The mission of Fairfield University is to educate its students through a variety of scholarly and professional disciplines. It offers opportunities for individual and common reflection, and it provides training in such essential human skills as analysis, synthesis, and communication. Computational thinking and processes permeate our daily lives, revolutionizing our understanding of the natural world, our tools, and of ourselves. Knowledge of computer science has become highly valued in such diverse fields as psychology, biology, and even philosophy. A degree in Computer Science gives one both marketable skills and intellectual breadth that can be applied to many career choices. At Fairfield, students can pursue multiple degree options as follows: a BA degree in Computer Science, a BS degree in Computer Science, a double major in Computer Science and Mathematics, and a minor in Computer Science that makes a strong addition to one’s resume. The BS in Computer Science is preparing for accreditation by the Computing Accreditation Commission of ABET (http://www.abet.org). Additionally, the BS in Computer Science can be continued with a specialization in Software Engineering through a five-year Bachelor and Master dual-degree track, and the BA in Computer Science can be continued with a specialization in Applied Data Science through a five-year Bachelor and Master dual-degree.

Fairfield recognizes that learning is a life-long process and sees the education which it provides as the foundation upon which its students may continue to build within their chosen areas of scholarly study or professional development.

### Programs

- Computer Science Major (BA) (http://catalog.fairfield.edu/undergraduate/engineering/computer-science-software-engineering/computer-science)
  - Concentration in Computer Engineering
  - Concentration in Software Engineering
- Computer Science Major (BS) (http://catalog.fairfield.edu/undergraduate/engineering/computer-science-software-engineering/computer-science-accredited)
  - Concentration in Computer Engineering
  - Concentration in Software Engineering
- Computer Science and Mathematics Double Major (http://catalog.fairfield.edu/undergraduate/engineering/computer-science-software-engineering/double-major-computer-science-mathematics)
  - Computer Science Minor (http://catalog.fairfield.edu/undergraduate/engineering/computer-science-software-engineering/computer-science-minor)
- Applied Data Science Five-Year Dual Degree Bachelor and Master of Science Program (http://catalog.fairfield.edu/undergraduate/engineering/computer-science-software-engineering/5-year-applied-data-science)
  - Software Engineering Five-Year Dual Degree Bachelor and Master of Science Program (http://catalog.fairfield.edu/undergraduate/engineering/computer-science-software-engineering/dual-degree-curriculum)

## Courses

### Computer Science

**CS 0101 Introduction to Computing**

3 Credits

In this course, students learn computational problem-solving techniques through the process of design, implementation, testing, and documentation using the programming language Python. The main ideas of computing are explored and students learn the most essential information about computers and technology in today's digital world and the latest computing trends and skills. Students will gain an understanding of the breadth of computing as a discipline and how it exists in the world by identifying computing applications in society and exposing them to a variety of computing topics.

**CS 0131 Fundamentals of Programming**

3 Credits

Attributes: BUEL Business Elective, ENPC Digital Journalism Production Component

This course introduces programming constructs and techniques in a logical progression beginning with small problems and basic algorithms through larger scale programs and design. While not an object oriented course, Classes and Objects are used in an ancillary capacity while working on broader topics of software architecture. Complete programs will be designed, coded, and debugged in both Java and the C programming language, developing skills necessary to work with more complex software systems.

**CS 0152 Introduction to Computer Game Modeling**

3 Credits

This is an introductory computer games modeling course which examines the basics of computer game design and visual effects. Students will use graphics software modeling packages to create characters and visual effects, and to develop a computer game idea, including storyline and plots. Basic programming techniques may also be taught.

**CS 0201 Technical Skills for Liberal Arts Majors**

3 Credits

Students today will need skills in a variety of computer programs and software applications. This course is designed for those in the liberal arts (humanities and social or behavioral sciences) who will need these technical skills for their future workplace. Specific skill sets include software for word processing, spreadsheet, presentation, conferencing, web page coding, and web page design; other software may be included. The course serves those students going into business, publishing, non-profit careers, and a variety of other enterprises for which the liberal arts training must be augmented with specific cyber-related technical competence. Open to students with majors in the humanities or social/behavioral sciences only.

**CS 0231 Programming Workshop**

3 Credits

Corequisite: CS 0231L.

**Prerequisite: CS 0131.**

This course covers advanced programming concepts in one or more current programming languages, including syntax and theories. It prepares students for adapting to various programming environments and coding in an efficient manner. Lab work will accompany the course.

**CS 0231L Programming Workshop Lab**

1 Credit

Corequisite: CS 0231.

This lab accompanies the Programming Workshop course for hands-on practice with course concepts.
This course covers abstract data structures such as queues, stacks, heaps, linked lists, trees, graphs, hash tables and sorting. Students apply data structure concept in advanced programming.

This lab accompanies the Data Structures course for hands-on practice with course concepts.

This sophomore clinic provides faculty guidance and supervision beyond the scope of existing courses. Students learn how to develop and structure their deliverables, as well as how to use computer science tools in the context of real-world or research projects.

This course introduces the student to developing applications for use on the World Wide Web. Students learn basic n-tier concepts for designing distributed applications and gain hands on experience through the construction of Web-based applications. The course covers concepts that allow communication over the Web. This includes designing and authoring Web pages, markup languages, the client-side document object model, usability, search engine optimization, and client-side dynamic Web pages.

This project-oriented course examines the fundamental aspects of mobile computing, application architecture, and mobile application design and development. Students will learn application development on the Android platform. Students will complete a hands-on project building a prototype mobile application. Topics include user interface design and building, input and data handling, and network techniques and GPS and motion sensing. Students are expected to work on a project that produces a professional-quality mobile application. Projects will be deployed in real-world applications.

This course is intended for individuals who need an understanding of the client-server environment, with any emphasis on network security. The OSI Model, network concepts, and network architecture are discussed. The components that make up a network, including cabling, wiring hubs, file servers, bridges, switches, routers, network interface cards, network operating systems, and network software and hardware configurations are discussed. Network architectural concepts, wide area networks, remote access, and segmentation are discussed. Operating systems will be discussed and demonstrated. Featured is the seven-layer OSI model, the foundation of today’s communication protocols. Students will work with various security protocols and configure routers and switches with security methods.

This course is designed to benefit computational and experimental biologists to understand the principles of analyzing biological data, building models and testing hypotheses using computer science paradigms. Students will learn how to build computational tools that are used to analyze DNA content, identify protein binding patterns, compare sequences, and discover variation within genomes.

This course will provide a practical introduction to analysis of biological and biomedical data. Basic statistical and machine learning techniques will be covered, including descriptive statistics, linear regression, non-linear regression, classification/prediction, and biomedical data visualization. Emphasis will be on how to choose appropriate data analysis models and how to assess statistical significance. This course will benefit data scientists to apply data science techniques to analyze biomedical data or clinical data. In addition, this course is also designed to benefit computational and experimental biologists to understand the principles of analyzing biological data, building models and testing hypotheses using computer science paradigms. To visualize data and carry out data analysis, students will learn R or Python, and other programming languages for statistical computing and graphics. The class will be a combination of lecture and computer lab.

In this course, students investigate visual analytics tools and techniques used to synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data and to communicate the findings effectively for decision-making. Extensive use of case studies based on real-world events will be used to illustrate course concepts. Students will apply visual analytics techniques toward a focused research problem in a real-world application or a domain of interest.

This course introduces the internal operation of modern operating systems. The topics cover a brief history of operating systems, the major components of modern operating systems, and the object-oriented methodology on UNIX-like platform. Various UNIX tools will be used in the course and participants study examples using object-oriented programs as well as large system integration by object-oriented methodology.

In this course, students will be given an extensive overview of the various components of cybersecurity, including software development, operating systems, databases, and networks. Students will learn cybersecurity concepts, issues, and tools that are critical in solving problems in the computing security domain. The course will use lectures, reading assignments, and interactive lab exercises to reinforce the concepts that are introduced.

This course introduces various algorithms and analyzes the complexity and efficiency of the algorithms. Topics cover classic and heuristic algorithms, searching, sorting and parsing techniques, and algorithm complexity analysis.

This first junior clinic provides faculty guidance and supervision beyond the scope of existing courses. Students learn how to develop and structure their deliverables, as well as how to use computer science tools in the context of real-world or research projects.
CS 0351L Computer Science Junior Clinic II 1 Credit
Corequisite: SW 0301.
This second junior clinic provides faculty guidance and supervision beyond the scope of existing courses. Students learn how to develop and structure their deliverables, as well as how to use computer science tools in the context of real-world or research projects.

CS 0354 Theory of Programming Languages 3 Credits
Prerequisite: CS 0232.
Topics in this course include the design of programming languages; organization, control structures, data structures; run time behavior of programs; and formal specification and analysis of programming languages. The course includes a comparative survey of several significantly different languages.

CS 0355 Artificial Intelligence 3 Credits
Prerequisite: CS 0232.
This course, which examines computational and theoretical accounts of human intelligence, includes knowledge representation, commonsense reasoning, planning, natural language understanding, machine learning, and deep learning.

CS 0357 Database Management Systems 3 Credits
Prerequisite: CS 0232.
This course examines data formats, organizations, representations and structures; design and analysis of searching, sorting, and other algorithms; data management systems; relational database model; domains and relational integrity; structured query language; database design - logical and physical; entity-relationship diagrams; normalization; transaction processing; and database administration.

CS 0397 Internship 1-3 Credits
The internship program provides computer science majors with an opportunity to gain practical, career-related experience in a variety of supervised field settings. Internships can be in any one of a number of areas, such as software applications or hardware applications. Interns spend a minimum of 10 hours per week in on-site work, complete a required academic component specified by a faculty advisor, and satisfy the University Internship Policy requirements. Students may register for internships during the summer session and/or one to two full semesters and may earn a maximum of six internship credits. Open to seniors only; requires approval of the field placement supervisor and the School of Engineering. An internship may not replace a computer science elective to fulfill the requirement for a major in computer science.

CS 0398 Internship 1-3 Credits
The internship program provides computer science majors with an opportunity to gain practical, career-related experience in a variety of supervised field settings. Internships can be in any one of a number of areas, such as software applications or hardware applications. Interns spend a minimum of 10 hours per week in on-site work, complete a required academic component specified by a faculty advisor, and satisfy the University Internship Policy requirements. Students may register for internships during the summer session and/or one to two full semesters and may earn a maximum of six internship credits. Open to seniors only; requires approval of the field placement supervisor and the School of Engineering. An internship may not replace a computer science elective to fulfill the requirement for a major in computer science.

Software Engineering

SW 0300 Software Engineering Methods 3 Credits
Prerequisite: CS 0232.
This course explores the requirements gathering, system analysis, and software design methods of software application following the software processes required for the production of high quality software. Techniques for creating documentation and using software development tools will be presented. Students will gain experience in software project management, requirements, analysis, and safety issues in software development; interpersonal skills for management and team membership; and the software engineering discernment of systems architecture.

SW 0301 Software Design Methods 3 Credits
Prerequisite: SW 0300.
This course is the continuation of SW 0300 with in-depth projects and further discussions of design and implementation topics. Through the use of case studies and project work that has the student gradually building a large design specification, students will achieve an understanding of how complex applications are designed and built.

SW 0312 Agile Software Engineering 3 Credits
Prerequisite: SW 0300.
In this course, students apply in-depth techniques and experience various roles incorporated into one of the main approaches to software development which is agile methodology. It uses detailed knowledge about each of the major traditional software engineering phases to explore a more iterative approach for development of faster and more adaptable software. Proficiency in programming is expected of the students entering this course.

SW 0320 Software Testing and Maintenance 3 Credits
Prerequisite: SW 0300.
This course will cover in-depth methods for software testing, reliability and maintenance of software. Students will learn the principles of software testing and how to apply software testing techniques to the development of quality software and how to deploy software systems, maintain, enhance, and reuse software systems.

SW 0321 Software Project Management 3 Credits
Prerequisite: SW 0300.
This course explores and practices fundamental project management skills and life cycles required for both the successful management and development of software. Quality management principles of Personal Software Process (PSP) and Team Software Process (TSP) are introduced and practiced. Students will learn how to develop a project plan, scope a project, identify project activities, create work breakdown structures, estimate and schedule resources, construct and analyze project network diagrams, finalize project schedule and cost based on resource activity, recruit team members, organize and manage a project team, monitor and control progress, understand critical path project management, and have knowledge of both agile and traditional project management methods.

SW 0382 Special Topics (Shell) 3 Credits
This course provides an in-depth study of selected topics in software engineering of particular interest to the students and instructor. The course is counted as a major elective/specialization course. The topics and prerequisites will be announced when this course is offered.
SW 0383 Independent Study  1-3 Credits
This course is an individualized study under the supervision of the faculty member. The course emphasizes individual creativity. Students work with a faculty mentor in studying and investigating topics of current interest in software engineering. Students may earn from one to four credits for an independent study course. Enrollment by permission only.

Faculty

Professors
Rusu, Adrian, chair

Associate Professors
Rusu, Amalia

Assistant Professor
Wang

Lecturers
Corcoran
Galasso
Govindaraja
Kramer
Ramsey
Wilson