Computer Science

Computer science is the study of computers and their applications. This discipline includes a variety of specialties, such as software engineering, human-computer interaction, systems programming, artificial intelligence, robotics, networking, and graphics. In preparation for transfer into more advanced fundamental and specialized areas, students learn basic programming, data structures, and computer architecture. Computer science majors will also need two or three semesters of calculus and discrete mathematics, depending on their choice of transfer institution. Career options in computer science include software engineer, computer engineer, systems analyst, database administrator, and software project manager.

(See also: Computer Studies & Information Technology)

Academic and Career Pathway: Math and Sciences.

Contact Information

Chair: Catherine Walker
Dean: Al Taccone
https://www.miracosta.edu/academics/degree-and-certificate-programs/math-and-sciences/computer-science/index.html

Department: Computer Science
Office: Building OC4800, 760.795.6811

Full-Time Faculty

Nery Chapeton-Lamas
Eduardo Mariscal
Michael Paulding
Catherine Walker

Associate Degrees

Associate in Arts Degree
Software Development

Students may earn the above-named associate degree by completing a certificate of achievement and the general education courses required for MiraCosta's Associate in Arts degree (see Associate Degrees). Students should meet with a MiraCosta counselor to identify required courses and to develop a written educational plan for the specific degree or certificate they wish to earn.

Program Student Learning Outcome Statement

► Upon completion of this program, the student will be able to analyze, design, and generate mature software solutions to complex and abstract programming problems utilizing efficient and appropriate algorithms while working alone or cooperatively in a team. Students will also be able to competently explain and present the code and the results both verbally and in writing.

Associate in Science Degree
Computer Science for Transfer

Students completing this associate degree will have completed lower-division major preparation requirements for a computer science degree, an emphasis or option within a computer science degree, or a degree considered similar to computer science at a participating California State University (CSU) campus. Following transfer to a participating CSU campus, students will be required to complete no more than 60 units to obtain a bachelor's degree; however, some CSU campuses may require additional lower-division major preparation. This degree may not be appropriate preparation for students transferring to a CSU campus not accepting this degree or to a university or college that is not part of the CSU system. Students should consult with a MiraCosta counselor for further information regarding the most efficient pathway to transfer as a computer science major and to determine which CSU campuses are participating in this program.

To complete the degree, students must fulfill the following requirements:

► Complete a minimum of 60 CSU-transferable semester units.
► Complete all courses required in the major with a "C" or "P" or better.
► Complete the CSU-GE (Plan B) or IGETC (Plan C)* general education pattern.
► Obtain a minimum CSU-transferable GPA of 2.0.
► Complete a minimum of 12 units in residence at MiraCosta College.

*Students completing IGETC may be awarded the degree, but they must complete a course from Area 1C: Oral Communication to meet CSU admission requirements.

Program Student Learning Outcome Statement

► Upon successful completion of this program, students will be able to develop communication, critical thinking, algorithmic analysis, programming, and problem solving skills that will effectively prepare them for advanced study in computer science at the university level.

Required courses:

CS 112	Introduction to Computer Science II: Java	3
CS 113	Basic Data Structures and Algorithms	3
CS 220	Computer Architecture and Assembly Language	3
CS 226	Discrete Structures	4
MATH 150	Calculus and Analytic Geometry I *	5
or MATH 150H	Calculus and Analytic Geometry I (Honors)	
MATH 155	Calculus and Analytic Geometry II *	4
or MATH 155H	Calculus and Analytic Geometry II (Honors)	
PHYS 151	Principles of Physics I *	4
or PHYS 151H	Principles of Physics I (Honors)	
BIO 204	Foundations of Biology: Biochemistry, Cell Biology, Genetics, and Molecular Biology *	4
or BIO 204H	Foundations of Biology: Biochemistry, Cell Biology, Genetics, and Molecular Biology (Honors)	
or CHEM 150	General Chemistry I: For Science Majors	

Computer Science

or CHEM 150H	General Chemistry I: For Science Majors (Honors)	
or PHYS 152	Principles of Physics II	
or PHYS 152H	Principles of Physics II (Honors)	
Total Units		**30**

* Course satisfies a general education requirement on the CSU-GE or IGETC general education (GE) pattern. To ensure this degree is completed with no more than 60 units, students should select courses that will also satisfy a GE requirement. Note: All of the BIO, CHEM, MATH, and PHYS courses listed are also GE courses.

NOTE: Students are strongly advised to select courses that meet lower-division major preparation requirements at their transfer university and to complete the History, Constitution, and American Ideals requirement prior to transfer.

Certificate

Certificate of Achievement
Software Development

This certificate provides a strong foundation for knowledge of basic object-oriented programming (OOP) techniques and data structures used in computer science. Students gain experience in several OOP programming languages and in analyzing and assessing problems and formulating appropriate solutions. Completing this certificate will help students advance in their careers as computer programming professionals and solidify the foundation of their computer science skills.

Program Student Learning Outcome Statement

▶ Upon completion of this program, the student will be able to analyze, design, and develop mature software solutions to complex and abstract programming problems utilizing efficient and appropriate algorithms while working alone or cooperatively in a team. Students will also be to competently explain and present the code and the results both verbally and in writing.

Required courses:		
CS 111	Introduction to Computer Science I: Java	3
CS 112	Introduction to Computer Science II: Java	3
CS 113	Basic Data Structures and Algorithms	3
CS 150	C++ Programming	3
CS 210	Software Engineering	3
CS 220	Computer Architecture and Assembly Language	3
Select at least 6 elective units from the following:		6
CS 130	Fundamentals of Scripting Languages	
CS 134	Mobile Application Development	
CS 138	Programming with Python	
CS 139	Advanced Programming with Python	
CS 151	Advanced C++ Programming	

CS 155	C# and the .NET Framework	
CS 226	Discrete Structures	
or MATH 226	Discrete Mathematics	
or MATH 226H	Discrete Mathematics (Honors)	
BUS 133	Project Management	
Total Units		**24**

Certificate of Proficiency
Game Developer

This certificate provides the focused skill set necessary for video game development. Students gain experience in writing efficient code and designing the user experience while developing strong problem-solving skills. Curriculum emphasizes the data structures, algorithms, and software architecture techniques used to create games. Completing this certificate will help students advance their careers as game developers and solidify the foundation of their computer science skills.

Program Student Learning Outcome Statement

▶ Upon completion of this program, students will be able to demonstrate proficiency in the fundamental techniques of video game design and development.

Required courses:		
CS 111	Introduction to Computer Science I: Java	3
or CS 150	C++ Programming	
CS 155	C# and the .NET Framework	3
Select one course from the following:		3
MAT 145	UI/UX Design	
MAT 150	Animation and Interactivity	
MAT 190	Programming for Animation, Interactivity, and Games	
Total Units		**9**

Certificate of Proficiency
Mobile Application Development

The Mobile Application Development certificate offers students familiar with programming languages or just starting out the opportunity to specialize in mobile app development for smartphones and tablets with an open-source platform. Students employ active learning with Integrated Development Environments (IDEs), frameworks, APIs, designing and handling user input, designing layouts, application events, managing navigation, and core device features, such as GPS, accelerometers, and touch gestures.

Program Student Learning Outcome Statement

▶ Upon successful completion of this program, students will be able to demonstrate proficiency in the fundamental techniques of mobile application development.

Required courses:		
CS 111	Introduction to Computer Science I: Java	3
CS 112	Introduction to Computer Science II: Java	3

CS 134	Mobile Application Development	3
Select one course from the following:		3
MAT 145	UI/UX Design	
MAT 150	Animation and Interactivity	
MAT 190	Programming for Animation, Interactivity, and Games	
Total Units		**12**

Courses

CS 101: Introduction to Computer Science Principles
Units: 3
Prerequisites: None
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring, and Summer

This course introduces beginning or exploring students to core principles, theories, and topics in computer science while emphasizing their relevance in society. Topics include general computational problem-solving techniques, design of digital devices, how computers work, and how to design, implement, test, and document programs in both an interactive and a high-level programming language (Python). This course is highly recommended for students prior to taking a beginning programming class (CS 111, CS 150, CS 130, CS 138, or CS 155), non-computer science majors, or those interested in exploring computer science.

CS 111: Introduction to Computer Science I: Java
Units: 3
Prerequisites: None
Advisory: CS 101 and MATH 64 or MATH 64S.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring, and Summer

This course introduces object-oriented programming and concepts designed primarily for students majoring in computer science and engineering who have some programming fundamentals. The course uses topics of personal and social relevance to investigate the impacts of computing through exploring language basics, including control structures, data types, input/output, operators, classes, methods and parameters, basic inheritance, and documentation practices as well as testing and verification techniques. UC CREDIT LIMITATION: No credit for CS 111 if taken after CS 112. C-ID COMP 122, ITIS-130.

CS 112: Introduction to Computer Science II: Java
Units: 3
Prerequisites: CS 111.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring, and Summer

This course uses topics of personal and social relevance to investigate the impacts of computing through exploring advanced object-oriented programming concepts such as abstraction, inheritance, polymorphism, and encapsulation. Topics include recursion, generics, event-driven programming, graphical user interfaces, file input and output, and exception handling.

CS 113: Basic Data Structures and Algorithms
Units: 3
Prerequisites: CS 112.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring, and Summer

The course uses topics of personal and social relevance to investigate the impact of computing through efficient algorithms and properly designed data structures. Students explore the software development process by developing effective solutions using industry-standard tools. Topics include searching, sorting, hashing, algorithm analysis, object-oriented design, collections, lists, stacks, queues, trees, sets, dictionaries, and graphs. C-ID COMP-132.

CS 130: Fundamentals of Scripting Languages
Units: 3
Prerequisites: None
Advisory: CS 101.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Spring

This course introduces students to scripting language(s) and webpage programming. Students develop, debug, implement, and integrate client-side scripts using automated tasking. Comparisons may be made among common scripting languages, such as JavaScript, PHP, Perl, and CGI.

CS 134: Mobile Application Development
Units: 3
Prerequisites: CS 112.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Spring

This course covers how to develop applications for smartphones and tablets with the Android open-source platform. Topics include the Model-View-Controller architecture, designing and handling user input, defining activity layouts, application events, displaying images, managing navigation, and core device features, such as GPS, the Google Maps API, accelerometers, and touch gestures. Students leverage and expand their Java programming knowledge to build mobile applications from start to finish.

CS 138: Programming with Python
Units: 3
Prerequisites: None
Advisory: CS 101.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring, and Summer

This course introduces the Python programming language and its features. Python, a dynamic, object-oriented, extensible language, is perfect for the beginner and also meets industry needs. Python is well-suited for applications ranging from simple data manipulation to large, complex applications.

Computer Science

CS 139: Advanced Programming with Python
Units: 3
Prerequisites: CS 138.
Acceptable for Credit: CSU
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring

This course continues with the Python programming language, using topics of personal and social relevance to investigate the impact of computing. This course includes data structures and libraries for modularization, data extraction and visualization, web APIs, server applications, and an introduction to machine learning.

CS 150: C++ Programming
Units: 3
Prerequisites: None
Advisory: MATH 64 or and CS 101 MATH 64S
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring, and Summer

Using an object-oriented approach to design and programming in the C++ language, this course covers data input/output, data types, control structures, operators, functions, and the operating environment. Upon successful completion of the course, students are able to construct moderately complex programs in C++. C-ID COMP-122.

CS 151: Advanced C++ Programming
Units: 3
Prerequisites: CS 150.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring

In this advanced programming course, students design and implement increasingly complex C++ programs that build upon skills acquired in C++ Programming (CS 150). Students also develop appropriate and efficient methods to test their programs. Topics include polymorphism, inheritance, class libraries, the standard template library, pointers, advanced file input/output operations, recursion, virtual functions, exception handling, dynamic memory management, bitwise operators, and data structures, such as linked lists, stacks, queues, and binary trees.

CS 155: C# and the .NET Framework
Units: 3
Prerequisites: None
Advisory: CS 101.
Acceptable for Credit: CSU
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall

This course provides an introduction to the C# ("C Sharp") programming language and the .NET framework. The C# language encompasses functional, generic, object-oriented, and component-oriented programming disciplines. The .NET framework provides the foundation of language interoperability for the Microsoft Windows operating system. C# is designed for modern software development and provides the tools to build applications quickly.

CS 210: Software Engineering
Units: 3
Prerequisites: CS 112.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring

This course introduces students to team-based software development and engineering methods, including specification, design, implementation, testing, and process. It emphasizes team development, agile methods, software design, and use of software engineering tools, such as integrated development environments, version control, automated test framework, build automation, and software containers. As a culminating project, each team produces three deliverables: a Software Requirements Specification document, a Software Design Specification document, and an executable application satisfying stakeholder requirements.

CS 220: Computer Architecture and Assembly Language
Units: 3
Prerequisites: CS 112.
Acceptable for Credit: CSU, UC
Lecture 2 hours, laboratory 3 hours.
Course Typically Offered: Fall, Spring, and Summer

This course introduces the fundamental physical and structural concepts of assembly language programming. Topics include machine architecture, memory addressing, input/output, interrupts, control structures, compiling, and linking. C-ID COMP-142.

CS 226: Discrete Structures
Units: 4
Prerequisites: CS 111, CS 138, or CS 150 and MATH 126, MATH 126S, or eligibility determined by the math placement process.
Acceptable for Credit: CSU, UC
Lecture 3.50 hours, laboratory 1.50 hours.
Course Typically Offered: Fall, Spring

Designed for students majoring in computer science, this course introduces discrete mathematics, including logic, methods of proof, number theory, sets, counting, relations, recursion, recurrence relations, Boolean algebra, graphs, trees, and networks. Topics are illustrated with applications to computer science, including design and analysis of algorithms, undecidability, program correctness, and digital logic design. C-ID COMP-152.

CS 292: Internship Studies
Units: 0.5-3
Prerequisites: None
Corequisite: Complete 75 hrs paid or 60 hrs non-paid work per unit.
Enrollment Limitation: Instructor, dept chair, and Career Center approval. May not enroll in any combination of cooperative work experience and/or internship studies concurrently.
Acceptable for Credit: CSU
Course Typically Offered: To be arranged

This course provides students the opportunity to apply the theories and techniques of their discipline in an internship position in a professional setting under the instruction of a faculty-mentor and site supervisor. It introduces students to aspects of the roles and responsibilities of professionals employed in the field of study. Topics include goal-setting, employability skills development, and examination of the world of work as it relates to the student's career plans. Students must develop new learning objectives and/or intern at a new site upon each repetition. Students may not earn more than 16 units in any combination of cooperative work experience (general or occupational) and/or internship studies during community college attendance.

CS 299: Occupational Cooperative Work Experience
Units: 1-4
Prerequisites: None
Corequisite: Complete 75 hrs paid or 60 hrs non-paid work per unit.
Enrollment Limitation: Career Center approval. May not enroll in any combination of cooperative work experience and/or internship studies concurrently.
Acceptable for Credit: CSU
Course Typically Offered: To be arranged

Cooperative Work Experience is intended for students who are employed in a job directly related to their major. It allows such students the opportunity to apply the theories and skills of their discipline to their position and to undertake new responsibilities and learn new skills at work. Topics include goal-setting, employability skills development, and examination of the world of work as it relates to the student's career plans. Students may not earn more than 16 units in any combination of cooperative work experience (general or occupational) and/or internship studies during community college attendance.