

# COMPUTER SCIENCE (CS)

---

# Course numbers with the # symbol included (e.g. #400) have not been taught in the last 3 years.

## CS 800 - Internship

**Credits:** 1

Provides an opportunity to apply academic experience in settings associated with future professional employment. A written proposal for the internship must be approved by the department chair. The proposal must specify what the student will learn from the internship, why the student is properly prepared for the internship, and what supervision will be available to the student during the internship. A mid-semester report and a final report are required. Permission required. Computer Science majors only. May be repeated up to a maximum of 3 credits. Cr/F.

## CS 812 - Compiler Design

**Credits:** 3

Formal languages and formal techniques for syntax analysis and parsing; organization of the compiler and its data structures; code generation. LL and LR parsing; automatic generation of scanners and parsers from high-level descriptions. Implementation of features from imperative and object-oriented languages. Students are required to design and implement a compiler for a simple language. Prereq: Machine Organization.

## CS 819 - Advanced Programming with Object-Oriented Design

**Credits:** 3

Advanced problem solving using software design, development and testing techniques that follow the software development life cycle. Object-oriented programming and design. Advanced data structures and algorithm analysis. Prereq: strong prog. skills, exp. with C/C++, match org.

## CS 820 - Operating System Programming

**Credits:** 3

Detailed discussion of operating system concepts and features. Practical examples and exercises that utilize advanced operating system features, including inter-process communication, synchronization, client-server communication, shared memory, threads, remote procedure calls, and device-level I/O. Discussion of POSIX 1003.1 Part I Standards. Prereq: operating system fundamentals or equivalent.

## CS 823 - Performance Evaluation of Computer Systems

**Credits:** 3

This class introduces the main concepts, techniques, and tools needed to evaluate the performance of computer systems under various configurations and workloads. The techniques allow one to perform capacity planning based on quality of service requirements of users and workload characteristics. The course is mainly based on the use of analytic queuing network models of computer systems. The performance techniques are applied to study the performance of centralized, distributed, parallel, and client/server systems. The course also discusses performance measuring tools for operating systems such as Unix and Windows NT. Prereq: operating systems fundamentals or equivalent.

## CS 825 - Computer Networks

**Credits:** 3

Introduction to fundamental concepts of computer networks and exploration of widely-used networking technologies. Topics include principles of congestion and error control; network routing; local, wireless and access networks; application protocol design; and network programming. In-depth discussion of the Internet suite of protocols.

## CS 830 - Introduction to Artificial Intelligence

**Credits:** 3

In-depth introduction to artificial intelligence concentrating on aspects of intelligent problem-solving. Topics include situated agents, advanced search techniques, knowledge representations, logical reasoning techniques, reasoning under uncertainty, advanced planning and control, and learning. Prereq: data structures.

## CS 833 - Mobile Robotics

**Credits:** 3

An introduction to the foundational theory and practices in mobile robotics. Topics include Kinematics of wheeled mobile robots. Seniors for mobile robots, robot navigation and perception, robot vision, localization and mapping of mobile robots. Hands-on experience directed towards implementation with a real robot. Prereq: Programming or permission of instructor.

## CS 835 - Introduction to Parallel and Distributed Programming

**Credits:** 3

Programming with multiple processes and threads on distributed and parallel computer systems. Introduces programming tools and techniques for building applications on such platforms. Course requirements consist primarily of programming assignments. Prereq: Undergraduate course in operating systems fundamentals and computer organization; or permission.

## CS 845 - Formal Specification and Verification of Software Systems

**Credits:** 3

Course focuses on the formal specification and verification of reactive systems, most notably concurrent and distributed systems. Topics relevant to these systems, such as non-determinism, safety and liveness properties, asynchronous communication or compositional reasoning, are discussed. We rely on a notation (T LA+, the Temporal Logic of Actions) and a support tool (TLC, the TLA+ Model Checker). Prereq: Students are expected to be knowledgeable in logic and to be able to write symbolic proofs in predicate calculus. A basic understanding of the notions of assertion, precondition, and post-condition is also assumed.

## CS 850 - Machine Learning

**Credits:** 3

An introduction to fundamental concepts and common methods in machine learning. In addition to theoretical topics, the course involves hands-on experience in making predictions using synthetic and real-world datasets. Prereq: Statistics, Programming or permission of instructor.

## CS 853 - Information Retrieval

**Credits:** 3

Fundamental algorithms and techniques for text processing and text-based information retrieval systems. Topics include how to build an end-to-end information retrieval system, such as a Web search engine. Prereq: Data Structures.

## CS 854 - System/Software Test and Evaluation

**Credits:** 3

This course identifies an integrated software test and evaluation process framework that emphasizes a "systems engineering" approach: the validation and viability of customer/user needs statements, verification of system design, full exercise of developmental testing, system integration/test dovetailing on the prior validations, plus evaluation of system quality attributes. This system engineering approach is intended to contain major problems, including interface issues, to phases preceding system test.

**CS 858 - Algorithms****Credits:** 3

An introduction to important concepts in the design and analysis of algorithms and data structures, including implementation, complexity, analysis, and proofs of correctness. Prereq: understanding of basic data structures, familiarity with proof methods and basic concepts from discrete mathematics and the ability to program with recursion.

**CS 860 - Introduction to Human-Computer Interaction****Credits:** 3

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. Prereq: operating systems fundamentals.

**CS 867 - Interactive Data Visualization****Credits:** 3

Detailed discussion of how an understanding of human perception can help us design better interactive displays of data. Topics include: color, space perception, object perception and interactive techniques. Students write interactive programs, give presentations and undertake a project designing and evaluating a novel display technique. Prereq: Introductory level C or C++ programming course. (Also listed as OE 867.)

**CS 870 - Computer Graphics****Credits:** 3

Input-output and representation of pictures from hardware and software points of view; interactive techniques and their applications; three-dimensional image synthesis techniques. Prereq: data structures.

**CS 871 - Web Programming Paradigms****Credits:** 3

In this course you will learn languages to program the Web. Languages integrated into browsers, like Javascript, and languages invoked on the server, like Ruby. You will also learn about frameworks, like Rails, and various techniques used to support the programming process. In addition, you will learn languages you will need to create, modify and process Web documents. Although we will learn how to read and write in these languages, our primary goal will be on understanding how the design of these multi-paradigm dynamic languages support the process of developing Web applications. Prereq: programming language concepts or permission.

**CS 875 - Database Systems****Credits:** 3

Database analysis, design, and implementation. Focus on the relational model. Data description and manipulation languages, schema design and normalization, file and index organizations, data integrity and reliability. Usage of selected DBMS. Prereq: data structures; mathematical proof.

**CS 880 - Topics****Credits:** 1-4

Material not normally covered in regular course offerings. May be repeated.

**CS 898 - Master's Project****Credits:** 3**CS 899 - Master's Thesis****Credits:** 1-6

May be repeated up to a maximum of 6 credits. Cr/F.

**CS 900 - Graduate Seminar****Credits:** 1

Regularly scheduled seminars presented by outside speakers, UNH faculty, and graduate students. Topics include reports of research ideas, progress, and results. Cr/F.

**CS 920 - Advanced Operating Systems****Credits:** 3

This course covers techniques for formally analyzing various fundamental concepts and mechanisms which form the basis of the design of advanced operating systems, including distributed, database, and multiprocessor operating systems. Topics covered include synchronization, mutual exclusion, distributed algorithms, security, fault-tolerance, and distributed resource management. Prereq: operating system fundamentals or equivalent.

**CS 925 - Advanced Computer Networks****Credits:** 3

Design and analysis of computer networks. Modeling and performance evaluation, queuing theory applied to computer networks. Traffic flow management and error control. Routing algorithms and protocols. Switch and router architectures. Selected issues in high-speed network design. Optical networks. Prereq: CS 825 or equivalent.

**CS #931 - Planning for Robots****Credits:** 3

Students read research papers and perform a research project pertaining to planning algorithms for autonomous robots. The main student deliverable is a conference-style paper reporting on the project. Projects may be done in teams and can be demonstrated on physical or simulated robots. Prior exposure to algorithms and artificial intelligence is helpful but not required. Graduate students from other departments are welcome but should consult with the instructor before enrolling.

**CS 980 - Advanced Topics****Credits:** 3**CS 981 - Advanced Topics in Database Systems****Credits:** 3**CS 986 - Advanced Topics in Formal Specification and Verification****Credits:** 3

This course explores more thoroughly some of the material introduced in CS 845. It focuses on concurrent and reactive systems and on temporal logics. Topics include safety and liveness properties, asynchronous communication, and compositional reasoning. Support tools, like interactive theorem provers and model-checkers, are presented and used in class. Prereq: introduction to formal specification and verification. May be repeated up to a maximum of 6 credits.

**CS 988 - Advanced Topics in Computer Graphics****Credits:** 3**CS 989 - Advanced Topics in Algorithms****Credits:** 3**CS 998 - Independent Study****Credits:** 1-6**CS 999 - Doctoral Research****Credits:** 0

Cr/F.