Computer Science at Vassar

2007-2008

Department Handbook (v 1.96.4)

Department of Computer Science
Campus Box 732 — Vassar College
Poughkeepsie, New York 12604-0732
Telephone (845) 437-5985

TABLE OF CONTENTS

Computer Science at Vassar

The Computer Science Program at Vassar integrates study of the important theoretical foundations of the field with in-depth study of the scientific methodologies that are central to our discipline, providing a solid background for future work in the field. Intense study of computer science in the context of a broad liberal arts education opens up perspectives and opportunities for cross-disciplinary study that are especially relevant in today's world. Recent graduates have been accepted for graduate study at top Computer Science programs, including Stanford, MIT, Cornell, Wisconsin–Madison, Maryland–College Park, Brown, Columbia, Harvard, and Yale. Students choosing industry have gone on to positions at companies of all sizes — from small start-ups to companies like Microsoft and Pixar.

The CS major is rigorous, requiring 13 units for the major, including two intermediate-level courses in Mathematics. At the introductory and intermediate levels, students are introduced to a variety of different programming paradigms and are exposed to the foundational concepts of data structuring and analysis as well as the theory of computation and algorithm analysis. Although substantial programming is a part of most courses, the emphasis throughout the curriculum is on the fundamental concepts and problem-solving strategies that make the discipline a science. In this way, students are prepared for future work in the field, independent of the particular language or paradigm that is in vogue at the moment.

The Department offers an unusually wide variety of advanced-level courses, where the student sees how the concepts and strategies studied in earlier courses come together in real-world applications. These courses are offered regularly and cover areas such as Operating Systems, Compilers, Networks, Graphics, Artificial Intelligence, Parallel Programming, Databases, Digital Electronics, and Computer Architecture. Advanced level study of Computer Science Theory is also available, which is especially valuable for students going on to graduate school.

Students majoring in other disciplines may wish to complete a Correlate Sequence in computer science, with the option of several different tracks including programming languages, architecture, algorithms, and advanced study of theoretical computer science and artificial intelligence.

The Computer Science Department Handbook is intended to provide answers to the most commonly asked questions about computer science at Vassar. Please feel free to contact the department chair or any member of the computer science faculty if you need additional information.

Contact Information

Faculty:

E-mail addresses for all faculty members are located in the "Faculty" section.

Department Office:

E-mail: csdept@vassar.edu

URL: http://www.cs.vassar.edu

Phone: (845) 437-5985

Fax: (845) 437-7498

Sun Lab:

Phone: x4989 (cannot be reached from off campus)

CS Lounge

Phone: x4799 (cannot be reached from off campus)

Students in Computer Science

Students in computer science are involved in many aspects of the department's operation. The *Majors' Committee* meets several times a semester, and has input on a wide range of academic issues including faculty tenure decisions and course offerings. The Majors' Committee is headed by a pair of computer science majors, with new committee chairs being elected at the end of each fall semester. This guarantees that the former chairs are in the department for at least a semester to help bring the new chairs up to speed. This committee normally hosts several social events during the academic year for all computer science majors, including a year-end picnic in May just before graduation.

The *Vassar Women in Computer Science* is a social organization dedicated to fostering female computer science students who are traditionally a minority in the computer science discipline. This social group normally has several activities scheduled during each semester. One of the major recent activities of this organization is a trip to Orlando, Florida, for the *Grace Hopper Celebration of Women in Computing*.

Computer Science Major and Correlate Requirements

The table below lists courses required for the major. In addition, students must take two other 300-level computer science electives. No course numbered 200 or higher may be elected NRO and counted towards the major.

Units	Course	Title
1	CMPU 101	Computer Science I
1	CMPU 102	Computer Science II
1	CMPU 203	Computer Science III
1	CMPU 224	Computer Organization
1	MATH 263	Discrete Mathematics*
1	MATH 221	Linear Algebra*
1	CMPU 240	Language Theory and Computation
1	CMPU 241	Algorithmics
1	CMPU 245	Declarative Programming Models
1	CMPU 331	Compilers
1	CMPU 334	Operating Systems

*** Calculus is required for these courses.**

Recommendations

- **Prospective majors are strongly advised to complete Computer Science 101 and 102 by the end of the freshman year. Students who intend to pursue graduate studies in computer science are strongly urged to take Computer Science 342.**
- **When planning the major, keep in mind that many courses are offered only during the first or second semester of each year. Check the Vassar course catalog for the latest information.**
- **Students with programming experience can choose to take CMPU 125, the condensed version of the 101/102 sequence, with permission of the department.**
- **Students even *considering* a major should try to complete CMPU 101/102 by the end of the Freshman year. This makes it much easier to schedule the rest of the courses required for the major.**
- **Students majoring in the sciences are advised to complete the CMPU 101/102 sequence, or to complete a correlate sequence in computer science.**

Sample Course Sequences for the Computer Science Major

The tables in this section illustrate possible course sequences through the major. ***Note that while calculus is not an explicit requirement for the Computer Science major, it is a prerequisite for 200-level math courses.*** **The first table below assumes that the Math department's calculus requirement has been met by an Advanced Placement course or other transfer credit, and that the Computer Science introductory sequence was begun in the Fall of the Freshman year:**

Year	A-Semester	B-Semester
Freshman	CMPU 101 MATH 221	CMPU 102 MATH 263
Sophomore	CMPU 203 CMPU 224	CMPU 245 CMPU 241
Junior	CMPU 240 CMPU 334	CMPU 331
Senior	CMPU 300-level elective	CMPU 300-level elective

This next table illustrates a possible sequence of courses assuming that Math 125 was taken at Vassar to satisfy the calculus requirement, and that the Computer Science introductory sequence was begun in the Fall of the Freshman year:

Year	A-Semester	B-Semester
Freshman	CMPU 101 MATH 125	CMPU 102 MATH 263
Sophomore	CMPU 203 MATH 221	CMPU 224 CMPU 245
Junior	CMPU 240 CMPU 334	CMPU 241 CMPU 331
Senior C	CMPU 300-level elective C	CMPU 300-level elective

If preferred, one can take a two-course sequence to fulfill the Math Department's calculus requirement. Math 263 therefore cannot be taken until the Sophomore year, as shown in the table below:

Year	A-Semester	B-Semester
Freshman	CMPU 101 MATH 121	CMPU 102 MATH 122
Sophomore	CMPU 203 MATH 221	CMPU 224 MATH 263
Junior	CMPU 240 CMPU 334	CMPU 241 CMPU 331
Senior C	CMPU 300-level elective	CMPU 245 CMPU 300-level elective

Finally, this last table shows a possible sequence of courses for those who begin the Computer Science introductory sequence in their second semester, and satisfy the calculus requirement at Vassar:

Year	A-Semester	B-Semester
Freshman	MATH 121	CMPU 101 MATH 122
Sophomore	CMPU 102 CMPU 224	CMPU 203 MATH 263
Junior	CMPU 240 MATH 221	CMPU 241 CMPU 331
Senior C	CMPU 300-level elective CMPU 334	CMPU 245 C CMPU 300-level elective

Scheduling a major becomes more difficult if CMPU 101 is delayed until the Sophomore year or the Math department's calculus requirement isn't completed during the Freshman year. Please see a faculty member for advice in those cases.

Correlate in Computer Science

Students majoring in other programs may complement their study by electing a correlate sequence in Computer Science. Selection of the appropriate option should be made in consultation with the Computer Science faculty to ensure exposure to the areas of Computer Science most useful to the field of concentration.

Requirements for the Correlate: **Computer Science 101, 102 and 203; any two of 224, 240, 241 and 245 (at least one of which must be either 240 or 241), plus any 300-level Computer Science course. No course numbered 200 or higher may be elected NRO and counted toward the requirements for the correlate. Suggested correlate sequences include the following, in addition to Computer Science 101, 102 and 203:**

Architecture: 224, 241 and (324 or 325).

Software Systems: 224, 241 and (334 or 335).

Programming Languages: 224, 240 and 331.

Artificial Intelligence: (240), 245 and 365.

Graphics: 241, (224, 240 or 245) and 378.

Theory: 240, 241 and 342.

Computing in Relation to Specific Disciplines

Students majoring in a variety of disciplines may require a level of computer understanding provided by taking one or more specialized courses. For example, a student majoring in Economics may need to modify a program in statistical analysis in order to apply it to a specific problem, or it may be necessary to write a routine that connects a statistics program to a database program. All students interested in computing should take the introductory level courses, CMPU 101, *Computer Science I* and CMPU 102, *Computer Science II.* Students first encounter the theory of computer science in the introductory courses. These courses provide a conceptual context for advanced and supplementary topics.

Computing Courses for Science Students

Students in the sciences should take CMPU 101, *Computer Science I* and CMPU 102, *Computer Science II*. Students who intend to take other advanced courses in computing should first take CMPU 224, *Computer Organization,* which examines the hierarchical structure of computer systems.

Computing Courses for Social Science Students

Students majoring in the social sciences should take CMPU 101, *Computer Science I* and CMPU 102, *Computer Science II.* Social science students with a deeper interest in computing should also take CMPU 203, *Computer Science III.*

Computing Courses for Humanities Students

Humanities students, especially those majoring in language and literature, should take CMPU 101, *Computer Science I* and CMPU 102, *Computer Science II.* Students with deeper interest may also elect to take CMPU 240, *Language Theory and Computation*.

Cognitive Science

Cognitive Science majors should take CMPU 101, *Computer Science I* and CMPU 102, *Computer Science II.* The intermediate and upper-level Computer Science courses which are of primary interest to Cognitive Science majors are: CMPU 203, *Computer Science III,* CMPU 240, *Language Theory and Computation*, CMPU 365, *Topics in Artificial Intelligence.* Other courses such as CMPU 334, *Operating Systems* may also be of interest to Cognitive Science majors.

Students emphasizing Computer Science as part of their Cognitive Science major should consult with their advisors about additional courses in Computer Science if they are planning to do graduate work. (See the section, Preparation for Graduate Study, below.)

Independent Program — Computer Science Major

An Independent Major can be declared which involves the study of some area within computer science along with one or more other disciplines. The student who applies to the Independent Program in order to create such a major must indicate the exact sequence of courses he or she intends to take. No specific course sequence is prescribed for an independent major involving computer science, and the student's choice of courses will be dictated to a large extent by the nature of the second discipline. However, independent computer science majors should take CMPU 101, *Computer Science I*, CMPU 102, *Computer Science II,* CMPU 224, *Computer Organization* and, depending upon the student's interests, at least one of CMPU 203, *Computer Science III* or CMPU 334, *Operating Systems*. Students should read the sections outlining the courses offered in Computer Science and consult with their advisors to determine additional courses in Computer Science relevant to their goals. *Consultation with the advisor in Computer Science is essential in designing an independent major. S*tudents interested in this option are encouraged to contact the department chair and the director of the Independent Program as early as possible, so that an appropriate advisor can be assigned.

Declaring a Major

Declaring a Computer Science Major

The first step in declaring a major in Computer Science is to consult with the chair of the department. At this meeting the general requirements for pursuing a major will be discussed and a faculty member will be selected to be the student's major advisor. A meeting with the advisor to determine which courses will be taken and when is then necessary. After this has been established, the major is declared by submitting the proposed course outline and the appropriate forms to the Registrar's Office.

Declaring a Cognitive Science Major with a Computer Science Component

Students who plan to major in Cognitive Science and elect an option involving Computer Science should meet with the director of the Cognitive Science Program as well as with a member of the Computer Science faculty involved in the Cognitive Science Program who will serve as an informal advisor within the CS Department.

Declaring an Independent Major with a Computer Science Component

Students choosing an Independent Major involving Computer Science must meet with the director of the Independent Program, with advisor(s) in disciplines involved in the major outside Computer Science, and with a member of the Computer Science faculty who will serve as advisor within the CS Department. Students who are unsure which CS faculty member would best serve as advisor for his or her independent major should meet with the chair. The advisors from CS and other relevant departments devise a course sequence appropriate to fulfill the independent major. The student must submit a proposal for the Independent Major together with a proposed course sequence, as outlined in the guidelines for the Independent Major in the Vassar Catalogue. The necessary forms for declaring a major are obtained from the CS department office and the Registrar's Office.

Advanced Placement

Students eligible for Advanced Placement may be able to bypass Computer Science 101. Please consult with the department.

Independent Research

The computer science department strongly encourages students to engage in independent research with faculty, and offers ungraded courses CMPU 298 and 399 for this purpose. The department also offers Computer Science 300-301, a graded research experience for senior majors. Computer Science 300-301 may not be substituted for 300-level elective courses satisfying the requirements for the major. Satisfactory completion of Computer Science 300-301 is required for departmental honors.

Transfer Credits

Students may be eligible to receive Vassar credit for courses in computer science taken elsewhere. In order to qualify for credit, a course must be consistent with the Vassar curriculum. Courses in data processing (e.g., COBOL, RPG, spreadsheet applications, etc.), systems operation, information services, etc., are not eligible. Requests for transfer credit approval should be made to the department chair. Transfer credit in computer science may be awarded regardless of a student's major field of study.

Field Work

Students who undertake summer or part-time jobs in computing may be eligible for field work credit. In order for work to qualify for credit a field work job must meet the following criteria:

- **The activity must be directly related to the academic study of computer science. (Activities in data processing, systems operation, management information systems, etc. do not qualify for credit.)**
- **The activity must enhance the student's knowledge of Computer Science. Learning a new language, new algorithm design techniques, etc., would qualify. Elementary applications coding would not.**

In order to obtain field work credit you must have the work approved by a member of the Computer Science Department *before undertaking the work*. Upon receiving such approval, you should obtain the necessary forms and guidelines from the Field Work Office. Upon completion of a field work project a written report (5-10 pages) must be submitted to the faculty supervisor. The report should indicate the nature of the work performed and the way(s) in which it enhanced your studies at Vassar.

Dartmouth Engineering Program

Vassar students interested in engineering can take advantage of our participation in Dartmouth's dual-degree engineering program. Students spend their Junior year at Dartmouth taking courses at the Thayer School of Engineering, return to Vassar for their Senior year, then do an additional fifth year at Dartmouth. At the end of five years, students will have earned a B.A. in Computer Science from Vassar and a B.E. (Bachelor of Engineering) from Dartmouth's professionally accredited engineering program.

Students interested in this program should speak with an advisor in the Computer Science department as early as possible, as careful planning is required to ensure all of the degree requirements are met. Students must apply to Dartmouth for the dual-degree program by March 1st of their Sophomore year. Additional information on the program is available in the department office or from the Dean of Studies.

Interns and Student Aides

Summer Interns

Periodically, organizations such as IBM, AT&T, DEC, NSF, etc. offer summer internships in computing to undergraduates. Within the college, there are opportunities for summer research internships within the URSI (Undergraduate Research Summer Institute) program, which enables students to be involved in research with faculty. Opportunities vary from year to year. They are intended primarily for students who have completed their Junior year, though occasionally Sophomores and Seniors are eligible. Students who are interested in these positions should prepare a résumé and submit it to the department chair.

Student Aides

In addition to Departmental Interns, the Computer Science Department employs approximately ten student aides during the academic year. The aides work as tutors for introductory computing courses, laboratory assistants for various faculty members and assistants in the department office. Support for these positions is provided by the Financial Aid Office. Interested students should contact the department.

Departmental Honors

In order to graduate with honors in Computer Science, students must meet all of the following criteria:

Satisfactory completion of Computer Science 300-301 is required for departmental honors. Also:

- **A G.P.A. of 3.5 or higher in the 200 and 300 level courses within the major.**
- **A G.P.A. of 3.3 or higher overall.**
- **Nomination by the faculty.**

or

- **A G.P.A. of 3.3 or higher in the 200 and 300 level courses within the major.**
- **Nomination by the faculty due to exceptional service and/or demonstrated excellence.**

Preparation for Graduate Studies

As mentioned in the section on Computing as a Major Field of Study, the major in Computer Science is the best preparation for further study in graduate departments of Computer Science. Students intending to pursue graduate work in computer science should take CMPU 342.

A number of top-quality graduate schools now offer programs that focus on artificial intelligence and computational linguistics, and for such programs the major in Cognitive Science or an independent major that emphasizes work in artificial intelligence and language processing would provide a good background.

The Graduate Record Examination in Computer Science is now required for admission to many graduate programs. Planning for this early — and taking the appropriate Mathematics and Computer Science courses — will make entry into graduate programs much easier.

Course Descriptions

This section gives short descriptions of the courses offered by the Computer Science department. The header before each description gives the course number, its title, and a list of prerequisite courses. In some cases, prerequisites can be bypassed with the permission of the instructor. Not all courses are offered each semester. Please consult the Vassar course catalog to determine which are being offered any given semester, and whether a course is typically offered during the A semester, B semester, or both.

CMPU 101 ***Computer Science I: Problem-Solving and Abstraction*** Pre**: NONE**

Introduces the design and implementation of algorithms to solve computational problems, using an object-oriented programming language. Topics include procedural abstraction, expression evaluation, flow of control constructs and recursion; data abstraction, classes, inheritance and interfaces; elementary data structures (e.g., arrays, strings, vectors, lists, stacks, queues); input/output and event-driven programming. The course emphasizes principles of program design and data organization. A weekly laboratory period provides guided hands-on experience.

CMPU 102 ***Computer Science II: Data Structures and Algorithms*** Pre**: CMPU 101**

Development of data structures and algorithms in an object-oriented programming language. Topics include hierarchic program refinement, preconditions, postconditions and invariants; data encapsulation and fundamental data structures (e.g., priority-queues, sets, maps, heaps, search trees, hash tables and graphs); fundamental algorithms (e.g., searching and sorting) and analysis of algorithm complexity. A weekly laboratory period provides guided hands-on experience.

CMPU 125 ***Topics in Computer Science*** Pre**: NONE**

Material from Computer Science 101/102 presented in one semester for students with previous experience in the field. Topics in 102 are fully developed and topics in 101 are reviewed.

CMPU 203 ***Computer Science III: Software Design and Implementation*** Pre**: CMPU 102**

Develops techniques for design and implementation of complex software systems. Topics include object-oriented modeling, design patterns, component libraries, multiple inheritance, parametric polymorphism, generic algorithms, containers, iterators, function objects and storage management. Development of a software system of significant complexity is required. A weekly laboratory period provides guided hands-on experience.

CMPU 224 ***Computer Organization*** Pre**: CMPU 102**

Examines the hierarchical structure of computing systems, from digital logic and micro-programming through machine and assembly languages. Topics include the structure and workings of the central processor, instruction execution, memory and register organization, addressing schemes, input and output channels and control sequencing. The course includes a weekly hardware/software laboratory where digital logic is explored and assembly language programming projects are implemented.

CMPU 240 ***Language Theory and Computation*** Pre**: CMPU 203, MATH 263**

Study of regular sets, context free grammars and languages, finite and push-down automata, as well as more powerful models of computation, such as Turing machines. Provides theoretical foundations for Computer Science 331, Compiler Design.

CMPU 241 ***Algorithmics*** Pre**: CMPU 203, MATH 263**

Introduces the systematic study of algorithms and their analysis with regard to time and space complexity. Topics include divide-and-conquer, dynamic programming, greediness, randomization, upper and lower-bound analysis, and introduction to NP completeness. Emphasis is placed on general design and analysis techniques that underlie algorithmic paradigms. Builds a foundation for advanced work in computer science.

CMPU 245 ***Declarative Programming Models*** Pre**: CMPU 102 and MATH 263**

Declarative programming languages are important alternatives to the imperative languages used in most software systems. This course covers two kinds of declarative programming: functional programming and logic programming. Topics include the operational and denotational semantics of declarative languages, techniques for programming in declarative languages, and the use of mathematical logic as a tool for reasoning about programs.

CMPU 295 ***Special Topics*** Pre**: Permission**

Intermediate level treatment of specialized topics in computer science such as networking, computational linguistics.

CMPU 300a-301b ***Senior Research and Thesis***

Pre**: 3.5 GPA in 200 and 300-level Computer Science coursework at the end of the junior year, and permission of the Department**

Investigation and critical analysis of a topic in experimental or theoretical computer science. Experimental research may include building or experimentation with a non-trivial hardware or software system. A student electing this course must first gain, by submission of a written research proposal, the support of at least one member of the computer science faculty with whom to work out details of a research strategy. The formal research proposal, a written thesis, and oral presentation of results are required for the course. A second faculty member participates in both the planning of the research and final evaluation.

CMPU 324 ***Computer Architecture*** Pre**: CMPU 224**

An exploration of current research areas in computer organization including an examination of data-flow, microcode, cache memory, distributed, parallel and other nonstandard architectures, and related topics.

CMPU 325 ***Microcomputers and Digital Electronics*** Pre**: CMPU 224**

An advanced seminar in the architecture and implementation of microprocessors. Topics include digital logic, memory and processor interfaces, interrupt handling, and serial I/O methods. Differences among logic implementations such as TTL, CMOS, ECL, etc., are considered. Students participate in the design and implementation of a microcomputer.

CMPU 331 ***Compilers*** Pre**: CMPU 224, CMPU 240, CMPU 245 or Permission**

Studies the theory of automata for language recognition as well as the implementation of actual compilers for programming languages. During the semester students develop modules comprising the front-end of a compiler for a high-level computer.

CMPU 334 ***Operating Systems*** Pre**: CMPU 203, CMPU 224**

Deals with the theory and implementation of the software which governs the management of system resources. Topics that are covered include file organization, process scheduling, system services, memory management, security methods, resource contention, and design principles. Operating systems for parallel and distributed processing, real-time processing, virtual machines, and networking are also considered.

CMPU 335 ***Software Development Methodology*** Pre**: CMPU 203**

Presents a systematic methodology for developing large software systems, focusing on the specification, modeling and design phases of the software development process. Topics include class hierarchies, aggregation, class relationships and use-case analysis, among others. The course also touches on relevant notions of software architecture and middleware. Concepts are reinforced in group projects.

CMPU 342 ***Topics in Theoretical Computer Science*** Pre**: Permission**

Investigates a selected topic in theoretical computer science. The topic will be chosen each year according to the interests of students and faculty. Potential topics include algorithms, complexity, computability, programming language semantics and formal methods, among others.

CMPU 353 ***Bioinformatics*** Pre**: CMPU 203 or Permission**

Bioinformatics is an interdisciplinary field of study that applies computational techniques to problems in biology. Biology and computer science majors learn to formulate novel questions in genomics, and conduct computational biology experiments of their own design. Topics include scientific modeling, experimental computer science, gene prediction and genome annotation, sequence alignment and phylogeny, and evolutionary and comparative genomics.

CMPU 365 ***Artificial Intelligence*** Pre**: CMPU 203, CMPU 245**

An introduction to Artificial Intelligence as a discipline of Computer Science, covering the traditional foundations of the field and a selection of recent advances. Traditional topics include: search, two-player adversarial games, constraint satisfaction, knowledge representation and reasoning, and planning. Additional topics will vary from year to year and will be selected from the following: reasoning about time, probabilistic reasoning, neural networks, philosophical foundations, multi-agent systems, robotics, recent advances in planning. Significant programming assignments and a course project highlight the material presented in class.

CMPU 375 ***Networks*** Pre**: CMPU 203 or Permission**

Provides a detailed introduction to network protocols and software, as well as a discussion of network architectures and technology. Topics covered include properties of various transmission media, methods for reliable transfer of data, Ethernet and local-area networks, TCP/IP and the Internet, routing, security. Programming assignments and a project emphasize the key concepts.

CMPU 376 ***Database Design*** Pre**: CMPU 203**

Concerned with the theory and techniques of database design and the organization of query and command languages. The differences among relational, hierarchical, and networked databases are considered. Topics include data independence, data dictionaries, data models, entity-attribute relationships, access methods and security issues.

CMPU 377 ***Parallel Programming*** Pre**: CMPU 203**

An introduction to parallel computing, with coverage of parallel architectures, programming models, and techniques. Topics include SIMD and MIMD models, shared-memory and message-passing styles of computation, synchronization, deadlock, and parallel language design. Students are exposed to common techniques for solving problems in sorting, searching, numerical methods, and graph theory, and gain practical experience through programming assignments run on a parallel processing system.

CMPU 378 ***Graphics*** Pre**: Math 221 and CMPU 203**

Introduction to computer graphics: 3D modeling and viewing, geometric transformations, visible surface detection methods, illumination and shading models, surface rendering methods (including ray-tracing and radiosity), and color models. A brief review of the mathematics for computer graphics: coordinate systems, vector products, linear algebra, and parametric representations.

CMPU 388 ***Computer Animation: Art, Science and Criticism*** Pre**: CMPU 378 and Permission**

An interdisciplinary course in Computer Animation aimed at students with previous experience in Computer Science, Studio Art or Media Studies, but not necessarily more than one of these areas. The course introduces students to mathematical and computational principles and techniques used to describe the shape and motion of three-dimensional figures in Computer Animation. It introduces students to artistic principles and techniques used in drawing, painting and sculpture, as they are translated into the context of Computer Animation. It also encourages students to critically examine Computer Animation as a medium of communication. Finally, the course exposes students to issues that arise when people from different scholarly cultures attempt to collaborate on a project of mutual interest. The course is structured as a series of animation projects interleaved with screenings and classroom discussions. Students carry out their projects working in pairs or small groups, using state-of-the-art modeling and animation software. In classroom discussions, students critically evaluate their project work, and reflect on the process of interdisciplinary collaboration itself.

CMPU 395 ***Advanced Special Topics*** Pre**: CMPU 101, 203 or Permission**

An in-depth treatment of specialized topics in computer science such as networking, computational linguistics, large system design or programming language semantics.

Topic for 2007/08: *Computational Linguistics.* This course addresses the fundamental question at the intersection of human languages and computer science: how can computers acquire, comprehend and produce natural languages such as English? Introduces computational methods for modeling human language, including morphology, syntax, semantics, and discourse; corpus-based and stochastic methods for language analysis; and natural language applications such as information retrieval, machine translation, and computational lexicography. Programming experience is recommended but not required.

CMPU 399 ***Senior Independent Work*** Pre**: Permission**

Equipment and Facilities

The Computer Science Department has a dedicated network of workstations running a variety of operating systems, housed in two laboratories located within the department. A laboratory of fourteen high-powered hyperthreading computers running a customized version of Linux supports introductory level courses. Students in intermediate and advanced-level courses have exclusive access to a laboratory of twenty-two dual-core high-resolution graphics workstations. Students also have access to a Linux-based high performance computing cluster supporting multiple parallel and distributed computing paradigms. The department's servers provide students with remote access to departmental equipment over secure authenticated connections.

Faculty

THOMAS ELLMAN (ellman@cs.vassar.edu), THOMAS ELLMAN (ellman@cs.vassar.edu), Associate Professor, earned his Ph.D. from Columbia University. His interests include computer animation, artificial intelligence and formal methods in software engineering. His research is currently focused on specification and synthesis of interactive animation programs.

LUKE HUNSBERGER (hunsberg@cs.vassar.edu), Assistant Professor, earned his Ph.D. in Computer Science from Harvard University in 2002 and holds an M.A. in mathematics from the University of Oregon. His interests include collaborative planning and decision making in multi-agent systems.

NANCY IDE (ide@cs.vassar.edu) is Professor of Computer Science. She has been a member of the computer science faculty at Vassar since 1982. She earned B.A., B.S., M.A., and Ph.D. degrees from The Pennsylvania State University. Professor Ide has written a textbook for introductory computing as well as numerous papers in the fields of computational linguistics. In 1987 she initiated the international Text Encoding Initiative, supported by the National Endowment for the Humanities and the European Community, and is the recipient of several grants funded by the National Science Foundation. Professor Ide is currently Editor-in-Chief of the journal *Computers and the Humanities* and co-editor of a book series *Text, Speech, and Language Technology* for Kluwer Academic Publishers.

MARC SMITH (mlsmith@cs.vassar.edu) Assistant Professor, earned his B.S., M.S., and Ph.D. degrees from the University of Central Florida. He worked for AT&T for fifteen years, in several different IT capacities, during which time his Ph.D. studies were sponsored by AT&T's Doctoral Support Program. His research spans elements of theoretical and experimental computer science, in the area of parallel and distributed computation. His interests include models of concurrency, bioinformatics (specifically, computational phylogeny), and programming languages (semantics, paradigms, and unifying theories).

LOUIS VOERMAN (voerman@cs.vassar.edu) is a Visiting Associate Professor of Computer Science at Vassar, where he has taught courses in operating systems, microprocessor design, and advanced architecture since 1983. He earned a B.S. in Electrical Engineering and an M.S. in Computer Science from Union College. He worked for IBM as a Senior Programmer until the summer of 2000, where he was involved in the design of intelligent workstations and operating systems. He has taught advanced computing courses within IBM for several years. His interests include operating systems, microprocessor architecture, and computer organization.

JENNIFER WALTER (walter@cs.vassar.edu), Assistant Professor, received her M.S. and Ph.D. from Texas A&M University. Prior to those degrees, she had earned a B.A. in Computer Science and a B.A. in Geology from the University of Minnesota, Morris. Her research involves the development and simulation of distributed algorithms for mobile, ad hoc networks and self-reconfigurable robotic systems. Her interests include the analysis of algorithms, theory of computation, computational complexity, computational geometry, and robotics.