester, the students design, configure, implement, test and evaluate in the laboratory devices and subsystems culminating in the final integrated mechatronic system. Lectures will complement the laboratory experience with comparative surveys, operational principles, and integrated design issues associated with the spectrum of mechanism, microcontroller, electronic, sensor, and control components.

Course Website: <http://www.ece.cmu.edu/courses/items/18578.html>

**16-782 Planning and Decision-making in Robotics**

Fall: 12 units

Planning and Decision-making are critical components of autonomy in robotic systems. These components are responsible for making decisions that range from path planning and motion planning to coverage and task planning to taking actions that help robots understand the world around them better. This course studies underlying algorithmic techniques used for planning and decision-making in robotics and examines case studies in ground and aerial robots, humanoids, mobile manipulation platforms and multi-robot systems. The students will learn the algorithms and implement them in a series of programming-based projects.

**16-785 Integrated Intelligence in Robotics: Vision Language Planning**

Intermittent: 12 units

This is a project-oriented course that covers interdisciplinary topics on cognitive intelligence in robotic systems. Cognitive abilities constitute high-level, humanlike intelligence that exhibits reasoning or problem-solving skills. Such abilities as semantic perception, use of language, and task planning can be built on top of low-level robot autonomy. The topics covered generally bridge across multiple technical areas, for example, vision-language intersection and language-action/plan grounding. The project theme in Spring 2023 is "movie making" that presents various robotics and machine learning challenges ranging from content generation such as scenario generation or scene/video synthesis/editing to robotics automation such as autonomous camera control or autonomous stop-motion control. This course is composed of 50% lectures and 50% seminar classes. The course objectives will also put a special emphasis on learning research skills, e.g., problem formulation, literature review, ideation, evaluation planning, results analysis, and hypothesis verification. The course is discussion intensive, and thus attendance is required.

Course Website: <http://www.cs.cmu.edu/~jeanoh/16-785/>

**16-791 Applied Data Science**

Spring: 12 units

This course explores the rapidly developing field of data science in the context of its pragmatic applications. Applied Data Science strives to achieve three main goals. The first is to optimize the efficacy of decision making by human managers. The second is to maximize the utilization of available data, so that no important clue is ever missed. The third is to improve understanding of data and the underlying processes that produce it. This course aims at building skills required to systematically achieve those goals in practice. The students will gain and solidify awareness of the most prevalent contemporary methods of Data Science, and develop intuition needed for assessing practical utility of the studied topics in application scenarios. They will be able to learn how to formulate analytic tasks in support of project objectives, how to define successful analytic projects, and how to evaluate utility of existing and potential applications of the discussed technologies in practice.

**16-792 Applied Machine Learning**

Intermittent

This course explores the rapidly developing field of machine learning in the context of its pragmatic applications. The domain of Applied Machine Learning strives to achieve three main goals. The first is to build effective models to optimize the efficacy of decision-making. The second is to maximize the utilization of available data so that no important clue is ever missed. The third is to gain or improve an understanding of data and the underlying processes that produce it. Students are required to register for 9 units to receive credit for lectures but may also register for 12 units which will include 3 units of capstone project.

**16-820 Advanced Computer Vision**

Fall: 12 units

16-820 is a required core course for MSCV students and is intended to move at a slightly faster pace compared to 16-720. This course introduces the fundamental techniques used in computer vision, that is, the analysis of patterns in visual images to reconstruct and understand the objects and scenes that generated them. Topics covered include camera geometry and calibration, multi-view stereo, 3D reconstruction, image detection, segmentation, and tracking, and physics-based vision. The homeworks involve considerable Python programming exercises.

**16-823 Physics-based Methods in Vision (Appearance Modeling)**

Intermittent: 12 units

Everyday, we observe an extraordinary array of light and color phenomena around us, ranging from the dazzling effects of the atmosphere, the complex appearances of surfaces and materials, and underwater scenarios. For a long time, artists, scientists, and photographers have been fascinated by these effects, and have focused their attention on capturing and understanding these phenomena. In this course, we take a computational approach to modeling and analyzing these phenomena, which we collectively call "visual appearance". The first half of the course focuses on the physical fundamentals of visual appearance, while the second half of the course focuses on algorithms and applications in a variety of fields such as computer vision, graphics and remote sensing and technologies such as underwater and aerial imaging.

Prerequisites: 16-720 or 16-385 or 16-820 or 15-462

Course Website: <http://www.cs.cmu.edu/afs/cs/academic/class/16823-f06/>

**16-824 Visual Learning and Recognition**

Spring: 12 units

This graduate-level computer vision course focuses on representation and reasoning for large amounts of data (images, videos, associated tags, text, GPS locations, etc.) toward understanding the visual world surrounding us. We will be reading an eclectic mix of classic and recent papers on topics including Theories of Perception, Mid-level Vision (Grouping, Segmentation, Poses), Object and Scene Recognition, 3D Scene Understanding, Action Recognition, Multimodal Perception, Language and Vision Models, Deep Generative Models, Efficient Neural Networks, and more. We will cover a wide range of supervised, semi-supervised, self-supervised, and unsupervised approaches for each topic above.

Prerequisites: 16-720 Min. grade B or 16-722 Min. grade B or 10-701 Min. grade B or 16-385 Min. grade B or 15-781 Min. grade B

Course Website: <https://visual-learning.cs.cmu.edu/>

**16-825 Learning for 3D Vision**

Spring: 12 units

Any autonomous agent we develop must perceive and act in a 3D world. The ability to infer, model, and utilize 3D representations is therefore of central importance in AI, with applications ranging from robotic manipulation and self-driving to virtual reality and image manipulation. While 3D understanding has been a longstanding goal in computer vision, it has witnessed several impressive advances due to the rapid recent progress in (deep) learning techniques e.g. differentiable rendering, single-view 3D prediction. The goal of this course is to explore this confluence of 3D Vision and Learning-based Methods.

**16-831 Introduction to Robot Learning**

Fall and Spring: 12 units

Robots need to make sequential decisions to operate in the world and generalize to diverse environments. How can they learn to do so? This is what we call the "robot learning" problem and it spans topics in machine learning, visual learning and reinforcement learning. In this course, we will learn the fundamentals of topics in machine/deep/visual/reinforcement-learning and how such approaches are applied to robot decision making. We will study fundamentals of: 1) machine (deep) learning with emphasis on approaches relevant for cognition, 2) reinforcement learning: model-based, model-free, on-policy (policy gradients), off-policy (q-learning), etc.; 2) imitation learning: behavior cloning, dagger, inverse RL and offline RL; 3) visual learning geared towards cognition and decision making including topics like generative models and their use for robotics, learning from human videos, passive internet videos, language models; and 4) leveraging simulations, building differentiable simulations and how to transfer policies from simulation to the real world; 5) we will also briefly touch topics in neuroscience and psychology that provide cognitive motivations for several techniques in decision making. Throughout the course, we will look at many examples of how such methods can be applied to real robotics tasks as well as broader applications of decision making beyond robotics (such as online dialogue agents etc.). The course will provide an overview of relevant topics and open questions in the area. There will be a strong emphasis on bridging the gap between many different fields of AI. The goal is for students to get both the high-level understanding of important problems and possible solutions, as well as low level understanding of technical solutions. We hope that this course will inspire you to approach problems in cognition and embodied learning from different perspectives in your research. (As of 3/21/2023)

Course Website: [https://docs.google.com/document/d/15FCdFSLj9uDKQgY\\_bdQgic0n2c5yYBbedVjKiRFTuM/edit?usp=sharing](https://docs.google.com/document/d/15FCdFSLj9uDKQgY_bdQgic0n2c5yYBbedVjKiRFTuM/edit?usp=sharing) ([https://docs.google.com/document/d/15FCdFSLj9uDKQgY\\_bdQgic0n2c5yYBbedVjKiRFTuM/edit?usp=sharing](https://docs.google.com/document/d/15FCdFSLj9uDKQgY_bdQgic0n2c5yYBbedVjKiRFTuM/edit?usp=sharing))

**16-833 Robot Localization and Mapping**

Spring: 12 units

Robot localization and mapping are fundamental capabilities for mobile robots operating in the real world. Even more challenging than these individual problems is their combination: simultaneous localization and mapping (SLAM). Robust and scalable solutions are needed that can handle the uncertainty inherent in sensor measurements, while providing localization and map estimates in real-time. We will explore suitable efficient probabilistic inference algorithms at the intersection of linear algebra and probabilistic graphical models. We will also explore state-of-the-art systems.

Course Website: <http://frc.ri.cmu.edu/~kaess/teaching/16833/Spring2018> (<http://frc.ri.cmu.edu/~kaess/teaching/16833/Spring2018/>)

**16-845 Insects and Robots**

Fall: 12 units

This course will cover all facets of modeling, design, fabrication, and analysis of robots operating on the insect scale, with a microrobotics perspective. Insects can perform different tasks, such as manipulation or locomotion, with their small scale bodies varying from 200µm to 16cm length. Similarly, we can define a micro-robotic system as an autonomous or semi-autonomous device with features on the micron scale or that make use of micron-scale physics for mobility or manipulation of objects. Due to their small size scales, microrobots will encounter difficulties unlike their macro-scale counterparts, in terms of fabrication and autonomy. In this project-based course, our aim will be on learning the physics of scaling, fabrication paradigms, actuation and sensing strategies, with numerous case studies, and to build an insect-inspired robotic system. We will also discuss multiple applications such as surgical robotics, mobile microrobots, multi-agent systems, and micro/nano manipulation.

**16-848 Hands: Design and Control for Dexterous Manipulation**

Spring: 12 units

Research related to hands has increased dramatically over the past decade. Robot hand innovation may be at an all time high, with new materials and manufacturing techniques promoting an explosion of ideas. Hands have become a priority in virtual reality and telepresence. Even the study of how people use their hands is seeing the growth of new ideas and themes. With all of this attention on hands, are we close to a breakthrough in dexterity, or are we still missing some things needed for truly dexterous manipulation? In this course, we will survey robotic hands and learn about the human hand with the goal of pushing the frontiers on hand design and control for dexterous manipulation. We will consider the necessary kinematics and dynamics for dexterity, what sensors are required to carry out dexterous interactions, the importance of reflexes and compliance, the role of machine learning in grasping and manipulation, and the challenge of uncertainty. We will explore state of the art manufacturing and design techniques, including innovations in soft robotics and embedded sensing. We will examine the human hand: its structure, sensing capabilities, human grasp choice and control strategies for inspiration and benchmarking. Students will be asked to present one or two research papers, participate in discussions and short research or design exercises, and carry out a final project.

Course Website: <http://graphics.cs.cmu.edu/nsp/course/16899-s18/>

**16-855 Special Topics: Tactile Sensing and Haptics**

Spring: 12 units

Touch is an important perception modality for both humans and robots. This course aims at providing an overview of the touch perception system for both robots and humans, and provide students with some hands-on experience with the popular touch sensors and devices. On the side of robot sensing, the course will cover the topics on the working principles and designs of robot touch sensors, signal processing algorithms for tactile sensing, and the application of tactile sensing in different robotic tasks; on the side of haptics, the course will introduce the neurological and cognitive study in human haptic system, and the designs and applications of haptic devices that provide a human-machine interface. The human-machine interface is a core part of Virtual Reality (VR) and teleoperation of robots when touch is involved. The course includes lectures, research paper presentation and discussion, and course projects with tactile sensors or haptic devices.

**16-873 Spacecraft Design-Build-Fly Laboratory**

Fall and Spring: 12 units

Spacecraft design is a truly interdisciplinary subject that draws from every branch of engineering. This course integrates broad skillsets from mechanical engineering, electrical and computer engineering, computer science, and robotics toward the goal of designing, building, testing, and flying a small spacecraft over the course of two semesters. Students will engage directly in all aspects of the spacecraft mission lifecycle from initial requirements definition through mission operations. YES, WE ARE REALLY GOING TO LAUNCH A SATELLITE INTO SPACE AS PART OF THIS COURSE. Students will work in subsystem teams, each focusing on some aspect of the spacecraft, but will be exposed to many different disciplines and challenges. Practical, hands-on, engineering skills will be emphasized, along with building and testing physical hardware and flight software.

**16-878 Special Topic: Advanced Mechatronic Design**

Fall: 12 units

This course is designed for students who have a background in mechatronics by having taken a mechatronics design course or through practice. The course will be a combination of laboratories and lectures and will culminate in a class project. The topics covered will be microcontroller hardware subsystems: timer systems, PWM, interrupts; analog circuits, operational amplifiers, comparators, signal conditioning, interfacing to sensors, actuator characteristics and interfacing; C language features for embedded software, register level programming, hardware abstraction layers, event driven programming, state machines, state charts.

**16-879 Medical Robotics**

Fall: 12 units

This course presents an overview of medical robotics intended for graduate students and advanced undergraduates. Topics include robot kinematics, registration, navigation, tracking, treatment planning, and technical and medical aspects of specific applications. The course will include guest lectures from robotics researchers and surgeons, as well as observation of surgical cases. The course is open to non-majors who have the requisite background.

**16-880 Special Topics: Engineering Haptic Interfaces**

Spring: 12 units

This course focuses on addressing challenges in the field of haptics from an engineer's perspective. We will begin by studying human haptic perception and an introduction into psychophysics. We will then study the design and control of haptic systems which provide touch feedback to a user. The class format will include lectures, discussion, paper presentations, laboratories and assignments using hardware that will be shipped to the students, and a class project. This class is designed to be a graduate/advanced undergraduate course and requires a background in dynamic systems, mechatronics, and basic programming. Mechanical prototyping, robotics, and feedback control knowledge are useful skills for this class but are not required.

**16-881 Seminar Deep Reinforcement Learning for Robotics**

Spring: 12 units

Deep RL has a lot of promise to teach robots how to choose actions to optimize sequential decision-making problems, but how can we make deep RL work in the real world? This is a seminar course in which we read papers related to deep learning for robotics and analyze the tradeoffs between different approaches. We will read mostly state-of-the-art papers that were very recently published (e.g. recent CoRL, RSS), but we will also look at some older papers that use different approaches. The goals of the course are to 1) understand what is needed to make deep learning work for robotics 2) analyze the tradeoffs between different approaches. Each class, 2 papers will be presented. These papers will both achieve a similar robotics task but will use different learning-based approaches. The class will discuss these papers and try to understand the strengths and limitations of the approach described in each paper. The list of papers that we will be discussing this year is still to be determined; please see the website for the list of papers that we have used in past semesters: <https://sites.google.com/view/16-881-cmu/paper-lists?authuser=0> The seminar is a great followup course to 16-831, 16-884, 10-403, or 10-703.

Course Website: <https://sites.google.com/view/16-881-cmu/home?authuser=0> (<https://sites.google.com/view/16-881-cmu/home/?authuser=0>)

**16-882 Integrated Systems Engineering and Project Management**

Spring: 12 units

Systems Engineering is a top-down, interdisciplinary, and systematic approach to conceptualizing, developing, using, and retiring systems. Project Management is a bottom-up approach that applies methods, skill, and knowledge to plan, monitor, control, and close out projects. In a broad sense, a system is a construct that consists of elements working together under well-defined requirements to achieve a unique purpose; a project is a finite team activity that aims to create a unique prototype, product, or system at a certain level of fidelity and maturity. In this course students will learn the theory, essential methods, and techniques of systems engineering and project management. They will apply these combined concepts and skills to develop a system of their choosing over the course of the semester, either independently or by collaborating with a team. Even though this course is offered by the Robotics Institute and some of its content will speak to the application of the concepts in robotic applications and systems, this course is open to and suited for all CMU students irrespective of discipline. In addition to graduate students, this course is also suited for juniors and seniors.

**16-883 Special Topics: Provably Safe Robotics**

Spring: 12 units

Safe autonomy has become increasingly critical in many application domains. It is important to ensure not only the safety of the ego robot, but also the safety of other agents (humans or robots) that directly interact with the autonomy. For example, robots should be safe to human workers in human-robot collaborative assembly; autonomous vehicles should be safe to other road participants. For complex autonomous systems with many degrees of freedom, safe operation depends on the correct functioning of all system components, i.e., accurate perception, optimal decision making, and safe control. This course deals with both the design and the verification of safe robotic systems. From the design perspective, we will talk about how to assure safety through planning, prediction, learning, and control. From the verification perspective, we will talk about verification of deep neural networks, safety or reachability analysis for closed loop systems, and analysis of multi-agent systems.

Course Website: <http://www.cs.cmu.edu/~cliu6/provably-safe-robotics.html>

**16-884 Deep Learning for Robotics**

Fall: 12 units

The goal of this course is to study relevant topics towards building intelligent robots that can learn to act and perceive in the real world. The course material should be a self-contained collection of key topics from the intersection of four research areas geared towards this common goal: a) Robot Learning and amp; Deep RL; (b) Computer Vision; (c) Control; (d) Psychology and amp; Neuroscience. This course is geared mainly towards learning and brainstorming. There will be two classes every week. In this first class, instructor will present an in-depth overview of a topic, and then in the second class, students will present instructor-assigned papers related to that topic. There will be no homeworks and just a course project. In the first quarter, we will cover state-of-the-art topics in robot learning (deep RL, inverse RL, etc.) and control (optimal control, dynamic movement primitives, etc.) by studying classical and recent papers in the area. In the second quarter, we will study the role of perception in control and vice-versa to build methods that can learn from high-dimensional raw sensory input. In the third quarter, we will discuss the state of the current understanding of how the brain integrates action and perception. We will also discuss relevant papers from ontogeny (child development literature in Psychology) and phylogeny (evolutionary development literature in Biology) of biological animals that have inspired ideas in learning and robotics. Finally, in the fourth quarter, we will bring these ideas together to brainstorm potential high-level directions that could guide the development of intelligent robots.

**16-885 Special Topics: Tactile Sensing and Haptics**

Fall: 12 units

Touch is an important perception modality for both humans and robots. This course aims at providing an overview of the touch perception system for both robots and humans. On the side of robot sensing, the course will cover the designs of robot touch sensors, signal processing algorithms for tactile sensing, and the application of tactile sensing in different robotic tasks; on the side of haptics, the course will introduce the neurological and cognitive study in the human haptic system, and the designs and applications of haptic devices that provide a human-machine interface. The course incorporates lectures, research paper presentations, and discussion. The combination of different modules aims to present both the basics and state-of-art research directions in the field.

**16-887 Special Topic: Robotic Caregivers and Intelligent Physical Collaboration**

Spring: 12 units

Robotics researchers and futurists have long dreamed of robots that can serve as caregivers. In this project-based course, you'll learn about intelligent physical human-robot collaboration and opportunities for robots that contribute to caregiving. You'll gain hands-on experience with teleoperation, autonomy, perception, navigation, manipulation, human-robot interaction, and machine learning. You'll also learn about robot design, collaborative research, and healthcare robotics.

Course Website: <https://zackory.com/rc2023/>

**16-888 Special Topic: Foldable Robots: Origami-inspired design meets mechatronics**

Intermittent: 12 units

The way we make robots has changed dramatically since the limitations on the material space was removed. Instead of using "nuts-and-bolts" approach that helped us to make robust, rigid, industrial robots, we can make light-weight, compliant, conformable robots out of paper, fabric, and polymers. In this class, we will explore foldable robots with a multifaceted perspective: Kinematics, design, fabrication, control, and application. We will design and manufacture mechanisms for targeted applications, such as manipulation, bio-inspiration, medical, architecture, using laminates with integrated joints and limited number of actuators.

**16-889 Special Topic: Learning for 3D Vision**

Spring: 12 units

Any autonomous agent we develop must perceive and act in a 3D world. The ability to infer, model, and utilize 3D representations is therefore of central importance in AI, with applications ranging from robotic manipulation and self-driving to virtual reality and image manipulation. While 3D understanding has been a longstanding goal in computer vision, it has witnessed several impressive advances due to the rapid recent progress in (deep) learning techniques e.g. differentiable rendering, single-view 3D prediction. The goal of this course is to explore this confluence of 3D Vision and Learning-based Methods.

**16-890 Special Topic: Robot Cognition for Manipulation**

Intermittent: 12 units

This seminar course will cover a mixture of modern and classical methods for robot cognition. We will review papers related to task planning and control using both symbolic and numeric methods. The goal of this course is to give students an overview of the current state of research on robot cognition.

**16-891 Multi-Robot Planning and Coordination**

Spring: 12 units

The course provides a graduate-level introduction to the field of multi-robot planning and coordination from both AI and robotics perspectives. Topics for the course include multi-robot cooperative task planning, multi-robot path/motion planning, learning for coordination, coordinating robots under uncertainty, etc. The course will particularly focus on state-of-the-art Multi-Agent Path Finding (MAPF) algorithms that can coordinate hundreds of robots with rigorous theoretical guarantees. Current applications for these technologies will be highlighted, such as mobile robot coordination for warehouses, drone swarm control, and multi-arm assembly. The course includes lectures, research paper presentations and discussions, and course projects.

**16-892 Seminar: Multimodal Foundational Models**

Fall: 12 units

This course will discuss recent foundation models proposed in the literature, with a focus on vision-language models. Topics include large language models, vision-language models, and vision-audio models. As time allows, this course will also discuss application of such models to visual, audio, and video content generation.

Prerequisite: 16-820

**16-895 Understanding and Critiquing Generative Computer Vision**

Spring: 12 units

In recent years, there have been significant advances in the field of large-scale generative modeling for visual data, such as DALL-E 2 and Stable Diffusion. This seminar course explores these advances beyond just reading and discussion. The goal is to not only inform state of the art but also develop critical and philosophical thinking among students. The course will involve reading papers, presentations, and discussions. The course will also involve reviewing and developing critical thinking.

## Software Societal Systems Courses

**17-200 Ethics and Policy Issues in Computing**

Fall and Spring: 9 units

Should autonomous robots make life and death decisions on their own? Should we allow them to select a target and launch weapons? To diagnose injuries and perform surgery when human doctors are not around? Who should be permitted to observe you, find out who your friends are, what you do and say with them, what you buy, and where you go? Do social media and personalized search restrict our intellectual horizons? Do we live in polarizing information bubbles, just hearing echoes of what we already know and believe? As computing technology becomes ever more pervasive and sophisticated, we are presented with an escalating barrage of decisions about who, how, when, and for what purposes technology should be used. This course will provide an intellectual framework for discussing these pressing issues of our time, as we shape the technologies that in turn shape us. We will seek insight through reading, discussion, guest lectures, and debates. Students will also undertake an analysis of a relevant issue of their choice, developing their own position, and acquiring the research skills needed to lend depth to their thinking. The course will enhance students' ability to think clearly about contentious technology choices, formulate smart positions, and support their views with winning arguments.

**17-210 Introduction to Social Networks**

Spring: 12 units

Course Description What makes a recommendation system on social media effective? Why do YouTube mega-influencers with tens of millions of subscribers exist, yet we have not heard of most of them? Why do echo chambers form and polarization deepen in social media platforms despite the utopian promise of the free flow of information and fluid connectivity between people that these platforms had envisioned? How do professionals land their dream jobs? How does mass adoption of technological innovations happen (e.g., Python, Bitcoin)? Underlying these seemingly unrelated questions is the powerful influence of social networks, the collection of social connections that people form on and offline. This course offers an introduction to the study of social networks as a powerful tool for understanding a wide range of questions and for solving real-world problems arising at the intersection of human social behavior and technological systems. The course first introduces network concepts and their mathematical operationalizations that give social network analysis the versatility for rigorous quantitative analysis. Students will subsequently learn how these building blocks have been applied to analyze and solve a wide range of online social phenomena. Finally, the course will introduce statistical models of networks that enable principled investigation of network formation mechanisms. Throughout this course, students will work towards a final team project that applies network concepts and measures to either (a) describe and/or explain an empirically puzzling social phenomenon with network data, (b) develop a network data collection pipeline that aims to solve a real-world problem, or (c) build a system utilizing the insights from the course that solves a real-world problem. The following class activities throughout the course are intended to support and augment the final project.

**17-213 Network Analysis: The Hidden Structures behind the Webs We Weave**

Fall: 12 units

Have you wondered how Linux and the subsequent development of open-source software ecosystems could thrive despite weak economic incentives? Why do large-scale complex software systems sometimes fail despite well-developed guardrails and practices? What makes a recommendation system on social media so effective and so toxic at the same time? Why do YouTube mega-influencers with tens of millions of subscribers exist, yet each of us only recognize a handful of them at best? Why do echo chambers form and polarization deepen in social media platforms despite the utopian promise of frictionless connectivity that these platforms initially envisioned? How can you land your dream jobs? How does mass adoption of technological innovations happen (e.g., Python, Bitcoin)? Underlying these seemingly unrelated questions is the powerful influence of social networks, the collection of on- and offline social connections, communications, and collaborations that people form with one another. This course offers an introduction to the study of social networks as a powerful tool for formalizing these wide range of questions, for understanding social dynamics in various settings, and for solving real-world problems arising at the intersection of human social behavior and technological systems. The course first introduces network science concepts and their mathematical operationalizations that give rigorous definitions to fuzzy words we use to describe the social world, such as "status" and "social group". Students will subsequently learn how these network concepts were used to analyze and solve a wide range of puzzling online social phenomena. Finally, the course will introduce statistical models of networks that enable principled investigation of network formation mechanisms.

**17-214 Principles of Software Construction: Objects, Design, and Concurrency**

Fall and Spring: 12 units

Software engineers today are less likely to design data structures and algorithms from scratch and more likely to build systems from library and framework components. In this course, students engage with concepts related to the construction of software systems at scale, building on their understanding of the basic building blocks of data structures, algorithms, and program and computer structures. The course covers technical topics in four areas: (1) concepts of design for complex systems, (2) object-oriented programming, (3) static and dynamic analysis for programs, and (4) concurrency. At the conclusion of this course, students will have substantial experience building medium-sized software systems in Java or JavaScript.

Prerequisites: (15-122 Min. grade C or 15-121 Min. grade C) and (21-127 Min. grade C or 21-128 Min. grade C or 15-151 Min. grade C)

Course Website: <https://www.cs.cmu.edu/~ckaestne/17214/f2021/>

**17-224 Influence, Persuasion, and Manipulation Online**

Fall: 9 units

This course will introduce the fundamental behavioral science of influence, persuasion, and manipulation, and the application of these scientific principles to online campaigns to influence attitudes and behavior. In particular, we will discuss the psychology of persuasion, nudging, social influence, bias, persuasive design, and the ethics of persuasion. Against this background, we will analyze case studies drawn from recent, high profile events such as election campaigns, targeted advertising, sowing political division, memes and virality, impact of social media, and propagation of "fake news." Countermeasures to these tactics will be explored, including personal measures, technologies, and policy.

**17-301 Special Topics in Societal Computing: Understanding Cyber Teams**

Intermittent: 9 units

Were you the victim of a ransomware attack today? No? Then you should probably thank your cyber security team! Nowadays, nearly everything modern organizations do is protected by some semblance of a cyber team. These cyber teams come in many forms: security operations center, malware analysis, threat emulation, information technology departments, etc. So what do we know about how these cyber teams operate? In this course cyber teams will be studied through the lens of computational organization theory. Different types of cyber teams will be presented with how they operate within the organization. An array of computational organization theory principles will be explored such as network science, information diffusion, policy analysis, and performance theory. A final project will be completed by each student. This project can be of many types such as policy analysis, virtual experimentation, dynamic network analysis, validation, etc. Students will be using the NetLogo scripting language to complete a project. The project will be created, developed, and presented by the student and approved by the instructor.

**17-302 Independent Study Mini Undergraduate**

Intermittent

Independent Study Mini

**17-303 Cryptocurrencies, Blockchains and Applications**

Fall and Spring: 9 units

Cryptocurrencies such as Bitcoin have gained large popularity in recent years, in no small part due to the fantastic potential applications they could facilitate. This course will first provide an overview of the technological mechanisms behind cryptocurrencies and distributed consensus and distributed ledgers ("blockchains"), introducing along the way the necessary cryptographic tools. It will then focus on more advanced blockchain applications, such as "smart contracts," that is, contracts written as code. Finally, the course will also introduce some of the legal and policy questions surrounding cryptocurrencies.

**17-313 Foundations of Software Engineering**

Fall and Spring: 12 units

Students gain exposure to the fundamental principles of software engineering. This includes both core CS technical knowledge and the means by which this knowledge can be applied in the practical engineering of complex software in real-world settings. Topics related to software artifacts include coding, software architecture, measurement, and quality assurance of various qualities (e.g., robustness, security, performance, maintainability) with static and dynamic analysis, testing, code review, and inspection. Topics related to software process include requirements engineering, process models and evaluation, personal and team development, and supply chain issues including outsourcing and open source. This course has a strong technical focus, a strong focus on developing team skills, and will include both written and programming assignments. Students will get experience with the latest software engineering tools and practices.

Course Website: <https://www.cs.cmu.edu/~ckaestne/17313/>**17-314 Formal Methods**

Fall: 6 units

Scientific foundations for software engineering depend on the use of precise, abstract models for describing and reasoning about properties of software systems. This course considers a variety of standard models for representing sequential and concurrent systems, such as state machines, algebras, and traces. It shows how different logics can be used to specify properties of systems, such as functional correctness, deadlock freedom, and internal consistency. Concepts such as compositionality, abstraction, invariants, non-determinism, and inductive definitions are recurrent themes throughout the course. After completing this course, students will: 1. Understand the strengths and weaknesses of certain models and logics including state machines, algebraic and process models, and temporal logic; 2. Be able to select and describe appropriate abstract formal models for certain classes of systems, describe abstraction relations between different levels of description, and reason about the correctness of refinements; 3. Be able to prove elementary properties about systems described by the models introduced in the course; and 4. Understand some of the strengths and weakness of formal automated reasoning tools. Prerequisites: Undergraduate discrete math including first-order logic, sets, functions, relations, and simple proof techniques such as induction.

**17-322 Agile Methods**

Fall: 6 units

Agile methods refers to a number of software development approaches that adopt self-organization, adaptive planning, evolutionary development, frequent delivery and working closely with and incorporating feedback from customers throughout the development process as their principles of operation to achieve responsiveness. This course will introduce students to two well known agile methods: Scrum and Kanban, connecting their practices to established group dynamics and knowledge management theories to explain why they work and under what circumstances

**17-323 Quality Assurance**

Fall: 6 units

This class is fundamentally about software quality assurance and control. This course will introduce various quality assurance tools and techniques to software engineering students. Students will build their "quality toolbox" not only with useful tools and techniques, but with the knowledge of when those tools should be used, how to evaluate their results, and what assurances they can provide. The key learning objectives of the course include: 1. Understand software quality: how to define it, analyze it, and measure it. 2. Select the proper analytical tool/technique for a given situation and explore how to analyze results. 3. Understand the strengths and weaknesses of different quality assurance techniques, such as software testing, static analysis, code review, and demonstration. 4. Learn to collect, manage, and evaluate quality metrics. 5. Analyze and verify a variety of software properties including, but not limited to, functionality, security, reliability, and performance. 6. Gain experience with real quality assurance tools including static analysis tools, software testing frameworks, and software quality measurement tools

**17-324 Advanced Formal Methods**

Fall: 6 units

This course builds on the introductory Models class to cover more advanced techniques for modeling and reasoning about complex software systems. Concepts introduced in this course include abstraction and refinement, declarative specifications, advanced temporal logics, and probabilistic modeling. The course will also explore applications of modeling and automated reasoning techniques in various domains, such as security, distributed computing, and cyber-physical systems. After completing this course, students will: 1. Understand how to specify and reason about operations over complex system structures, 2. Understand relationships between software artifacts at different levels of abstraction; 3. Be able to model and reason about systems with uncertainty and stochastic behaviors; and 4. Understand potential applications of modeling techniques to practical software engineering problems. Prerequisites: Completion of Mini 1: Models of Software Systems. Sections D, PP and G are NOT available for on-campus students. Admission to the class is by approval from the instructor: If you are not a software engineering master's student, send email to [garlan@cs.cmu.edu](mailto:garlan@cs.cmu.edu) for permission to enroll. The email should briefly describe your background, whether you have taken a course with similar materials as in Mini 1, and why you would like to take the course. The course must be taken for a letter grade (not pass/fail). This is a graduate level course.

Prerequisite: 17-314 Min. grade B

**17-331 Information Security, Privacy, and Policy**

Fall: 12 units

As layers upon layers of technology mediate increasingly rich business processes and social interactions, issues of information security and privacy are growing more complex too. This course takes a multi-disciplinary perspective of information security and privacy, looking at technologies as well as business, legal, policy and usability issues. The objective is to prepare students to identify and address critical security and privacy issues involved in the design, development and deployment of information systems. Examples used to introduce concepts covered in the class range from enterprise systems to mobile and pervasive computing as well as social networking. Format: Lectures, short student presentations on topics selected together with the instructor, and guest presentations. Target Audience: Primarily intended for motivated undergraduate and masters students with CS background. Also open to PhD students interested in a more practical, multi-disciplinary understanding of information security and privacy.

**17-332 Software Project Management**

Spring: 6 units

Projects are temporary organizations set up to achieve a one time objective in an agreed time frame. They are characterized by requiring the execution of interrelated, normally non repeating activities, by multidisciplinary groups. Because of its temporary nature and the interrelatedness of its activities, projects require prescriptive planning, budgeting, staffing and risk management. This course will introduce student to fundamental project management techniques and tools such as activity planning, milestone planning, estimation, work breakdown structures, critical paths. The course will also look at hybrid methods such as Milestone Driven Agile Execution and Disciplined Agile Delivery.

**17-333 Privacy Policy, Law, and Technology**

Fall: 9 units

This course focuses on policy issues related to privacy from the perspectives of governments, organizations, and individuals. We will begin with a historical and philosophical study of privacy and then explore recent public policy issues. We will examine the privacy protections provided by laws and regulations, as well as the way technology can be used to protect privacy. We will emphasize technology-related privacy concerns and mitigation, for example: social networks, smartphones, behavioral advertising (and tools to prevent targeted advertising and tracking), anonymous communication systems, big data, and drones. This is part of a series of courses offered as part of the MSIT-Privacy Engineering masters program. These courses may be taken in any order or simultaneously. Foundations of Privacy (Fall semester) offers more in-depth coverage of technologies and algorithms used to reason about and protect privacy. Engineering Privacy in Software (Spring semester) focuses on the methods and tools needed to design systems for privacy. This course is intended primarily for graduate students and advanced undergraduate students with some technical background. Programming skills are not required. 8-733, 19-608, and 95-818 are 12-unit courses for PhD students. Students enrolled under these course numbers will have extra assignments and will be expected to do a project suitable for publication. 8-533 is a 9-unit course for undergraduate students. Masters students may register for any of the course numbers permitted by their program. This course will include a lot of reading, writing, and class discussion. Students will be able to tailor their assignments to their skills and interests. However, all students will be expected to do some writing and some technical work.

**17-334 Usable Privacy and Security**

Spring: 9 units

There is growing recognition that technology alone will not provide all of the solutions to security and privacy problems. Human factors play an important role in these areas, and it is important for security and privacy experts to have an understanding of how people will interact with the systems they develop. This course is designed to introduce students to a variety of usability and user interface problems related to privacy and security and to give them experience in designing studies aimed at helping to evaluate usability issues in security and privacy systems. The course is suitable both for students interested in privacy and security who would like to learn more about usability, as well as for students interested in usability who would like to learn more about security and privacy. Much of the course will be taught in a graduate seminar style in which all students will be expected to do a weekly reading assignment and each week different students will prepare a presentation for the class. Students will also work on a group project throughout the semester. The course is open to all graduate students who have technical backgrounds. The 12-unit course numbers (08-734 and 5-836) are for PhD students and masters students. Students enrolled in these course numbers will be expected to play a leadership role in a group project that produces a paper suitable for publication. The 9-unit 500-level course numbers (08-534 and 05-436) are for juniors, seniors, and masters students. Students enrolled in these course numbers will have less demanding project and presentation requirements.

**17-335 Software Architectures**

Spring: 6 units

Successful design of complex software systems requires the ability to describe, evaluate, and create systems at an architectural level of abstraction. This course introduces architectural design of complex software systems. The course considers commonly-used software system structures, techniques for designing and implementing these structures, models and formal notations for characterizing and reasoning about architectures, tools for generating specific instances of an architecture, and case studies of actual system architectures. It teaches the skills and background students need to evaluate the architectures of existing systems and to design new systems in principled ways using well-founded architectural paradigms. After completing this course, students will be able to: 1. describe an architecture accurately 2. recognize major architectural styles in existing software systems 3. generate architectural alternatives for a problem and choose among them 4. construct a medium-sized software system that satisfies an architectural specification 5. use existing definitions and development tools to expedite such tasks 6. understand the formal definition of a number of architectures and be able to reason about the properties of those architectures 7. use domain knowledge to specialize an architecture for a particular family of applications.

**17-336 Applied Distributed Systems**

Spring: 6 units

Modern computing systems are frequently hosted on the cloud. That is, they are inherently distributed systems. To appropriately build and deploy these systems developers should know not only about development tools such as container management tools but also the structure of the cloud - in particular how it utilizes virtual machines, containers and networks. They should also understand security mechanisms both in the internet and how to authorize users and maintain credentials securely. Finally, to protect the system once it is placed into production, a developer needs to know how to enable the detection of problems during execution through collection and navigation of logs produced by the system. These are the topics covered by this course.

**17-340 Green Computing**

Intermittent: 9 units

Note: Previously offered as 08-340. Energy is a key societal resource. However, our energy usage is rising at an alarming rate and therefore it has become critical to manage its consumption more efficiently for long term sustainability. This course introduces students to the exciting area of "Green Computing", and is organizationally divided into two tracks. The first track is "Energy-Efficient Computing", which considers the state of the art techniques for improving the energy efficiency of mobile devices, to laptop and desktop class computers and finally to data centers. We will cover energy efficiency across the hardware/software stack, starting from the individual components like processors and radio interfaces to system level architectures and optimizations. The second track is "Applying Computing towards Sustainability", covering topics that leverage computing to reduce the energy footprint of our society. In particular, we will focus on Smart Buildings and the Smart Grid, covering topics such as sensing, modeling and controlling the energy usage of buildings, new operating systems or software stacks for the smart infrastructure, as well as the privacy and security issues with the new "internet of things". The goal of this course is to help students acquire some of the knowledge and the skills needed to do research in this space of "Green Computing". Although the course is listed within SCS, it should be of interest to students in several departments, including ECE, MechE, CEE, EPP and Architecture.

**17-346 DevOps and Continuous Integration**

Spring: 6 units

DevOps: Engineering for Deployment and Operations": DevOps is the term given to a modern movement to establish practices that significantly reduce the time to production of committed code. This time involves deployment - the period between the completion of the code by the developers and the placing of the code into normal production and dealing with operations issues. Deployment time can be days, weeks, or even months when using normal development practices. Operational issues such as dealing with incidents and errors introduce other delays. Modern internet companies deploy a system multiple or even dozens of times every day. Achieving this velocity requires coordinated process and design activities together with supporting tooling. This course will cover the deployment process and the associated tooling, it will highlight reasons why release schedules can be slow, and it will introduce the practices that are used to enable high velocity deployments. It will also cover the kinds of problems that are created because of high velocity and how modern internet companies deal with these problems. Please note: This is a required course for MSE-SS students. Students outside of the software engineering department may take this course but students of the MSE programs will have first priority. Prerequisite: 17-336 Min. grade B

**17-350 Information Technology Policy: Evidence, Communication, & Advocacy**

Spring: 9 units

In recent decades, developments in Information and Communication Technologies (ICTs) have rapidly moved from research environments to products and services used by billions of people. This rapid rate of change has often resulted in a public which does not understand the technologies shaping their lives and lawmakers who are poorly equipped to make sound policy. It is therefore incumbent upon specialists to communicate how ICTs work to the public and lawmakers so policy making is shaped by evidence and reflects public desires. This course will train students to be effective communicators and advocates in the ICT space. Students taking this course will learn about the broader scope of technology policymaking including formal lawmaking, agency rule-making, strategic litigation, and corporate social responsibility. Current ICT policy topics in privacy, free expression, net neutrality, and competition will be covered. Public communication strategies such as writing op-eds, interviewing with journalists, producing explanatory videos and interactive games will be explored. Finally, students will learn how to perform an expert role in areas such as writing policy briefs and providing testimony. The course is open to advanced undergraduate and graduate students. Graduate students whose research has public policy implications are encouraged to develop projects related to their research. There is no requirement for programming knowledge, but students with experience in developing interactive media and games will be encouraged to utilize such skills. The class will focus heavily on readings, critical evaluation of real ICT advocacy campaigns, and homework will provide hands-on experience with numerous strategies for public engagement. At the end of the semester students will have a portfolio of projects which they may release publicly.

**17-355 Program Analysis**

Spring: 12 units

This course covers both foundations and practical aspects of the automated analysis of programs, which is becoming increasingly critical to find software errors and assure program correctness. The theory of abstract interpretation captures the essence of a broad range of program analyses and supports reasoning about their correctness. Building on this foundation, the course will describe program representations, data flow analysis, alias analysis, interprocedural analysis, dynamic analysis, Hoare Logic and verification, program synthesis and repair, model checking, and symbolic execution. Through assignments and projects, students will design and implement practical analysis tools that find bugs and verify properties of software. This course satisfies the Logic and Languages constrained elective category of the Computer Science major, the Theoretical Foundations requirement of the Computer Science master's degree, and the Technical Software Engineering requirement for the Software Engineering minor. Prerequisites: 15-251 Min. grade C and (15-150 Min. grade C or 17-214)

Course Website: <https://cmu-program-analysis.github.io/>**17-356 Software Engineering for Startups**

Spring: 12 units

Startup engineering is critical to innovation. The skills required to effectively prototype, launch, and scale products are vital to engineers everywhere, from fledgling companies founded in dorm rooms to local mid-size companies to internal startups from multi-national tech giants. However, developing software in a startup environment poses unique engineering challenges. These challenges include making and justifying foundational architectural and technical decisions despite extreme uncertainty; rapidly prototyping and evaluating new ideas and features, while building minimum viable products; prioritizing engineering effort in severely constrained environments; and communicating effectively both within a small engineering team and with internal and external non-technical stakeholders. This course teaches the skills necessary to engineer successfully in a startup environment, through lectures, group projects, case study discussions, and guest speakers drawn from experienced, practicing startup engineers. This is an engineering-focused course; no entrepreneurship background is required or expected. Students do not need to have a startup idea to participate fully. Prerequisites: 15-213 or 17-514 or 17-214 or 15-214

**17-363 Programming Language Pragmatics**

Fall: 12 units

This course provides a broad and pragmatic foundation in the most basic tool of the programmer: programming languages. It starts with the fundamentals of syntax, parsing, and binding, the core structural concepts in programming languages. The course will then cover program semantics and type systems, and students will learn to relate them with a type soundness theorem. Finally, a coverage of intermediate optimization and code generation offers the opportunity to discuss both producing efficient code and reasoning about the correctness of program transformations. Assignments involve a combination of tool-assisted formal reasoning and proofs about programming languages, and implementing these language constructs in a compiler. This course fulfills the Logic and amp; Languages constrained elective of the B.S. in Computer Science. Students with substantial math and programming experience who have not satisfied the specific prerequisites can contact the instructor for permission to enroll. Prerequisites: 15-150 Min. grade C and (15-251 Min. grade C or 21-228 Min. grade C)

Course Website: <http://www.cs.cmu.edu/~aldrich/courses/17-363/>**17-396 Language Design and Prototyping**

Spring: 12 units

Many programmers think of programming languages as having a fixed, standard set of features and #8212;but in fact, languages are being extended all the time, and new languages are constantly being developed, providing great expressive power. In this course, students will learn about techniques for designing and prototyping programming languages. Design topics include language features, a practical approach to semantics, conceptual design techniques, and examples of both general purpose and domain-specific language designs. Prototyping topics include interpreters, source-to-source translators, languages as libraries, and paper prototyping techniques used for lightweight user evaluations. In course assignments, students will practice design and prototyping techniques, implementing prototype languages in several different styles. The course will end with a project in which students design, implement, and evaluate their own programming language. Prerequisites: 17-514 or 15-213 or 17-214

Course Website: <http://www.cs.cmu.edu/~aldrich/courses/17-396/>**17-397 Intro to Qualitative Research: Social Media Apps and Video Content Creation**

Spring: 12 units

Short-video apps, such as TikTok, have revealed themselves to be highly accessible and increasingly ubiquitous in regards to online social interaction and are currently the most popular social media among mostly young(er) users. This course focuses on how humans as users interact with and understand social technology and how the use of social media apps is connected to and integrated into our everyday life by asking questions like: What are users' motivations to participate in video creation and sharing on social media? What are their practices, strategies, and routines in creating short-form videos as part of digital online culture? What do they know about socio-technical aspects in using short-video apps and creating short-form video content? How does their understanding of socio-technological aspects influence their use of social media apps? This course is designed to enable students to develop and conduct their own individual research project. For this, students will learn how to design, conduct, and analyze qualitative interviews to research socio-technical, cultural, and political perspectives of usage of and engagement with short-form videos and short-video apps, such as TikTok. The course instructor will closely mentor and supervise students' research projects throughout the semester and will provide expertise and background in qualitative methods and social media research to guide students through the research process.

**17-400 Machine Learning and Data Science at Scale**

Fall and Spring: 12 units

Datasets are growing, new systems for managing, distributing, and streaming data are being developed, and new architectures for AI applications are emerging. This course will focus on techniques for managing and analyzing large datasets, and on new and emerging architectures for applications in machine learning and data science. Topics include machine learning algorithms and how they must be reformulated to run at scale on petabytes of data, as well as data management and cleaning techniques at scale. In addition to large-scale aspects of data science and machine learning, this course will also cover core concepts of parallel and distributed computing and cloud computing, including hands-on experience with frameworks like Spark, streaming architectures like Flink or Spark Streaming, MLlib, TensorFlow, and more. The course will include programming assignments and a substantial final project requiring students to get hands-on experience with large-scale machine learning pipelines or emerging computing architectures.

Prerequisites: 10-601 or 10-701 or 15-211 or 10-301 or 17-514 or 17-214 or 15-214

Course Website: <http://euro.ecom.cmu.edu/program/courses/tcr17-803>  
(<http://euro.ecom.cmu.edu/program/courses/tcr17-803/>)

**17-401 Software Engineering for AI-Enabled Systems**

Fall: 12 units

New Course Need Description

Course Website: <https://ckaestne.github.io/seai/>

**17-402 AI and Emerging Economies**

Intermittent: 3 units

The course will cover some of the unique aspects of emerging economies as it relates to AI consumption and responsible AI development. The importance of local data and domain knowledge will be illustrated. Germane to budding technologists, entrepreneurs and policy influencers. Multi-disciplinary.

**17-405 Grand Challenges in AI: Past, Present and Future**

Intermittent: 3 units

Innovative, bold initiatives that capture the imagination of researchers and system builders are often required to spur a field of science or technology forward. A vision for the future of artificial intelligence was laid out by Turing Award winner and Moza Bint Nasser University Professor at CMU, Raj Reddy in his 1988 Presidential address to the Association for the Advancement of Artificial Intelligence. It is time to provide an accounting of the progress that has been made in the field, over the last three decades, toward the challenge goals. While some tasks such as the world-champion chess machine were accomplished in short order, many others, such as self-replicating systems, require more focus and breakthroughs for completion. A new set of challenges for the current decade is also proposed, spanning the health, wealth, and wisdom spheres. The above - plus commentary from half a dozen AI thought leaders - forms the basis of an article in the 2021 Spring Issue of AI Magazine.

**17-413 Software Engineering Practicum**

Spring: 12 units

This course is a project-based course in which students conduct a semester-long project for a real client in small teams. This is not a lecture-based course; after the first few weeks the course consists primarily of weekly team meetings with the course instructors, with teams making regular presentations on their software development process. Students will leave the course with a firsthand understanding of the software engineering realities that drive SE practices, will have concrete experience with these practices, and will have engaged in active reflection on this experience. After the course, students will have the teamwork, process, and product skills to be immediately competent in a software engineering organization, and will be able to evaluate the new processes and techniques they will encounter in the workplace.

**17-415 Software Engineering Reflection**

Fall: 6 units

This course is an opportunity to reflect on a software engineering experience you have had in industry. It is structured as a writers workshop, in which you will work with the instructor and other students to identify and flesh out a software engineering theme that is illustrated by your industry experience. You will prepare a 10-page report on this theme, comparable to a practitioner's report at a conference like ICSE or OOPSLA, and a 30-minute presentation to match. This course fulfills a requirement of the Software Engineering Minor program, but students in other programs may take the course if they meet the prerequisite industry experience and if space is available.

**17-422 Building User-Focused Sensing Systems**

Fall and Spring: 12 units

These days we are surrounded by sensing and computation. Smart devices, such as smartphones, smartwatches, are packed with sensors. While they are already very useful devices, we have only started to scratch the surface here. The aim of this class will be to introduce the students to building and understanding smart sensing devices. The course will include discussion into contribution of various fields, including human-computer interaction, embedded computing, computer vision, distributed systems, machine learning, signal processing, security, and privacy. We will discuss how these various disciplines are coming together to form an end-to-end system that generates useful and user-actionable data. We will take a hands-on approach towards building and evaluating these systems. The students will gain practical experience in developing sensing systems in different application domains, such as activity recognition, health sensing, gestural interaction, etc. You will learn about embedded systems and understand the advantages and limitations of different platforms. You will learn about sensors and how to interface them with the real world to be able to get useful and actionable data. You will learn how to build a network of sensors that can communicate with each other. You will also learn about storing the sensor data for visualization, analysis and presentation both locally and to the cloud. The course will be a combination of lectures, tutorials, class discussions, and demonstrations. Students will be evaluated based on 5 mini-projects/assignments, class participation, weekly reading summaries, and a final project. All hardware resources will be provided to the students and they will be given an option to take their final prototypes with them for the cost of the hardware components. Students should have reasonable programming experience and an interest in tinkering.

**17-428 Machine Learning and Sensing**

Fall: 12 units

Machine learning and sensors are at the core of most modern computing devices and technology. From Amazon Echo to Apple Watch to Google Photos to self-driving cars, making sense of the data coming from powerful but noisy sensors is the key challenge. The aim of the course will be to explore this intersection of sensors and machine learning, understand the inner workings on modern computing technologies, and design the future ones. We will cover data collection, signal processing, data processing, data visualization, feature engineering, machine learning tools, and some prototyping technologies. The course will focus on class discussions, hands-on demonstrations, and tutorials. Students will be evaluated on their class participation, multiple mini projects, and a final team project.

**17-437 Web Application Development**

Fall and Spring: 12 units

This course will introduce concepts in programming web application servers. We will study the fundamental architectural elements of programming web sites that produce content dynamically. The primary technology introduced will be the Django framework for Python, but we will cover related topics as necessary so that students can build significant applications. Such topics include: HTTP, HTML, CSS, Javascript, JSON, Design Patterns, Relational and Non-relational Databases, Object-Relation Mapping tools, Security, Web Services, Cloud Deployment, Internationalization, and Scalability and Performance Issues. Students must be comfortable programming in Python to register for this course. Students must provide their own computer hardware for this course. Please visit the Course URL for more information about the course.

Prerequisites: 18-613 or 15-513 or 18-213 or 14-513 or 17-214 or 15-214 or 17-514 or 15-213

Course Website: <https://www.cmu-webapps.org/static/index/syllabus.pdf>

**17-442 Software Management Theory**

Spring: 6 units

This course will look at software development from an organizational perspective and its designed for students who want to understand the relationship between business context, software development processes, knowledge creation, culture and organizational structure with the purpose of becoming change agents or manage the software development function at the department, business unit level or above. The course will also highlight the need to follow good work principles in order to avoid ethical failures as evidenced by recent affairs



**17-443 Quality Management**

Spring: 6 units

Managing software quality is a critical part of all software projects. Software engineers must consider quality during every phase of a project from inception to delivery and beyond. This class will introduce students to the managerial challenges of developing high quality software systems. The key learning objectives of this course include: 1. Define a quality management process in the context of a software project. 2. Understand the costs associated with achieving quality goals and not achieving them 3. Understand the tradeoffs required to implement quality assurance techniques. 4. Gain experience using collected quality metrics to inform project-level decisions.

Prerequisite: 17-323 Min. grade B

**17-445 Machine Learning in Production**

Fall and Spring: 12 units

The course takes a software engineering perspective on building software systems with a significant machine learning or AI component. It discusses how to take an idea and a model developed by a data scientist (e.g., scripts and Jupyter notebook) and deploy it as part of a scalable and maintainable system (e.g., mobile apps, web applications, IoT devices). Rather than focusing on modeling and learning itself, this course assumes a working relationship with a data scientist and focuses on issues of design, implementation, operation, and assurance and how those interact with the data scientist's modeling. This course is aimed at software engineers who want to understand the specific challenges of working with AI components and at data scientists who want to understand the challenges of getting a prototype model into production; it facilitates communication and collaboration between both roles.

Course Website: <https://ckaestne.github.io/seai/>**17-446 DevOps and Continuous Integration**

Spring: 6 units

TBD

Prerequisite: 17-336 Min. grade B

**17-450 Crafting Software**

Fall and Spring: 12 units

Do you use programming to solve problems in your field of study? Do you know enough to be dangerous, but wish you could be proud of your code? This course aims to provide students with sufficient knowledge and skills to use programming as part of their work. In this class, you will learn how to identify and find problems in your code. You will learn to read, parse, organize, and transform data. We will teach you to write code collaboratively and refine your programs so others can use them. The course will be a mixture of lecture and guided exercise with a recitation focused on hands on instruction. In this course, students are expected to have been exposed to some basic programming concepts, such as variables, if-statements, loops, and arrays. However, students are not expected to have extensive programming experience. This course is not appropriate for students that have completed more than two courses involving programming. We expect students in this class to have diverse backgrounds and experience. Some students will be self-taught, while others will have taken a programming course such as 02-201, 15-110, 95-898, or the library's Software Carpentry workshop. If you have questions about your background and the fit for this class, please don't hesitate to reach out to the instructors.

**17-480 API Design and Implementation**

Fall and Spring: 12 units

This class focuses on the design of programming interfaces, the APIs, within larger real-world software and ecosystems. We discuss the history and importance of APIs, and the principles behind designing good APIs. This includes study of specific examples of APIs, both good and bad, for inspiration and precaution. Students gain experience with the major steps of API design: gathering requirements, documenting, testing, implementing, refining, evolving, and reimplementing APIs. The principles taught are largely language-independent, though most examples are in Java or C. Students may be able to do assignments in other languages, within reason. Prerequisites: 15-213 or 15-214 or 17-214

**17-514 Principles of Software Construction: Objects, Design, and Concurrency**

Fall and Spring: 12 units

Software engineers today are less likely to design data structures and algorithms from scratch and more likely to build systems from library and framework components. In this course, students engage with concepts related to the construction of software systems at scale, building on their understanding of the basic building blocks of data structures, algorithms, and program and computer structures. The course covers technical topics in four areas: (1) concepts of design for complex systems, (2) object-oriented programming, (3) static and dynamic analysis for programs, and (4) concurrency. At the conclusion of this course, students will have substantial experience building medium-sized software systems in Java or JavaScript.

Prerequisites: (15-121 Min. grade C or 15-122 Min. grade C) and (15-151 Min. grade C or 21-128 Min. grade C or 21-127 Min. grade C)

Course Website: <https://www.cs.cmu.edu/~ckaestne/17214/f2021/>**17-536 Pervasive and Ubiquitous Computing**

Intermittent: 12 units

Note: Previously offered as 08530. The aim of the class will be to explore the area of Ubiquitous Computing (ubicomp) and allow students to work on a variety of small technology projects. Students will be exposed to the basics of building ubicomp systems, emerging new research topics, and advanced prototyping techniques. This course will focus more on class discussions and hands on demonstrations, while formal lectures will be conducted only as needed. Students will be evaluated on their class participation, reading summaries, and mini projects.

**17-537 Artificial Intelligence Methods for Social Good**

Spring: 9 units

Optimization: mathematical programming, robust optimization, influence maximization Game Theory and Mechanism Design: security games, human behavior modeling, auction and market equilibrium, citizen science Machine Learning: classification, clustering, probabilistic graphical models, deep learning Sequential Decision Making: Markov Decision Processes (MDPs), partially observable MDPs, online planning, reinforcement learning In addition to providing a deep understanding of these methods, the course will introduce which societal challenges they can tackle and how, in the areas of (i) healthcare, (ii) social welfare, (iii) security and privacy, (iv) environmental sustainability. The course will also cover special topics such as AI and Ethics and AI and Humans. The course content is designed to not have too much overlap with other AI courses offered at CMU. Although the course is listed within SCS, it should be of interest to students in several other departments, including ECE, EPP and SDS. The students in this 9-unit course are expected to have taken at least three mathematics courses covering linear algebra, calculus, and probability. The students will work in groups on a systematic literature review or a project exploring the possibility of applying existing AI tools to a societal problem, with a survey paper or technical report and presentation delivered at the end of the semester.

**17-562 Law of Computer Technology**

Fall: 9 units

A survey of how legislatures and courts cope with rapidly advancing computer technologies and how scientific information is presented to, and evaluated by, civil authorities. The course is also an introduction to the legal process generally and the interaction between the legal system and technology organizations. Topics include: patents, copyrights in a networked world, law of the Internet, free speech, data security, technology regulation, international law, and trans-border crime. Open to juniors, seniors and graduate students in any school. Open to sophomores by permission of the instructor. Prerequisites: none.

**17-599 Advanced Topics in Machine Learning and Game Theory**

Fall: 12 units

This course is designed to be a graduate-level course covering the topics at the intersection of machine learning and game theory. Recent years have witnessed significant advances in machine learning and their successes in detection, prediction, and decision-making problems. However, in many application domains, ranging from auction and ads bidding, to entertainment games such as Go and Poker, to autonomous driving and traffic routing, to the intelligent warehouse, to home assistants and the Internet of Things, there is more than one agent interacting with each other. Game theory provides a framework for analyzing the strategic interaction between multiple agents and can complement machine learning when dealing with challenges in these domains. Therefore, in the course, we will introduce how to integrate machine learning and game theory to tackle challenges in multi-agent systems. The course will multiple topics as listed below

Course Website: <https://feifang.info/advanced-topics-in-machine-learning-and-game-theory-fall-2021/>

**17-612 Business and Marketing Strategy**

Fall: 6 units

This course prepares technically minded students to understand and use essential business concepts in their careers. With this competency, students will be able to make use of, contribute to, and influence the business and marketing decisions that affect engineering and technology decisions, quality, and performance. Students will be better equipped to make business arguments for supporting their product, technology and engineering ideas in their future, too. This 6-unit course emphasizes learning-by-doing to achieve the learning objectives. Each student works on a course-long project where they'll conceive and plan a new technology product idea for an existing company. They'll make product, technological, pricing, marketing, and sales decisions to achieve success and are in support of the company's strategy and business decisions. Their work will culminate in building a compelling business case, with financial projections, to persuade the company executives to invest in the product idea.

Course Website: <https://mse.isri.cmu.edu/>**17-619 Introduction to Real-Time Software and Systems**

Intermittent: 12 units

Introduction to Real-Time Software and Systems presents an overview of time as it relates engineering complex systems. Any system that responds at the pace of relevant events has real-time constraints whether the timescale is short, like the flight controls for an aircraft, or longer, like the flight reservation system for an airline. Fundamental concepts, terminology, and issues of real-time systems are introduced in this course. The focus is on software solutions to real-time problems-solutions that must be both correct and timely. Software development is examined with emphasis on real-time issues during each phase of the software lifecycle. Real-time requirements analysis, architecting real-time systems, designing and modeling system timing, and implementation and testing strategies are studied. Modeling techniques using UML 2.0 are applied. Particular emphasis is placed on real-time scheduling to achieve desired timing, reliability, and robustness. Languages and operating systems for real-time computing, and real-time problems in concurrent and distributed systems are explored. This course provides a comprehensive view of real-time systems with theory, techniques and methods for the practitioner. After successfully completing this course, the student will be able to identify constraints and understand real-time issues in system development, and propose approaches to typical real-time problems. The aim of this course is to motivate and prepare students to pursue more in-depth study of specific problems in real-time computing and systems development. REQUIREMENT: Proficiency with a high-level programming language such as C or Ada and basic concepts of computing systems. Familiarity with software engineering concepts and system development lifecycle.

**17-621 Computer Simulation of Complex Socio-Technical Systems**

Intermittent: 12 units

How likely is an intervention like social distancing to save lives? Will a law legislating sanctions against social media platforms that spread disinformation stop the spread? We live and work in complex adaptive and evolving socio-technical systems where questions such as these arise constantly. Questions such as these are often only addressable through computational modeling, i.e., through simulation. Simulation models are a critical method for understanding how to adaptation and learning will change the status-quo. Computational modeling can be used to help analyze, reason about, predict the behavior of, and possibly control complex human systems of "networked" agents. Using simulation it is possible to advance theory, test policies before enacting them, and think through non-linear social effects.

Course Website: <http://www.casos.cs.cmu.edu/courses/>**17-624 Advanced Formal Methods**

Fall: 6 units

This course builds on the introductory Models class to cover more advanced techniques for modeling and reasoning about complex software systems. Concepts introduced in this course include abstraction and refinement, declarative specifications, advanced temporal logics, and probabilistic modeling. The course will also explore applications of modeling and automated reasoning techniques in various domains, such as security, distributed computing, and cyber-physical systems. After completing this course, students will: 1. Understand how to specify and reason about operations over complex system structures, 2. Understand relationships between software artifacts at different levels of abstraction; 3. Be able to model and reason about systems with uncertainty and stochastic behaviors; and 4. Understand potential applications of modeling techniques to practical software engineering problems. Prerequisites: Completion of Mini 1: Models of Software Systems. Sections D, PP and G are NOT available for on-campus students. Admission to the class is by approval from the instructor: If you are not a software engineering master's student, send email to [garlan@cs.cmu.edu](mailto:garlan@cs.cmu.edu) for permission to enroll. The email should briefly describe your background, whether you have taken a course with similar materials as in Mini 1, and why you would like to take the course. The course must be taken for a letter grade (not pass/fail). This is a graduate level course.

Prerequisite: 17-614 Min. grade B

**17-626 Requirements for Information Systems**

Fall: 6 units

Software engineering requires understanding the problem, before identifying solutions. In this course, students study ways to elicit and analyze problem statements using scenarios, use cases and mockups.

**17-627 Requirements for Embedded Systems**

Fall: 6 units

Software engineering requires understanding the problem, before identifying solutions. In this course, students study ways to elicit and analyze problem statements for real-time systems along multiple dimensions, including concurrency, dependability and safety.

Prerequisite: 17-614 Min. grade B

**17-634 Applied Machine Learning**

Spring: 6 units

Autonomous and intelligent systems increasingly rely on automated decision making based on statistical models used for classification or prediction. The practical application of machine learning requires understanding the underlying theoretical assumptions behind a wide variety of statistical models, how to analyze the performance of such models, and how to integrate models into data processing pipelines. This course introduces students to supervised and unsupervised machine learning in the context of software engineering, including the analysis of natural language in bug reports and mobile app reviews. Techniques covered include latent Dirichlet allocation, TF/IDF, naive Bayes, linear regression, decision trees, and random forests.

**17-640 IoT, Big Data, and ML: A Hands-on Approach**

Intermittent: 12 units

This course is designed to teach IoT concepts, big data, and machine learning techniques using a hands-on approach. An IoT system simulating an order fulfillment process is central to the hands-on learning of the concepts and techniques. Students will work in 4-5 person teams to enable the system and implement the requirements. In doing so, they will incorporate sound design principles of software engineering acquired in lectures. Students will capture the data generated during the execution of the system as it fulfills orders that are received from a front-end system developed by the students. Students will be expected to prepare, process, and model the data for statistical analysis applying techniques taught in class. They will then visualize, analyze and interpret the results, and implement improvements to obtain a 360-degree experience of a business application using the automated system. This course will provide insight into the ways in which business enterprises think about leveraging technology and software in the management of their production operations. The course prepares students for professional opportunities requiring such skills allowing them to identify use cases that facilitate innovation and promote competitiveness.

**17-644 Applied Deep Learning**

Spring: 6 units

Deep neural networks have made in-roads in virtually every industry, propelled by exponential increases in compute power and fundamental progress in modeling. Knowledge of these models is fast becoming a key asset for software engineers, as current systems are quickly starting to include many neural components, and the practice of software engineering itself is starting to benefit from neural program assistance (incl. automated bug finding, translation between programming languages). This course equips the next generation of software engineers with knowledge of neural models, the software engineering challenges involved in using these, and hands-on experience with their applications. It teaches both a rich vocabulary of general, essential concepts (including architectures), and recent work on applications of these models, aimed primarily at applications for and in software engineering itself. The course includes a group project aimed at constructing a neural solution for an existing application that will be used to teach the various stages (and their pitfalls) of building and deploying deep learners.

**17-651 Models of Software Systems**

Fall: 12 units

Scientific foundations for software engineering depend on the use of precise, abstract models for describing and reasoning about properties of software systems. This course considers a variety of standard models for representing sequential and concurrent systems, such as state machines, algebras, and traces. It shows how different logics can be used to specify properties of systems, such as functional correctness, deadlock freedom, and internal consistency. Concepts such as compositionality, abstraction, invariants, non-determinism, and inductive definitions are recurrent themes throughout the course. After completing this course, students will: 1. Understand the strengths and weaknesses of certain models and logics including state machines, algebraic and process models, and temporal logic; 2. Be able to select and describe appropriate abstract formal models for certain classes of systems, describe abstraction relations between different levels of description, and reason about the correctness of refinements; 3. Be able to prove elementary properties about systems described by the models introduced in the course; and 4. Understand some of the strengths and weakness of formal automated reasoning tools. Prerequisites: Undergraduate discrete math including first-order logic, sets, functions, relations, and simple proof techniques such as induction. Sections D, PP and G are NOT available for on-campus students. Admission to the class is by approval from the instructor: If you are not MSE/MSIT-SE/MITS, send email to garlan@cs.cmu.edu for permission to enroll. The email should briefly describe your background, whether you have taken an undergraduate discrete math course, and why you would like to take the course. The course must be taken for a letter grade (not pass/fail). This is a graduate level course.

**17-654 Analysis of Software Artifacts**

Spring: 12 units

Analysis is the systematic examination of an artifact to determine its properties. This course will focus on analysis of software artifacts and #8212;primarily code, but also including analysis of designs, architectures, and test suites. We will focus on functional properties, but also cover non-functional properties like performance and security. In order to illustrate core analysis concepts in some depth, the course will center on static program analysis; however, the course will also include a breadth of techniques such as testing, model checking, theorem proving, dynamic analysis, and type systems. The course will balance theoretical discussions with lab exercises in which students will apply the ideas they are learning to real artifacts. After completing this course, students will: \* know what kinds of analyses are available and how to use them \* understand their scope and power, when they can be applied and what conclusions can be drawn from their results \* have a grasp of fundamental notions sufficient to evaluate new kinds of analysis when they are developed \* have some experience selecting and writing analyses for a real piece of software, applying them and interpreting the results Ph.D. students taking the 17-754 version of the course will gain a broad overview of the analysis research literature and in-depth knowledge of a particular sub-area through a course project. Requirement: A recent discrete math course and programming experience. Strongly Recommended: Models of SW Development course (17-651) before taking this course. This course is for letter grade only (no pass/fail grades). This is a graduate course. Only undergrad SE minors may take this course with the instructor's permission. Please note: Students outside of the software engineering department may take this course but students of the MSE programs will have first priority.

**17-663 Programming Language Pragmatics**

Fall: 12 units

This course provides a broad and pragmatic foundation in the most basic tool of the programmer: programming languages. It starts with the fundamentals of syntax, parsing, and binding, the core structural concepts in programming languages. The course will then cover program semantics and type systems, and students will learn to relate them with a type soundness theorem. Finally, a coverage of intermediate optimization and code generation offers the opportunity to discuss both producing efficient code and reasoning about the correctness of program transformations. Assignments involve a combination of tool-assisted formal reasoning and proofs about programming languages, and implementing these language constructs in a compiler. This course fulfills the Logic and amp; Languages constrained elective of the B.S. in Computer Science. Students with substantial math and programming experience who have not satisfied the specific prerequisites can contact the instructor for permission to enroll. Prerequisites: 15-251 Min. grade C and (15-150 Min. grade C or 21-228 Min. grade C)

Course Website: <http://www.cs.cmu.edu/~aldrich/courses/17-363/>**17-670 Virtual Machines and Managed Runtimes**

Fall: 12 units

Traditional compiler and programming language courses focus on language implementations that generate code statically (offline) for a target machine. Yet in today's landscape, many programming languages run on virtual machines where either a virtual instruction set architecture (bytecode) or the source code of the program is translated (and optimized) dynamically for an underlying machine. Such languages come with additional challenges beyond traditional compilation as taught in courses, including efficient representation of high-level language constructs and automatic memory management in the form of garbage collection. Together, the components of the virtual machine and services form what is known as a managed runtime system. This course focuses on implementation techniques for managed runtime systems. Students will learn the basics of virtual machine implementation and build working prototypes to run actual programs. Prior knowledge of compilers and some familiarity with programming language implementation is required.

Prerequisites: 15-611 or 15-411

**17-685 Dynamic Network Analysis**

Spring: 12 units

Who knows who? Who knows what? Who is influential? What is the social network, the knowledge network, the activity network? How do ideas, products and amp; diseases propagate through groups and impact these networks? Does social media change the way these networks operate? Questions such as these and amp; millions of others require a network perspective and an understanding of how ties among people, ideas, things, and amp; locations connect, constrain and amp; enable activity. In the past decade there has been an explosion of interest in network science moving from the work on social networks and graph theory to statistical and computer simulation models. Network analysis, like statistics, now plays a role in most empirical fields. Network science is a broad and multi-disciplinary field. In this class, students will gain an appreciation of the history of the field, the difference between social networks and social media, the difference graph-based metrics for network analysis and graphical models, the use of traditional and high dimensional network models, and the advances in this field. Applications and issues discussed will include: social media analytics, semantic networks, task networks, organizational design and teams, machine learning and network analysis, generative models, terrorism and crime, health, and fake news. Methods for network data collection, analysis, visualization, and interpretation are covered. Students produce original research in which network data is analyzed using the methods covered in the class

Course Website: <http://www.casos.cs.cmu.edu/courses/>

**17-691 Machine Learning in Practice**

Spring: 6 units

As Machine Learning and Artificial Intelligence methods have become common place in both academic and industry environments the majority of resources have focused on methods and techniques for applications. However, there are many considerations that must be addressed when deploying such techniques into practice (or production). The purpose of this course is to cover topics relevant to building a machine learning systems deployed into operations. Such systems have technical requirements including data management, model development, and deployment. However, business/organizational impacts must also be considered. Machine learning systems can be expensive to produce and operate. Students will learn about trade-offs in design, implementation, and expected value. After completing this course, students will: 1. Have the ability to deploy produces with machine learning and AI components; 2. Understand how to implement data pipelines and data engineering systems; 3. Calculate the approximate value provided by a machine learning system to an organization; 4. Understand how to continually assess the value and quality of a deployed machine learning system. Prerequisites: understanding of basic machine learning concepts (i.e. supervised/unsupervised learning). This is a graduate level course. Prerequisite: 17-634 Min. grade B

**17-692 Product Management Essentials**

Spring: 6 units

This course prepares technically minded students to understand and use the essential product management concepts and practices at the earliest stage of every new product innovation idea. With this understanding and skills, students will be able to identify and define a worthwhile customer problem to solve, conceive of an innovative, differentiated product solution, design and quantify a compelling customer value proposition and set a strategy for continued innovation and growth. This 6-unit course emphasizes learning-by-doing to achieve the learning objectives. You will work on a hands-on, course-long project focused on a problem space selected by the student and approved by the instructor for course fit. PRIORITY is given to students in the Master of Software Engineering (MSE), MS in Engineering and amp; Technology Innovation Management and Master of Information Systems Management degree programs. Make sure you register for the section according to the criteria listed in "Extra Time Commitments."

**17-702 Current Topics in Privacy Seminar**

Fall and Spring: 3 units

In this seminar course students will discuss recent papers and current public policy issues related to privacy. Privacy professionals from industry, government, and non-profits will deliver several guest lectures each semester.

**17-712 Fantastic Bugs and How to Find Them**

Spring: 12 units

This advanced course studies the nature of software bugs and security vulnerabilities arising in complex application domains and surveys specialized program analysis + automated testing techniques for identifying such issues proactively. The course will take a tour of various domains such as mobile systems, databases, web browsers, distributed and networked systems, autonomous vehicles, and smart contracts. For each domain, the class will review case studies of high-impact software bugs that have manifested in production and will then discuss state-of-the-art research techniques that aim to uncover such bugs automatically. Apart from the literature review, students will engage significantly with software system design and engineering via hands-on assignments and a semester-long project involving real-world applications and analysis tools for one or more domains. Students completing this course will be able to (a) identify practical challenges of applying well-known program analysis and testing techniques to complex application domains, (b) formulate and leverage domain-specific assumptions for making analysis techniques tractable in a specialized setting, and (c) build practical tools for improving software quality in large-scale systems. The course assumes that students have some background in reasoning about software quality, system security, and/or working with program representations. The course builds upon mathematical principles introduced in foundational classes on program analysis, verification, or compiler design, as well as system design principles encountered in introductory security or software engineering courses. Please check the course website for classes that qualify as sufficient prerequisites, or contact the instructor to discuss your background.

Course Website: <https://cmu-fantastic-bugs.github.io/>**17-731 Foundations of Privacy**

Fall: 12 units

Privacy is a significant concern in modern society. Individuals share personal information with many different organizations - healthcare, financial and educational institutions, the census bureau, Web services providers and online social networks - often in electronic form. Privacy violations occur when such personal information is inappropriately collected, shared or used. We will study privacy in a few settings where rigorous definitions and enforcement mechanisms are being developed - statistical disclosure limitation (as may be used by the census bureau in releasing statistics), semantics and logical specification of privacy policies that constrain information flow and use (e.g., by privacy regulations such as the HIPAA Privacy Rule and the Gramm-Leach-Bliley Act), principled audit and accountability mechanisms for enforcing privacy policies, anonymous communication protocols - and other settings in which privacy concerns have prompted much research, such as in social networks, location privacy and Web privacy (in particular, online tracking and amp; targeted advertising).

**17-733 Privacy Policy, Law, and Technology**

Fall: 12 units

NOTE: Previously offered as 08-733. This course focuses on policy issues related to privacy from the perspectives of governments, organizations, and individuals. We will begin with a historical and philosophical study of privacy and then explore recent public policy issues. We will examine the privacy protections provided by laws and regulations, as well as the way technology can be used to protect privacy. We will emphasize technology-related privacy concerns and mitigation, for example: social networks, smartphones, behavioral advertising (and tools to prevent targeted advertising and tracking), anonymous communication systems, big data, and drones. This is part of a series of courses offered as part of the MSIT-Privacy Engineering masters program. These courses may be taken in any order or simultaneously. Foundations of Privacy (Fall semester) offers more in-depth coverage of technologies and algorithms used to reason about and protect privacy. Engineering Privacy in Software (Spring semester) focuses on the methods and tools needed to design systems for privacy. This course is intended primarily for graduate students and advanced undergraduate students with some technical background. Programming skills are not required. 8-733, 19-608, and 95-818 are 12-unit courses for PhD students. Students enrolled under these course numbers will have extra assignments and will be expected to do a project suitable for publication. 8-533 is a 9-unit course for undergraduate students. Masters students may register for any of the course numbers permitted by their program. This course will include a lot of reading, writing, and class discussion. Students will be able to tailor their assignments to their skills and interests. However, all students will be expected to do some writing and some technical work.

**17-735 Engineering Privacy in Software**

Spring: 12 units

Privacy harms that involve personal data can often be traced back to software design failures, which can be prevented through sound engineering practices. In this course, students will learn how to identify privacy threats due to surveillance activities that enhance modern information systems, including location tracking, behavioral profiling, recommender systems, and social networking. Students will learn to analyze systems to identify the core operating principles and technical means that introduce privacy threats, and they will learn to evaluate and mitigate privacy risks to individuals by investigating system design alternatives. Strategies to mitigating privacy risk will be based on emerging standards and reliable privacy preference data. Students will have the opportunity to study web-, mobile- and cyber-physical systems across a range of domains, including advertising, healthcare, law enforcement and social networking. In addition, students will know how, and when, to interface with relevant stakeholders, including legal, marketing and other developers in order to align software design with privacy policy and law.

**17-756 Computational Social Science Research Design and Data Analytics**

Spring: 12 units

This course surveys how digital trace data of human activity online, combined with careful research design and data analytics, have led to surprising discoveries and development of novel theoretical explanations in the field of computational social science (CSS). The course has three aims. (a) It is intended to stimulate new ways of formulating research questions on pressing/emerging societal issues of interest (e.g., political polarization, gender representation in open-source software development, decentralized governance based on blockchain technology) given the possibilities afforded by a wide array of digital trace data (e.g., social media, open-source software collaboration, cryptocurrency transactions, satellite imagery). This course will (b) discuss exemplary works in CSS that made novel discoveries or made theoretical and/or methodological breakthroughs and (c) cover practical considerations that commonly arise in a CSS research project cycle. These issues may include, but are not limited to, data collection (sampling methods and bias), repurposing existing datasets that were collected in a different research context, constructing metrics based on theoretical insights, analytic approaches adequate for a given research question (statistical modeling, network analysis, online experiments, simulation-based theory development), big data processing and analytic tools (distributed data processing frameworks, visualization), ethical considerations, and data sharing. In the end, this course will help participants develop "good taste" for theoretically insightful, methodologically creative, and well thought out research, while learning to identify potential pitfalls that could arise in their own research.

**17-801 Dynamic Network Analysis**

Spring: 12 units

Who knows who? Who is influential? What is the social network, the knowledge network, the activity network? How do ideas, products and amp; diseases propagate through groups and impact these networks? Does social media change the way these networks operate? Questions such as these and amp; millions of others require a network perspective and an understanding of how ties among people, ideas, things, and amp; locations connect, constrain and amp; enable activity. In the past decade there has been an explosion of interest in network science moving from the work on social networks and graph theory to statistical and computer simulation models. Network analysis, like statistics, now plays a role in most empirical fields. Network science is a broad and multi-disciplinary field. In this class, students will gain an appreciation of the history of the field, the difference between social networks and social media, the difference graph-based metrics for network analysis and graphical models, the use of traditional and high dimensional network models, and the advances in this field. Applications and issues discussed will include: social media analytics, semantic networks, task networks, organizational design and teams, machine learning and network analysis, generative models, terrorism and crime, health, and fake news. Methods for network data collection, analysis, visualization, and interpretation are covered. Students produce original research in which network data is analyzed using the methods covered in the class.

Course Website: <http://www.casos.cs.cmu.edu/courses/>**17-821 Computer Simulation of Complex Socio-Technical Systems**

Spring: 12 units

How likely is an intervention like social distancing to save lives? Will a law legislating sanctions against social media platforms that spread disinformation stop the spread? We live and work in complex adaptive and evolving socio-technical systems where questions such as these arise constantly. Questions such as these are often only addressable through computational modeling, i.e., through simulation. Simulation models are a critical method for understanding how to adaptation and learning will change the status-quo. Computational modeling can be used to help analyze, reason about, predict the behavior of, and possibly control complex human systems of "networked" agents. Using simulation it is possible to advance theory, test policies before enacting them, and think through non-linear social effects.

Course Website: <http://www.casos.cs.cmu.edu/courses/>**17-880 Algorithms for Private Data Analysis**

Spring: 12 units

We study the following question in this course: How do we perform useful analysis on a data set that contains sensitive information about individuals without compromising the privacy of those individuals? To study this question, we will introduce differential privacy, a framework of designing data analysis algorithms with strong, meaningful, and mathematically provable privacy guarantees. We will survey a set of algorithmic tools that allow us to privately perform a wide range of statistical analyses. Of course, privacy does not come for free, and we will also study some of the fundamental limitations imposed by the requirement of differential privacy. Through the discussion of these results, we will also demonstrate some of the most novel and surprising connections between differential privacy and other areas of theoretical computer science, including machine learning theory, cryptography, convex geometry, and game theory.