Colby

COMPUTER SCIENCE

Chair, *Professor Bruce Maxwell*
Professors Dale Skrien and Bruce Maxwell; Assistant Professor Stephanie Taylor

Computer science studies the design of computational processes, computing systems, and virtual objects. Our goal is to provide students with a strong background in computer science, including the integration of knowledge from other disciplines. Our graduates will have the ability and experience to enable and to produce new and innovative discoveries.

Students with a variety of interests may want to explore computer science, as it affects and interacts with virtually every discipline. Many advances in the natural and social sciences, engineering, and the humanities would not have been possible without the exponential growth in computing power and the corresponding design of advanced algorithms by computer scientists. Students who become majors or minors, or take just a few courses, will extend their potential by knowing more about how to effectively use computers and computation.

Students in computer science courses learn primarily through programming projects that provide them with experience in design, the application of computational thinking, and problem solving. Computational thinking is the ability to deconstruct a problem or process and describe it at the level of computable operations. Computational thinking integrates abstraction, hierarchical design, information management, and an understanding of complexity. The projects students undertake increase in scope and complexity both within a single course and as they progress through the major.

The computer science major prepares students for graduate work in computer science and related areas or a wide variety of careers. The computer science minor provides students with the ability to effectively apply computational thinking to other disciplines. The interdisciplinary computation majors in biology, environmental studies, or theater and dance give students the opportunity to integrate computer science with a focus discipline. Students interested in any of these programs should enroll in Computer Science 151 in their first year.

Requirements for the Major in Computer Science

Computer Science 151, 231, 232, 251, 333, and 375 or 378; one elective numbered 200 or above; three electives numbered 300 or above, including at least one fall-spring sequence; and one of the following courses: Mathematics 212, 231, 253, or 274. Students may count only Computer Science 151, 231, and 251 towards both the computer science major and any other major.

Requirements for the Honors Program in Computer Science

An honors program is available for students who wish to pursue a topic more deeply. Students must have a grade point average of at least 3.25 in all computer science courses numbered 200 or higher and complete a year-long, preapproved honors project (Computer Science 483 and 484) culminating in both a written paper and a colloquium presentation. The fall semester project satisfies an elective in the major requirements. Students who successfully complete the requirements and receive the recommendation of the department will graduate “With Honors in Computer Science.”

Requirements for the Minor in Computer Science

Computer Science 151, 231, 251, one course numbered 200 or above, one course numbered 300 or above, and a capstone experience. The capstone experience can be one of (a) the second semester of a two-semester elective sequence, (b) a project associated with a course in the student’s major (Computer Science 481/482), or (c) a four- (or more) credit independent study with a significant computing component in the student’s major department. Options (b) and (c) must be preapproved by a computer science advisor.

The point scale for retention of the major/minor applies to all courses in the major/minor. No requirement for the major/minor may be taken satisfactory/unsatisfactory.

Requirements for the Majors in Interdisciplinary Computation

Listed under "Biology," "Environmental Studies," and "Theater and Dance."

Computer Science 151, 231, 251, and two upper-level electives appropriate for the focus area. In addition, courses in the student's focus discipline from one of the approved tracks, listed below, and a capstone experience of at least four credits (Computer Science 491 or 492). Each student will have an advisor in computer science and an advisor in his or her focus department. The advisors will oversee the student's plan of study and capstone project.

Biology Track (without Advanced Placement Biology): Biology 163, 164, 279; 320 or 379; and one elective numbered 200 or above.

Biology Track (with Advanced Placement Biology): Biology 279, 320, 379, and two electives numbered 200 or above.

Environmental Studies Track: Environmental Studies 118, 212, 214, 233, 234, 271, 352, 356; one of 241, 265, 266, 297B, 319, 336, 342, 346; 401, 402.

Theater and Dance Track: Theater and Dance 113 or 114; 135; 171 or two of 115, 116, 117, 119; 281 or 285; 235 or 365.

Course Offerings

CS151fs Computational Thinking An introduction to computational thinking: how we can describe and solve problems using a computer. Using the Python language, students will learn how to write algorithms, manipulate information, and design programs to make computers useful tools. Through lectures, short homework assignments, and weekly programming projects, they will learn about abstraction, how to divide and organize a process into appropriate components, to describe processes in a computer language, and to analyze and understand the behavior of their programs. Students will communicate the results of their work through project reports. *Four credit hours.* **Q** SKRIEN, TAYLOR

[CS157] Digital Media Introduction to computational thinking in the context of digital media. Computational thinking is the basis for describing processes that can be automated on a computer, including image and sound creation, manipulation, and analysis. Using the Python language, students write algorithms, manipulate information, and design programs to make computers tools for creative expression. Through lectures, homework assignments, weekly programming projects and write-ups, they will learn about abstraction, how to divide and organize a process into components, how to describe processes in a computer language, and how to analyze and understand the behavior of their programs. *Four credit hours.* **Q**

CS231f Data Structures and Algorithms Focuses on the common structures used to store data and the standard algorithms for manipulating them. Standard data structures include lists, stacks, queues, trees, heaps, hash tables, and graphs. Standard algorithms include searching, sorting, and traversals. Along with implementation details, students will learn to analyze the time and space efficiency of algorithms and how to select appropriate data structures and algorithms for a specific application. In homework, labs, and programming projects, students will implement their own data structures and make use of existing libraries to solve a variety of computational problems. *Prerequisite:* A grade of C- or higher in Computer Science 151. *Four credit hours.* SKRIEN

CS232s Computer Organization Computer organization focuses on how computers work. Students learn the fundamental hardware components, including storage (RAM, hard disks), input/output mechanisms, and the central processing unit (CPU). They learn how computer components are designed and built on several levels, including the design of the electrical component, machine language, and assembly language. They also learn to program in assembly language for one or more simple computer processors. Students learn primarily through projects where they design digital circuits, design components of a CPU, or write programs in assembly language. *Prerequisite:* Computer Science 151. *Four credit hours.* SKRIEN

CS251s Data Analysis and Visualization Prepares students to apply computational data analysis and visualization approaches to real information from a variety of disciplines and applications. Data visualization is the interactive visual exploration of 2-D and 3-D graphic information using techniques that highlight patterns and relationships. Data analysis incorporates data management, data transformations, statistical analysis, data mining, and machine learning. Through programming projects, students will gain hands-on experience with the fundamentals of data analysis and visualization using data from active research projects at Colby and other institutions. *Prerequisite:* Computer Science 231. *Four credit hours.* TAYLOR

CS269j Computer Game Design Students will learn how to design 2-D computer games using a commercial game engine. Topics include game design, artistic concepts, image manipulation, game scripting, and basic artificial intelligence concepts. Students will work in groups to design and develop a game to be distributed at the end of the term. Each group will make weekly presentations to the class, demonstrating their progress. *Prerequisite:* Computer Science 151. *Three credit hours.* MAXWELL

[CS283] Introductory Bioinformatics An introduction to bioinformatics methods and theory for students with a basic understanding of molecular biology. Topics will include genetic databases, sequence alignment, phylogenetic methods, structure prediction, genome assembly and annotation, protein structures, and gene expression analysis. A research project with a final oral presentation and term paper is required. Previously offered as CS297J (January 2010). *Prerequisite:* Biology 163 or Computer Science 341. *Three credit hours.* **Q**

[CS325] Web Programming The art and science of building dynamic (interactive) websites. Students will learn the fundamentals of the Internet and its HTTP/TCP/IP protocols, HTML and CSS, and how to use them to create well-designed Web pages that follow industry standards. They will learn to program in JavaScript to create client-side dynamic Web pages in PHP or another language to create server-side dynamic Web pages, and in SQL to create, access, and modify a relational database. Finally, they will learn about XML, DOM, and AJAX, and how to use them to add Web 2.0 features to web pages. *Prerequisite:* Computer Science 231. *Four credit hours.*

CS333f Programming Languages A survey of programming languages and paradigms focusing on the design of programming languages and comparing and contrasting different language families, including imperative, object-oriented, functional, and logic paradigms. Topics include syntax, context-free grammars, parsing, semantics, abstract representations of

programming processes and structures, memory management, and exceptions. Students will undertake small programming projects in various languages and more extensive projects in two languages of their choice, presenting the characteristics of their chosen languages to their peers at the end of the term. *Prerequisite:* Computer Science 231. *Four credit hours.* MAXWELL

[CS336] Parallel and Distributed Processing An introduction to the principles and applications of parallel and distributed computing, with an emphasis on parallel computing. Within the context of (1) multi-threaded programming with POSIX threads and (2) MPI programming, we study dead-lock avoidance, load-balancing with appropriate data distribution schemes, basic parallel abstractions such as scan and reduce, parallel sorting algorithms, and performance analysis. Students will learn through a series of programming projects and problem sets. The final project is a significant report analyzing the performance of three sorting algorithms. *Prerequisite:* Computer Science 231 and 232. *Four credit hours.*

CS341f Systems Biology I An introduction to the field of molecular systems biology, which aims to understand the mechanisms underlying complex biological processes. Key to this endeavor is the process of formulating and analyzing mathematical models. Students will learn how to develop, simulate, and analyze ordinary differential equation models of biological systems as well as to read and understand relevant journal articles and perform in-depth analysis of model dynamics. *Prerequisite:* Computer Science 231, and Mathematics 122 or equivalent, and one of the following: Biology 163 or 164; Mathematics 212, 253 or a 300-level course; or any 300-level computer science course. *Four credit hours.* TAYLOR

[CS351] Computer Graphics An introduction to computer graphics covering 2-D graphic primitives, clipping graphic objects to boundaries, linear transformations, creating and representing 3-D objects, converting 3-D models into 2-D images, and rendering complex 3-D scenes made of thousands of polygons. Students will build a comprehensive 3-D rendering engine in sequential weekly projects for which they generate images and develop portfolios of their own work. *Prerequisite:* Computer Science 251. *Four credit hours.*

[CS356] Introduction to Compiler Construction Introduction to the theory, basic techniques, and design of compilers and interpreters of general purpose programming languages; grammars, symbol tables, lexical analysis, semantic analysis, code generation, and optimization. Offered in alternate years. *Prerequisite:* Computer Science 333. *Four credit hours.*

[CS361] Object-Oriented Design Object-oriented design focuses on the art and science of designing programs so that they are reusable, readable, maintainable, extensible, and robust. Students will learn object-oriented design techniques for producing such software, focusing on learning good programming style, object-oriented design principles, and design patterns. Students will also examine case studies of moderately large programs, will learn to use tools such as CRC cards and the UML, and will undertake significant programming projects. *Prerequisite:* Computer Science 231. *Four credit hours.*

CS363s Robotics Addresses the problems of controlling and motivating mechanical devices to act intelligently in dynamic, unpredictable environments. Major topics will include sensing, navigation and control, mapping and localization, robot perception using vision and sonar, and robot kinematics. In addition to short homework assignments, more extensive projects will be undertaken by students using both existing software and implementing their own algorithms on medium-sized mobile robots capable of functioning in human spaces. Projects will focus on enabling the robots to execute tasks, explore, and interact with people and objects in their environment. *Prerequisite:* Computer Science 251. *Four credit hours.* MAXWELL

[CS365] Computer Vision Investigates designing computer programs that extract information from digital images. Major topics include image formation and acquisition, gray-scale and color image processing, image filters, feature detection, texture, object segmentation, classification, recognition, and motion estimation. Students are introduced to classic and contemporary vision techniques with examples for homework and programming assignments drawn from biological and medical imaging, robotics, augmented reality, and digital photography. They will develop a medium-scale vision system using data from active research projects at Colby. *Prerequisite:* Computer Science 251. *Four credit hours.*

CS369j Computer Game Design Design of 2-D computer games using a commercial game engine, for computer science majors. Topics include game design, artistic concepts, image manipulation, game scripting, and basic artificial intelligence concepts. Students will work in groups to design and develop a game to be distributed at the end of the term. Each group will make weekly presentations to the class, demonstrating their progress. Can be repeated once for credit. *Prerequisite:* Computer Science 231. *Three credit hours.* MAXWELL

CS375f Analysis of Algorithms Focuses on classical algorithms in computer science and the analysis of the space and time efficiency of such algorithms as those that sort arrays and lists and search various data structures, including lists, trees, graphs, and strings. All major categories of algorithms are discussed, including iteration, divide and conquer, brute force, exhaustive search, greedy, dynamic programming, and approximation. Unsolvable and intractable problems are also covered, as is the role of NP-completeness. If time permits, some parallel and distributed algorithms will be discussed. Students will learn through problem sets and short programming projects. *Prerequisite:* Computer Science 231. *Four credit hours.* SKRIEN

[CS378] Introduction to the Theory of Computation Focuses on formal languages, automata, computability, complexity classes, and undecidability. Languages discussed include regular languages, context-free languages, and recursively enumerable languages. Both deterministic and non-deterministic forms of the corresponding machines (finite automata, push-down automata, and Turing machines) are also discussed. Unsolvable and intractable problems are addressed, as is the role of NP-completeness. Students will learn through problem sets and short programming projects. *Prerequisite:* Computer Science

231 and either Mathematics 274 or 275. *Four credit hours.*

CS441s Systems Biology II The application of principles learned in Systems Biology I to a particular biological system. Students will work in interdisciplinary teams to complete a project focusing on one biological system and one or more mathematical models of this system. Involves reading journal articles, designing and running numerical experiments, analyzing results, and presenting challenges and results. Culminates in both a poster presentation and a comprehensive journal article-styled report and oral presentation. *Prerequisite:* Computer Science 341. *Four credit hours.* **N** TAYLOR

[CS451] Advanced Computer Graphics Advanced topics in computer graphics for computer science majors. Focuses on advanced algorithms for rendering both hyper-photorealistic and non-photorealistic images of objects and scenes. Topics will include ray tracing, radiosity and other global illumination methods, animation, motion capture and mapping, modeling unique materials, modeling painting and drawing techniques, and other topics selected by students. Each week students will prepare written summaries and critiques of technical papers in computer graphics. Programming projects will include OpenGL, a photorealistic project, a non-photorealistic project, and a final project of the student's choice. *Prerequisite:* Computer Science 351. *Four credit hours.*

[CS461] Object-Oriented Software Systems Students will learn how to design and implement a significant software project that is robust, maintainable, extensible, and modular, building on their experience in Computer Science 361. The focus will vary from year to year. For example, students may gather specifications and then analyze, design, and implement a business application or dynamic website following standard software engineering practices. Students will learn through creating, implementing, and refining their own software designs in an iterative design process. *Prerequisite:* Computer Science 361. *Four credit hours.*

CS481f, 482s Minor Capstone Independent project and capstone experience for minors. Taken in tandem with a course in the student's major to develop a computing project in consultation with his or her computer science advisor that relates to or extends a topic from the related course. *Three or four credit hours.* FACULTY

CS483Jj Honors Research in Computer Science *Noncredit.* MAXWELL

CS483f, 484s Honors Research in Computer Science The independent study component of the honors program in computer science. *Prerequisite:* Permission of the instructor and admission to the honors program. *Three or four credit hours.* FACULTY

CS491f, 492s Independent Study Independent study in an area of computer science of particular interest to the student. *Prerequisite:* Permission of the instructor. *One to four credit hours.* FACULTY