Department of Computer Science

Professor: *Hildreth, Shull (Chair)*

Associate Professor: *Metaxas, Tjaden, Turbak*

Assistant Professor: *Shaer*

Hess Fellow: *TBA*

Lecturer: *Anderson*

Instructor in Computer Science Laboratory: *Herbst, Kakavouli, Lee*

Computer Science encompasses the systematic study of computing systems and computation. It is continually evolving and expanding, making it an exciting field of study. All of the traditional areas in computer science as well as newer directions are represented in our faculty's expertise (including algorithms, programming languages, data structures, artificial intelligence, human-computer interaction, databases, computer architecture, networks, security, vision, graphics, parallel computing, robotics, bioinformatics, Web information retrieval, multimedia), allowing us to offer a large variety of courses and substantial research opportunities for students.

The computer science department offers four introductory computer science courses: CS 110, 111, 112, and 114. For advice in making a choice consult "Choosing an Introductory CS Course" online at http://cs.wellesley.edu/~cs/Curriculum/intro.html.

Goals for the Major

We expect every computer science major upon graduation to possess the following knowledge and competencies:

A firm foundation in fundamental areas of computer science including an understanding of the connections between computer science and other disciplines;

A solid preparation for either graduate work or direct entry into the computing profession;

A commitment to problem-solving, principles, theory, and analysis, in keeping with the tenets of a liberal arts setting;

Strong written and oral communication skills, including the ability to work on a team-based project and to solve problems in a team-based setting;

The ability to read a paper from a general computer science journal and to be able to understand parts of more technical papers in specialized journals and conferences.

CS 110 Computer Science and the Internet

Anderson, Shaer

This course explains the basics of how the Internet works and how to build a Web site. Topics include client-server architecture, structuring Web pages with HTML, CSS, and JavaScript, the representation of colors, images, and sound on the computer, encryption, cookies, and CGI forms. We also discuss accessibility, copyright, intellectual property, and critical thinking in the context of the Internet. The required project models most phases of the standard software lifecycle. Students are introduced to programming by building an interactive Web site using JavaScript. *Students are required to attend an additional 70-minute discussion section each week. Students considering additional computer science courses should take 111, not 110.*

Prerequisite: None. No prior background with computers is expected.
Distribution: Mathematical Modeling
Semester: Fall, Spring Unit: 1.0

CS 111 Computer Programming and Problem Solving

Shaer, Tjaden, Turbak

An introduction to problem solving through computer programming. Using the Java programming language, students learn how to read, modify, design, debug, and test algorithms that solve problems. Programming concepts include control structures, data structures, abstraction, recursion, modularity, and object-oriented design. Students explore these concepts in the context of interactive programs involving graphics and user interfaces. *Students are required to attend an additional two-hour laboratory section each week. Required for students who wish to major or minor in computer science or elect more advanced courses in the field.*

Prerequisite: None. No prior background with computers is expected.
Distribution: Mathematical Modeling. Does not satisfy the laboratory requirement.
Semester: Fall, Spring Unit: 1.0

CS 112 Computation for the Sciences

Hildreth

An introduction to computer programming that provides the tools necessary for students to use computers effectively in scientific work, including physical sciences, biological sciences, medicine, mathematics, psychology and economics. Students learn to write software to solve problems, visualize and analyze data, perform computer simulations, and implement and test computational models that arise in a wide range of scientific disciplines. The course introduces MATLAB, an extensive and widely used technical computing environment with advanced graphics, visualization and analysis tools, and a rich high-level programming language. *Students are required to attend an additional two-hour laboratory section each week.*

Prerequisite: None. No prior background with computers is expected.
Distribution: Mathematical Modeling. Does not satisfy the laboratory requirement.
Semester: Spring Unit: 1.0

CS 114 The Socio-Technological Web

Metaxas

As more and more people use the technologies and services made available from computer science, online environments like Facebook, Second Life, MySpace, Wikipedia, blogs, and open source development communities, have been flourishing. It is becoming clear that problems existing in our real world transfer and become amplified in the virtual world created by our interconnectivity. This course will start by studying the structure of the traditional Web and its recent successor, the Social Web, and will focus on issues of virtual identity, personal and group privacy, trust evaluation and propagation, online security, critical thinking, online propaganda, googlearchy, fraud and manipulation, restricted resources, class differences, self-perception, and decision-making. *Students are required to attend an additional 70-minute discussion section each week.*

Prerequisite: None
Distribution: Mathematical Modeling
Semester: Fall Unit: 1.0

CS 116/PHYS 116 Robotic Design Studio

Berg (Physics)

This first-year seminar introduces liberal arts students to the essence of engineering while designing and assembling robots out of LEGO® parts, sensors, motors, and tiny computers. Fundamental robotics skills are leamed in the context of studying and modifying a simple robot known as SciBorg. Then, working in small teams, students design and build their own robots for display at a robot exhibition. These projects tie together aspects of a surprisingly wide range of disciplines, including computer science, physics, engineering, and art. *Students may register for either CS 116 or PHYS 116 and credit will be granted accordingly.*

Prerequisite: None. Open to first-year students only.
Distribution: Natural and Physical Science
Semester: Fall Unit: 1.0

CS 215 Multimedia Design and Programming

Metaxas

The purpose of this course is to give students a broad foundation in issues related to creating multimedia and hypermedia applications. Topics to be covered include history and philosophy of hypermedia; principles of human-computer interaction; multimedia programming; optimizing for CD-ROMs and the World Wide Web; digital representation and editing of media (audio, graphics, video); media compression and transmission; and delivery of multimedia applications. *Students are required to attend an additional 70 minute laboratory section each week.*

Prerequisite: At least 111 (preferred) or 110 is required. At least one of ARTS 105, ARTS 108/CAMS 138, or ARTS 109/CAMS 139 is recommended.
Distribution: Mathematical Modeling. Does not satisfy the laboratory requirement.
Semester: Spring Unit: 1.0

CS 220 Human-Computer Interaction

Shaer

Human-Computer Interaction is one of the areas that have transformed the way we use computers in the last 30 years. Topics include methodology for designing and testing user interfaces, interaction styles (command line, menus, graphical user interfaces, virtual reality, tangible user interfaces), interaction techniques (including use of voice, gesture, eye movements), design guidelines, and user interface software tools. Students will design a user interface, program a prototype, and test the results for usability. *Students are required to attend an additional 70-minute discussion section each week.*

Prerequisite: One of: 110, 111, 112, 114
Distribution: Mathematical Modeling
Semester: Spring Unit: 1.0

CS 230 Data Structures

Metaxas, Tjaden

An introduction to techniques and building blocks for organizing large programs. Topics include: modules, abstract data types, recursion, algorithmic efficiency, and the use and implementation of standard data structures and algorithms, such as lists, trees, graphs, stacks, queues, priority queues, tables, sorting, and searching. Students become familiar with these concepts through weekly programming assignments using the Java programming language. *Students are required to attend an additional two-hour laboratory section each week.*

Prerequisite: 111 or permission of the instructor. Students who received a grade of C+ or lower in 111 must contact the instructor before enrolling.
Distribution: Mathematical Modeling. Does not satisfy the laboratory requirement.
Semester: Fall, Spring Unit: 1.0

CS 231 Fundamental Algorithms

Tjaden

An introduction to the design and analysis of fundamental algorithms. General techniques covered: divide-and-conquer algorithms, dynamic programming, greediness, probabilistic algorithms. Topics include: sorting, searching, graph algorithms, compression, cryptography, computational geometry, and NP-completeness.

Prerequisite: 230 and either MATH 225 or permission of the instructor.
Distribution: Mathematical Modeling
Semester: Spring Unit: 1.0

CS 232 Artificial Intelligence

Hildreth

An introduction to artificial intelligence (AI), the design of computer systems that possess and acquire knowledge and can reason with that knowledge. Topics include knowledge representation, problem solving and search, planning, vision, language comprehension and production, learning, common sense reasoning, and expert systems. To attain a realistic and concrete understanding of these problems, CommonLisp, an AI programming language, will be taught and used to implement the algorithms of the course. *Alternate year course.*

Prerequisite: 230 or by permission of the instructor.
Distribution: Mathematical Modeling
Semester: Fall Unit: 1.0

CS 235 Languages and Automata

Turbak

This course offers an introduction to the concepts of languages and automata. Topics include languages, regular expressions, finite automata, grammars, pushdown automata, and Turing machines. The first half of the semester covers the Chomsky hierarchy of languages and their associated computational models. The second half of the semester focuses on decidability issues and unsolvable problems and the course closes with a brief introduction to complexity theory. The course includes a programming component investigating the application of automata theory to the scanning and parsing of programming languages.

Prerequisite: 230 and either MATH 225 or permission of the instructor.
Distribution: Mathematical Modeling
Semester: Fall Unit: 1.0

CS 240 Introduction to Machine Organization with Laboratory

Shull

This course is intended to demystify the computer (open up the “black box”) and teach how information at the highest level is processed and ultimately executed by the underlying circuitry. To this end, the course provides an introduction to machine organization and assembly language programming. Specific topics include the fundamentals of computer organization (introduction to numeric representation, Boolean logic, digital logic and all associated technology), a basic data path implementation, assembly language programming, how to assess and understand the performance of a computer, and brief overviews of assemblers, compilers and operating systems. *Students are required to attend one three-hour laboratory weekly.*

Prerequisite: 111
Distribution: Mathematical Modeling. This course satisfies the laboratory requirement.
Semester: Fall Unit: 1.25

CS 242 Computer Networks

Turbak

NOT OFFERED IN 2010-11. OFFERED IN 2011-12. A systems-oriented approach to data networks, including a theoretical discussion of common networking problems and an examination of modern networks and protocols. Topics include point-to-point links, packet switching, internet protocols, end-to-end protocols, congestion control, and security. Projects may include client-server applications and network measurement tools. *Alternate year course.*

Prerequisite: 230 or permission of the instructor.
Distribution: Mathematical Modeling
Semester: N/O. Offered in 2011-12 Unit: 1.0

CS 249 Topics in Computer Science

NOT OFFERED IN 2010-11.

Prerequisite: 110 or 111.
Distribution: Mathematical Modeling
Semester: N/O Unit: 1.0

CS 250 Research or Individual Study

Prerequisite: 230 or permission of the instructor.
Distribution: None
Semester: Fall, Spring Unit: 1.0

CS 250H Research or Individual Study

Prerequisite: 230 or permission of the instructor.
Distribution: None
Semester: Fall, Spring Unit: 0.5

CS 251 Theory of Programming Languages

Shull

This course offers an introduction to the dimensions of modern programming languages. Covers major programming paradigms: function-oriented, imperative, object-oriented, and logic-oriented. Dimensions include syntax, naming, state, data, control, concurrency, nondeterminism, and types. These dimensions are explored via mini-language interpreters written in OCaml, Scheme, and Haskell that students experiment with and extend.

Prerequisite: 230 and either 235 or permission of the instructor.
Distribution: Mathematical Modeling
Semester: Spring Unit: 1.0

CS 303/BISC 303 Bioinformatics

Tjaden

NOT OFFERED IN 2010-11. A multidisciplinary course exploring the origins, present and future applications, and challenges of the intersection of biological and computer sciences. The field of bioinformatics, generated in response to the era of genomics, encompasses all aspects of biological data acquisition, storage, processing, analysis and interpretation with a view to generating in silico models of cellular function. *Students may register for either CS 303 or BISC 303 and credit will be granted accordingly.*

Prerequisites: 231, BISC 219 or BISC 220. Not open to students who have taken CS 313.
Distribution: Natural and Physical Science
Semester: N/O Unit: 1.0

CS 304 Databases with Web Interfaces

Anderson

A study of the three-layer architecture commonly used for Web-based applications such as e-commerce sites. We will learn to model and design databases uses entity-relationship diagrams, and the Standard Query Language (SQL) for managing databases. We will learn PHP, CGI/Python, and Java Servlets, which are three important technologies for Web-based architectures. We will also discuss performance, reliability and security issues. Finally, we will create dynamic Web sites driven by database entries.

Prerequisite: 230
Distribution: Mathematical Modeling
Semester: Spring Unit: 1.0

CS 307 Computer Graphics

Anderson

NOT OFFERED IN 2010-11. OFFERED IN 2011-12. A survey of topics in computer graphics with an emphasis on fundamental techniques. Topics include: graphics hardware, fundamentals of three-dimensional graphics including modeling, projection, coordinate transformation, synthetic camera specification, color, lighting, shading, hidden surface removal, animation, and texture-mapping. We also cover the mathematical representation and programming specification of lines, planes, curves, and surfaces.

Prerequisite: 230
Distribution: Mathematical Modeling
Semester: N/O. Offered in 2011-12. Unit: 1.0

CS 310 Foundations of Cryptology

Shull

When is a cryptographic system secure and how will we ever know? This course introduces the computational models and theory computer scientists use to address these issues. Topics include one-way functions, trapdoor functions, probabilistic complexity classes, pseudorandom generators, interactive proof systems, zero-knowledge proofs, and the application of these theories to modern cryptology.

Prerequisite: 231 or 235 or permission of the instructor.
Distribution: Mathematical Modeling
Semester: Spring Unit: 1.0

CS 313 Computational Biology

Tjaden

NOT OFFERED IN 2010-11. OFFERED IN 2011-12. Many elegant computational problems arise naturally in the modern study of molecular biology. This course is an introduction to the design, implementation, and analysis of algorithms with applications in genomics. Topics include bioinformatic algorithms for dynamic programming, tree-building, clustering, hidden Markov models, expectation maximization, Gibbs sampling, and stochastic context-free grammars. Topics will be studied in the context of analyzing DNA sequences and other sources of biological data. Applications include sequence alignment, gene-finding, structure prediction, motif and pattern searches, and phylogenetic inference. Course projects will involve significant computer programming in Java. No biology background is expected. *Not open to students who have taken 303. Alternate year course.*

Prerequisite: 230
Distribution: Mathematical Modeling
Semester: N/O. Offered in 2011-12. Unit: 1.0

CS 315 Web Search and Mining

Metaxas

In the last decade we have experienced an explosive growth of information through the web. Locating information seems to be very easy, while determining the quality of information can be tricky. This course is for students who want to know why search engines can answer your queries fast and (most of the time) accurately, why other times seem to be missing the point and provide untrustworthy information, and how one can design a Web site that acquires high visibility on the Web. We will cover traditional information retrieval methods and web search algorithms such as crawlers and spiders, with a focus on probabilistic and graph-theoretic methods that can detect Web spam. We will also cover some basic understanding of text mining and data clustering. Time permitting, we will examine other relevant issues of the information explosion era, such as the shape and structure of the Web, epistemology of information and properties of large random networks. *Not open to students who have taken this topic as CS 349. Alternate year course.*

Prerequisite: 230
Distribution: Mathematical Modeling
Semester: Fall Unit 1.0

CS 320 Tangible User Interfaces

Shaer

Tangible user interfaces emerge as a novel human-computer interaction style that interlinks the physical and digital worlds. Extending beyond the limitations of the computer mouse, keyboard and monitor, tangible user interfaces allow users to take advantage of their natural spatial skills while supporting collaborative work. Students will be introduced to conceptual frameworks, the latest research, and a variety of techniques for designing and building these interfaces. Developing tangible interfaces requires creativity as well as an interdisciplinary perspective. Hence, students will work in teams to design, prototype and physically build tangible user interfaces. *Not open to students who have taken this course as a topic of CS 349.*

Prerequisite: CS 220 or CS 215 or CS 230, or permission of instructor
Distribution: Mathematical Modeling
Semester: Fall Unit: 1.0

CS 332 Visual Processing by Computer and Biological Vision Systems

Hildreth

NOT OFFERED IN 2010-11. OFFERED IN 2011-12. An introduction to algorithms for deriving symbolic information about the three-dimensional environment from visual images. Aspects of models for computer vision systems will be related to perceptual and physiological observations on biological vision systems. Computer vision software written in MATLAB will be used to implement and test models. Topics include: edge detection, stereopsis, motion analysis, shape from shading, color, visual reasoning, object recognition. *Students are required to attend an additional 70-minute discussion section each week. Alternate year course.*

Prerequisite: 112 or 230 or permission of the instructor.
Distribution: Mathematical Modeling
Semester: N/O. Offered in 2011-12. Unit: 1.0

CS 342 Computer Security

Turbak

An introduction to computer security. Topics include ethics, privacy, authentication, access control, information flow, operating system security (with a focus on Linux), cryptography, security protocols, intrusion prevention and detection, firewalls, viruses, network security, Web security, programming language security. Assignments include hands-on exercises with security exploits and tools in a Linux environment. Participants will independently research, present, and lead discussions on security-related topics. *Alternate year course.*

Prerequisite: 230 and 240 or permission of the instructor. CS 242 recommned.
Distribution: Mathematical Modeling
Semester: Fall Unit: 1.0

CS 349 Advanced Topics in Computer Science

TBA

Topic for Spring 2011: TBA

Prerequisite: TBA
Distribution: TBA
Semester: Spring Unit: 1.0

CS 350 Research or Individual Study

Prerequisite: Open by permission to juniors and seniors.
Distribution: None
Semester: Fall, Spring Unit: 1.0

CS 350H Research or Individual Study

Prerequisite: Open by permission to juniors and seniors.
Distribution: None
Semester: Fall, Spring Unit: 0.5

CS 360 Senior Thesis Research

Prerequisite: By permission of the department. See Academic Distinctions.
Distribution: None
Semester: Fall, Spring Unit: 1.0

CS 370 Senior Thesis

Prerequisite: 360 and permission of department.
Distribution: None
Semester: Fall, Spring Unit: 1.0

Requirements for the Major

Students majoring in computer science must complete CS 111, 230, 231, 235, 240, 251, two 300-level courses other than 350, 360 or 370, and at least one additional computer science course at the 200 or 300 level. Students who do not take CS 111 must replace this requirement with an additional one-unit computer science course. Computer science courses at other institutions used to meet the nine-course requirement must be approved **in advance** by the department chair on an individual basis. In addition, all majors in computer science are expected to complete (1) MATH 225 (Combinatorics and Graph Theory) and (2) at least one additional course in mathematics at the 200 or 300 level. Particularly relevant mathematics courses are MATH 206 (Linear Algebra), MATH 220 (Probability and Elementary Statistics), MATH 223 (Number Theory), MATH 305 (Abstract Algebra), MATH 309 (Foundations of Mathematics), and MATH 325 (Graph Theory). The MATH courses are considered prerequisites for the CS degree and could be counted toward another major. Students should consult a computer science faculty member for advice in choosing mathematics courses best suited to their interests. Students are encouraged to consult the current computer science student handbook (linked from the department's Web site http://cs.wellesley.edu) for suggestions of possible course schedules for completing the major. Students considering a junior year abroad should consult a faculty member in the department as soon as possible in their sophomore year to plan a schedule of courses to complete the major.

All computer science majors are required to participate in computer science student **seminars** held throughout the academic year. In these seminars, students have the opportunity to explore topics of interest through reading and discussion, field trips, invited speakers, independent research projects, or software development projects.

Requirements for the Minor

The computer science five-course minimum minor is recommended for students whose primary interests lie elsewhere, but who wish to obtain a fundamental understanding of computer science. The minor consists of CS 111, 230, one of 231, 235, or 240, at least one computer science course above 100-level, and at least one 300-level computer science course other than 350. Students who do not take CS 111 must replace this requirement with one additional one-unit computer science course except 110.

Honors

Students can earn honors in computer science by successfully completing an honors-quality senior thesis. A detailed description of the senior thesis project in computer science can be found in the document *Independent Studies in Computer Science* (http://cs.wellesley.edu/~cs/Research/thesis.html). Majors who are interested in undertaking a senior thesis project are urged to discuss their plans with either their advisor or the department chair as early as possible in their junior year.

Graduate Study

Students who plan to pursue graduate work in computer science are strongly encouraged to develop their background in mathematics, particularly in the areas of linear algebra, logic, probability and statistics, number theory, and graph theory. In addition, students who are planning either graduate work or advanced technical research or development work are strongly encouraged to (1) obtain laboratory experience by electing one or more of 303, 307, 332, 342, or appropriate courses at MIT and (2) pursue at least one independent study or research project before graduating, in the form of a Wellesley course (250/350/360), an MIT UROP, or a summer internship. Consult http://cs.wellesley.edu/~cs/Research for more details.

Advanced Placement Policy

Students may receive a maximum of one unit of college credit for a score of 5 on the Computer Science A or AB Advanced Placement exam. This unit does not count towards the computer science major or minor. Students receiving AP credit for computer science should consult with the department regarding enrollment in 230 or 240. Computer science majors and minors should consult with a computer science faculty advisor before electing to take a computer science course as credit/noncredit.

Computer Engineering

Students interested in computer engineering should consult the course listings in Extradepartmental and enroll in EXTD 160, Introduction to Engineering Science. This course is intended to be a gateway experience for possible subsequent engineering studies, such as the engineering certificates from the Olin College of Engineering. The Special Academic Programs section contains a description of these certificates that represent groups of engineering courses at Olin designed to complement a major at Wellesley. More information at http://cs.welleseley.edu/~cs/Curriculum/olin.html.

Interdepartmental Majors

Students interested in an interdepartmental major (or minor, if applicable) in cognitive and linguistic sciences, media arts and sciences, or neuroscience are referred to these listings in the catalog.

Students interested in engineering should consult the course listings in Extradepartmental.