COMPUTER SCIENCE MAJOR

Suggested Course Sequence

3rd-Class Year  2nd-Class Year  1st-Class Year
Chem 200      Aero Engr 315     Academy Opt/Comp Sci Opt
Comp Sci 210   Beh Sci 310      Astro Engr 310
Comp Sci 220   Biology 315      Comp Sci 426
Comp Sci 351   Comp Sci 359     Comp Sci 453
Econ 201      Comp Sci 364      Comp Sci 454
English 211   Comp Sci 380      Comp Sci Opt
Engr Mech 220  Comp Sci 467      Comp Sci Opt
Law 220       Comp Sci 483      English 411
Math 340      History 300       ECE 315
MSS 200       Math 356         Mgt 400
Physics 215   Math Opt         MSS 415/416
Pol Sci 211   Philos 310       Soc Sci 412

COMPUTER SCIENCE (Comp Sci)

Offered by the Department of Computer Science (DFCS).

Comp Sci 110. Introduction to Computing. Introduction to principles, applications, capabilities and limitations of computer systems. Topics include computer hardware, algorithms, information representation, networks, computer security, computers and society, system and application software, and computer programming. Students learn how to use their own computers more effectively.

Comp Sci 110S. Introduction to Computer Science for Scholars. An overview of the great ideas in computing. We will investigate the “big questions” in computing and humanity’s best attempts to answer them, by reading the original writings of significant contributors to the field. Problems explored include “What is computing?”, “What can and cannot be computed?”, “How can a computer be constructed?”, “How can we best express our ideas to computers?”, and “How long does it take to compute something?”. Particular emphasis is placed on reading original materials, learning by doing, and on the military and social consequences of computing. Programming experience is not required.

Comp Sci 210. Introduction to Programming. Introduces the fundamentals of software development as a foundation for a more advanced study of computer science. Topics include programming constructs, appropriate use of fundamental data types and class libraries, error handling exceptions, problem-solving strategies, algorithms, data structures, recursion and object-oriented concepts. Considerable attention is devoted to developing effective software engineering practices, emphasizing design, documentation, encapsulation, procedural abstraction, testing, debugging and software reuse.
**Comp Sci 211. Introduction to Programming for Scientists and Engineers.** Introduces and applies the fundamental techniques of scientific programming as a foundation for solving scientific and engineering problems using computers. Topics include programming constructs, problem-solving strategies, testing and debugging, effective use of pre-defined functions and vector and matrix operations, with a focus on engineering and scientific applications. Students learn a programming language and development environment that is suited for science and engineering disciplines.

**Comp Sci 212. Introduction to Programming for Analysts.** Introduces the fundamental techniques of software development as a foundation for solving analytical problems using computers. Topics include programming constructs, problem-solving strategies, algorithms, and data structures, with a focus on analysis applications. Considerable attention is devoted to developing effective software engineering practices, emphasizing design, decomposition, encapsulation, modularity, testing, debugging, and software reuse. Cadets will learn a programming language and development environment that is widely used within the analytic discipline.

**Comp Sci 220. Data Abstraction.** Continues the introduction of software development, with a particular focus on the ideas of data abstraction, object-oriented programming and fundamental data structures. Topics include recursion, computational complexity, event-driven programming, graphical user interface design and implementation, and fundamental computing algorithms.

**Comp Sci 223. Data Structures and Systems Programming.** Continues the introduction of software development, with a particular focus on the ideas of data abstraction, data structures, and memory management. Topics include recursion, computational complexity, memory management, reference semantics, and fundamental computing algorithms.

**Comp Sci 351. Computer Organization and Architecture.** Introduces basic computer logic systems, major types of computing system organizations, and machine and assembly language programming. Topics include digital logic, processor architecture, data representation, memory architecture, performance analysis, computer arithmetic, pipelining, and multiprocessing.

**Comp Sci 359. Programming Paradigms.** Applied course studying four different programming paradigms. Imperative, Object-Oriented, Functional and Logic programming paradigms are covered. Programming languages and specific language constructs supporting the four paradigms are covered, but the emphasis is on how to think about programming in each paradigm. At least one programming project is assigned for each of the four paradigms.

**Comp Sci 360. Software Reverse Engineering.** This lab-based course focuses on the analytical process of interpreting and deconstructing program executable files, viewed in low-level assembly language. Cadets will learn how to interpret the purpose and intent of malicious software, how to overcome anti-reverse engineering techniques, and defenses to exploit strategies that may be employed in order to alter the functionality of a target executable.

**Comp Sci 362. Computer Simulation.** Introduction to modeling and simulation. Topics include principles of computer simulation methodologies; simulation data analysis and input design; simulation development; analysis of results; and verification of system design, implementation, and assumptions. The course includes the development of small conceptual simulations, statistical analysis of data sets, and a group project involving the simulation and analysis of real-world systems.

**Comp Sci 364. Information Storage and Retrieval.** Introduction to the basic concepts of database and information storage systems. Topics include data models, database design theory, database performance, transaction processing, web database interaction, techniques for handling large volumes of data and contemporary database issues. Hands-on projects emphasize basic database and information storage and retrieval techniques.
Comp Sci 380. Design and Analysis of Algorithms. Advanced design and analysis of algorithms used in modern computing systems. Topics include analysis of algorithms, basic structures, advanced abstract data types, recursion, computability and complexity. Problem solving and analytical skills are improved by examining the application of abstract data types to several problem domains with an emphasis on the impact of design decisions on algorithm performance. Concepts are reinforced by several programming exercises.

Comp Sci 385. Software Dev for Mobile Devices. Introduces the design and development of applications (apps) for mobile devices. Mobile device topics include runtime environments; human interface guidelines; display optimization; device orientation, location, and movement; user gesture handling; interactions metaphors; user interface paradigms; user content expectations; user-centric terminology; data reach-back; and multitasking. Hands-on projects emphasize the unique aspects of software development to meet device-specific user expectations and the unique aspects of mobile device sensors, input and output modes, and limitations.

Comp Sci 426. Languages and Machines. Students learn the theoretical foundations of computer science and apply these concepts to appropriate stages in compiler implementation. Topics include finite automata, formal language theory, grammars, scanners, parsing techniques, code generation, symbol tables and run-time storage allocation. Students design and implement a syntax-directed compiler for a high-order programming language.

Comp Sci 431. Cryptography. Introduces the principles of cryptography and number theory. Topics include: primes, random numbers, modular arithmetic and discrete logarithms, symmetric encryption, public key encryption, key management, hash functions, digital signatures, authentication protocols and protocols for secure electronic commerce. Elliptic curves and quantum cryptography will also be introduced.

Comp Sci 438. Advanced Computer and Network Security I. Cadets are immersed in technical aspects of computer and network security. Emphasis is on the theory behind common computer and network exploitation methods and how security fundamentals help thwart attacks. Topics include secure design principles, network address security, email security, file security, web security, remote exploits, access and flow controls, security models, and encryption. Cadets complete a security-based final project.

Comp Sci 453. Software Engineering I. This is the first course of a two-semester capstone software engineering sequence. Students are introduced to and apply modern software development techniques used in team-based software systems development. Topics include software development process lifecycles, software project management, configuration management, quality assurance, requirements elicitation, system analysis, system design, system implementation, system integration, and system testing. Students begin work on a two-semester team-based software development project.

Comp Sci 454. Software Engineering II. This is the second course of a two-semester capstone software engineering sequence. Students, working as a team, continue to apply modern software development processes to evolve a real-life software system. Teams conduct and evaluate beta-tests, conduct acceptance testing, plan for production release, and plan for long-term software maintenance and support. Students complete work on a two-semester software development project.

Comp Sci 467. Computer Networks. Examination of modern data communications systems and related security issues. Topics include the TCP/IP reference model, data transmission theory, network design issues, internetworking, routing, network protocols, implementation of networks, and communications security.

Comp Sci 468. Advanced Computer and Network Security II. Cadets study and apply advanced computer and network security topics. Topics include boundary protection, domain controls, attack sequencing and timing, penetration testing, social engineering, and disaster recovery. Cadets participate in a team-based cyber security exercise.
**Comp Sci 471. Artificial Intelligence.** Introduction to major subjects and research areas in artificial intelligence (AI). Topics include: problem solving techniques, knowledge representation, machine learning, heuristic programming, AI languages, expert systems, natural language understanding, computer vision, pattern recognition, robotics and societal impacts. Also explores current and projected uses of AI in the Air Force.

**Comp Sci 474. Computer Graphics.** Basic concepts of interactive computer graphics including both vector and raster graphics. Topics include mathematics of 2-dimensional and 3-dimensional geometric transformations, interactive techniques, graphics hardware architectures, graphic algorithms and realism in computer-generated images. Includes several computer projects.

**Comp Sci 476. Computer and Network Forensics.** Covers digital information storage and recovery techniques including network, live-memory, and static memory forensic retrieval and analysis. Topics include data capture, file memory recovery, internet browser forensics, network traffic reconstruction and analysis, and live-memory examination.

**Comp Sci 483. Operating Systems.** Examines the design and implementation of programs that manage hardware resources and provide abstract interfaces for hardware control. Topics include resource allocation, synchronization primitives, virtual memory, information protection, performance measurement, I/O sub-subsystems and distributed computing.

**Comp Sci 495. Special Topics.** Selected topics in computer science.

**Comp Sci 496. Computer Science Seminar.** Advanced topics in computer science. Students participate in and lead discussions on significant issues in current computer science research as well as key historical developments.

**Comp Sci 499. Independent Study.** Individual study and research supervised by a faculty member. Topic established with the department head.