Computer Science (CSCI)

CSCI 101 Introduction to Computer Science (4)
An introductory survey of computer science designed for liberal arts students, including such topics as machine architecture, language translation, artificial intelligence, and noncomputability.

CSCI 157 Introduction to Modeling and Programming (4)
An introduction to creative modeling of both natural and virtual worlds, in which students gain understanding of human interaction with computing devices as well as the expertise needed for further course work in computer science. Lab experiences using the explicit notation of a programming language reinforce the application of abstractions while affording practice in algorithmic problem solving and relevant theory.

CSCI 257 Object-Oriented Programming (4)
Software design and development using object-oriented programming techniques. Topics include abstract data types and their implementation using classes, design methodologies, recursion, elementary data structures, and implementation of larger programs. *Prerequisite: CSCI 157.*

CSCI 270 Computer Systems and Organization (4)
This course explores levels of abstraction in computer systems, processors and related hardware components, instruction sets, program execution, and process management. *Prerequisite: CSCI 157.*

CSCI 276 Multimedia Programming and Design (4)
An introduction to object-oriented programming techniques that underlie the creation, manipulation, and transmission of digital media, including digital photography, audio, and video. Topics include scaling and transforming pictures, sound waveform visualization and manipulation, MIDI, chromakey, frame-based animation, and compression, encoding, and transmission of digital media over the Internet. *Prerequisite: CSCI 157.*

CSCI 277 Modeling Physical Systems (4)
This course introduces the tools and techniques used to model natural systems in software. Techniques studied are important not only in scientific programs, but also in graphics, game engines, and animation; these include vectors, particle systems, fractals, and autonomous agents. Students will apply these to abstractions of physical systems such as predator-prey models, flocking, cellular automata, genetic algorithms, and artificial neural networks. *Prerequisite: CSCI 157.*

CSCI 284 Database Design with Web Applications (4)
This course provides students with a working knowledge of the power and potential of modern networked databases as well as of common uses and abuses. Students receive hands-on experience with open source development tools, which are widely used for building and placing databases on the web. Database development is explored, from conceptual elaboration through design and implementation, and interview techniques for effective database design are considered. Programming techniques are introduced for building, maintaining, accessing, interacting, and protecting the information in large data depositories. Discussions include consideration of concerns driving policy decisions for amassing and managing sensitive, and sometimes dangerous, information collections. *Prerequisite: CSCI 157.*

CSCI 286 Computer Mapping and Geolocation (4)
An introduction to computer location algorithms and networks with an emphasis on fundamental principles, this course provides basic understanding of location services and underlying technologies, including comparisons of popular mapping and geolocation programming technologies appropriate for web and mobile platforms. *Prerequisite: CSCI 257.*

CSCI 290 Data Mining (4)
Data mining is the automated analysis of large quantities of data to extract previously unknown patterns such as clusters, anomalies, relationships, and dependencies. As large columns of data accumulate, techniques are needed to make sense of the stored information and predict future trends. Data mining applications have become important in fields such as finance, healthcare, manufacturing, and marketing. This course introduces students to the principal ideas in statistical learning, including areas such as classification, clustering, and data extraction. Along the way, students develop problem-solving skills and an understanding of programming techniques and data structures. *Prerequisite: CSCI 257 or CSCI 284.*

CSCI 320 Data Structures and Algorithms (4)
The design, implementation, and application of data structures and their algorithms, including stacks, queues, linked structures, trees and graphs. Other topics include elementary searching and sorting algorithms, pattern matching, and runtime performance analysis. *Prerequisite: CSCI 257 and MATH 215.*

CSCI 326 Functional Programming (4)
Data abstraction and data-driven recursion, higher-order functions, and the management of state. An exploration of the functional paradigm in the context of concurrent and distributed computing. *Prerequisite: CSCI 257.*

CSCI 336 Algorithmic Design and Analysis (4)
Systematic study of the correctness and complexity of algorithms, including design strategies, proof techniques, asymptotic bounds and recurrence relations for analyzing recursive algorithms. NP-complete and intractable problems. Case studies based on searching, sorting, numeric and graph algorithms; advanced tree and graph structures, and symbol tables. *Prerequisite: CSCI 320 and MATH 215.*

CSCI 352 Human-Robot Interaction (4)
This course focuses on the emerging field of human-robot interaction, bringing together research and application of methodology from robotics, human factors, human-computer interaction, interaction design, cognitive psychology, education, and other fields to enable robots to have more natural and more rewarding interactions with humans throughout their spheres of functioning. This course is a combination of state-of-art reading and discussions, focused exercises and problem-solving sessions in human-robot interaction, and a group project that includes the design, development, and evaluation of a human-robot interaction system. *Prerequisite: CSCI 257.*

CSCI 356 Artificial Intelligence (4)
Knowledge representation, expert systems, natural language processing, computer vision, machine learning, game playing, cognition. *Prerequisite: CSCI 257 and MATH 215.*

CSCI 360 Principles of Interactive Computer Graphics (4)
Introduction to interactive computer graphics including 2D and 3D viewing, clipping, hidden line/surface removal, shading, interaction handling, geometrical transformations, projections, and hierarchical data structures. Brief introductions to related and dependent fields of physically-based modeling and scientific visualization will be included. *Prerequisite: CSCI 257 and MATH 215.*

CSCI 370 Computer Architecture (4)
Advanced processor design including pipelining, speculative execution, hyperthreading and their effects on program execution. The memory hierarchy and virtual memory design. *Prerequisite: CSCI 257 and CSCI 270.*

CSCI 376 Programming Languages (4)
Imperative, object-oriented, declarative, and functional programming language paradigms. *Prerequisite: CSCI 257 and MATH 215.*

CSCI 415 Special Topics (2 or 4)
This course explores emerging concepts, methodologies, and applications to other disciplines in the field of computer science. This course may be repeated for credit when the topic differs. *Prerequisite: CSCI 257.*

CSCI 428 Operating Systems (4)
Process management, memory management, processor scheduling, file systems, concurrent programming, distributed processing, security. *Prerequisite: (CSCI 270 or CSCI 370) and MATH 215.*

CSCI 444 Independent Study (2 or 4)
An opportunity for advanced students to pursue topics of special interest. *Prerequisite: Instructor prerequisite override required.*

CSCI 457 Software Engineering (4)
The software systems we often find the most useful and magical are also the largest and most complex to build and understand. To be reliable, maintainable, and secure, these systems must be built according to disciplined and well-founded methods. This course examines these methods, both large-scale (defining requirements, system design, architecture patterns, software processes, etc.) and small-scale (version control, testing, benchmarking, code review, etc.). Students will work in small teams to construct production-quality software. *Prerequisite: CSCI 320.*

CSCI 460 Senior Capstone (2)
In the first semester, students work with faculty mentors to design and begin working on substantial projects that require integrating and building on what they have learned in their other courses. In the second semester, students complete these projects and present their results in a public venue. May be repeated once for credit. *Open only to seniors pursuing majors in computer science.*