

# COMPUTER SCIENCE (CS)

---

## CS-1000 Introduction to Computer Science (3 Credits)

This course is an introduction to computer science and the art of programming with or without prior programming experience. As an entry-level course, it focuses on teaching students how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structure, encapsulation, resource management, security, software engineering, and web development.

**Typically offered:** All Sessions

## CS-2000 Discrete Structures (3 Credits)

**Requisite(s):** CS 1000 or IT 1104, MAT 2202

This course covers fundamental ideas from discrete mathematics essentials for computer science students. It focuses on topics that are foundational for future courses including algorithms, artificial intelligence, programming languages, theoretical computer science, computer systems, cryptography, networks, computer/network security, databases, and compilers.

**Typically offered:** All Sessions

## CS-3000 Software Engineering (3 Credits)

**Requisite(s):** IT-2105, IT-3510

This course presents modern software engineering techniques and examines the software life-cycle, including software specification, design, implementation, testing, and maintenance. The course evaluates past and current trends in software development practices including agile software development, methods such as Extreme Programming (XP), Agile Modeling (AM), Scrum DevOps, and Site Reliability Engineering (SRE). This course gives an overview of methods and techniques used in agile software processes, contrasts agile approaches with traditional software development methods, and discusses the advantages of both classes of methodologies. The course also explores ethical and legal issues related to software development, such as intellectual property rights, data privacy, and security concerns.

**Typically offered:** All Sessions

## CS-4998 Computer Science Capstone Project (3 Credits)

**Requisite(s):** IT-3301, CS-3000

The goal of this course is to provide students with an opportunity to integrate what they have learned in earlier computer science courses, deepen their understanding of that material, extend their area of knowledge, and apply their knowledge and skills in a realistic simulation of professional experience. The end result must be a substantial software product.

**Typically offered:** All Sessions

## CS-6001 Foundations of Programming, Data Structures & Algorithms (3 Credits)

This course explores methods of computer programming—the algorithmic aspects of computer science and the theoretical constructs common to all high-level programming languages.

**Typically offered:** All Sessions

## CS-6002 Computer Organization and Architecture (3 Credits)

**Requisite(s):** CS-6001

This course focuses on the principles of computer organization and architecture concepts. The course emphasizes performance and cost analysis, instruction set design, pipelining, memory technology, memory hierarchy, virtual memory management, and I/O systems. Technical writing skills are emphasized.

**Typically offered:** All Sessions

## CS-6003 Design and Analysis of Algorithms (3 Credits)

**Requisite(s):** CS-6001

This course introduces methods for the design and analysis of efficient algorithms emphasizing methods useful in practice. Different algorithms for a given computational task are presented and their relative merits are evaluated based on performance measures. The following computational problems are discussed: sorting, searching, elements of dynamic programming and greedy algorithms, advanced data structures, graph algorithms (shortest path, spanning tree, tree traversals), string matching, elements of computational geometry, and NP-completeness.

**Typically offered:** All Sessions

## CS-6004 Principles of Programming Languages (3 Credits)

**Requisite(s):** CS-6003

A programming language provides a way to express an algorithm and computational structure. Different languages have produced distinct styles of expression or paradigms. This course explores some of these important paradigms. Students will develop practical competency in languages representing distinct paradigms while also exploring criteria for language selection. Topics will include object-oriented programming, functional programming, logic programming, procedural programming, and programming for concurrency and distributed computing.

**Typically offered:** All Sessions

## CS-6006 Software Engineering (3 Credits)

**Requisite(s):** CS-6002, CS-6003

Software engineering is the branch of computer science that creates practical, cost-effective solutions to computing and information processing problems. This course covers the fundamentals of software engineering including understanding system requirements, finding appropriate engineering compromises, effective methods of design, coding, and testing, team software development, and the application of engineering tools.

**Typically offered:** All Sessions

## CS-6998 Capstone Project in Computer Science (3 Credits)

**Requisite(s):** CS-6006

This course applies the foundational concepts of computer science through a comprehensive project, chosen by the student following their career goals. Students will be required to plan, research and develop in conjunction with the student's interests such as cyber security, information technology, or programming. The result must be a substantial software product.

**Typically offered:** All Sessions