

---

# Computer Science GRADUATE HANDBOOK

---



Science II Building, Room C255  
559.278.4373

***“AS IN YOUR FUTURE IS 278-H-E-R-  
E”***

**M.S. in Computer Science**

<http://www.csufresno.edu/csci/e-mail:cs@csufresno.edu>

<http://www.csufresno.edu/gradstudies>

Welcome to the graduate program in Computer Science!

The field of Computer Science is always changing. It is interesting, exciting, and challenging to study Computer Science. You can learn the classics, explore a new area, and create new problems and answers.

In order to maintain a good “graduate life”, you have to know what rules and deadlines you will be expected to follow. This handbook is just a first step for you to complete your Master’s program. Our department and our graduate faculty are ready to help you face this new challenge.

I ask that you also pay close attention to your [Chart Your Course Brochure](#) provided by the Graduate Office to learn about all the processes you will have to follow to maneuver successfully through your program of study.

Again welcome and we look forward to serving your needs.

Shigeko Seki  
Graduate Coordinator  
Department of Computer Science  
278-3026  
[seki@csufresno.edu](mailto:seki@csufresno.edu)

## **The Department**

Computer science is applied reasoning using both art and science: It requires the ability to communicate ideas through a combination of language and powerful technology. It is concerned with the interaction of humans and computers, as well as the application of computers to a myriad of specialized problems.

---

## **Program Description**

The goal of the Department of Computer Science is to offer programs to a diverse audience: (1) students interested primarily in computing, (2) students interested primarily in applying computing to some other field of study, and (3) students who wish to include computing as part of their general education.

---

## **Career Opportunities**

Computer use pervades our society, and the industry supporting that use has grown rapidly. Graduates from this program find job opportunities in such diverse fields as computer design, software engineering, systems analysis, database design, computer graphics, and technical programming. Because of the strong theoretical foundation of the program, graduates are attractive to companies involved in computer manufacturing and to those industries using computers in high-technology applications.

Our proximity to two of the largest areas using computers in the nation, Silicon Valley and Los Angeles, provides our graduates with a broad-based collection of potential employers. Graduates have also obtained exciting and challenging positions at Air Force and Naval bases in California. A significant proportion of our graduates pursue graduate studies. Students who obtain the master of science degree will be in an excellent position to pursue a Ph.D.

---

## **Organizations**

Student chapters of the Association for Computing Machinery (ACM) and the IEEE Computer Society are very active in the department. They organize field trips to major computer manufacturers and users in California. The ACM chapter sponsors the fall Programming Contest.

Computer science majors who have a distinguished academic record in computer science are invited to join Upsilon Pi Epsilon, the Honor Society for the Computing Sciences.

---

## **Co-op Program**

Through the Cooperative Education program, students receive academic credit and are employed in computer-related industries. This is an excellent opportunity for a student to obtain experience, a reasonable salary, and college credit in this field.

---

## **Faculty and Facilities**

The faculty comes from a variety of areas including computer systems and architecture, theoretical computer science, programming languages, software engineering, computer graphics, distributed systems and parallel processing, neural networks, image processing, computer vision, pattern recognition, wireless communication and mobile computing, robot swarm communication, evolutionary computation, domain-specific languages, and real-time and embedded systems. They have in common a desire to provide a program that will give the student a broad range of experience in computer science as well as the depth of education that will be needed in the student's later career, whether professional or academic.

Students and faculty have access to a networked environment of UNIX workstations (Sun Microsystems and Linux systems) and microcomputer laboratories of PCs. These systems are connected to campus and international networks.

### **Faculty/\*Graduate Faculty**

\*J. Todd Wilson, Chair  
\*Shigeko Seki, Graduate Coordinator

\*Brent J. Auernheimer  
\*Lan Jin  
\*Ming Li  
\*Shih-Hsi Liu

### **Emeiriti Faculty:**

\*Grace C. N. Wei  
\*Henderson Yeung

**Part-time Faculty:**

Prudence Lowe

Hovsep Tanielian, Spring 2011

Jennifer Yan, Spring 2011

---

## **Department of Computer Science**

# **GRADUATE PROGRAM**

---

The Master of Science degree program in Computer Science is designed to offer the advanced principles, applications, and current topics in computer science. Students who obtain the M.S. will be ready to do significant developmental work in the computer industry or in an important application area and will also be well qualified to pursue a Ph.D.

Applicants may hold an acceptable bachelor's degree in any field of study and must submit Graduate Record Examination (GRE) scores.

To attain classified standing at the time of admission, an applicant must:

1. have a minimum grade point average of 2.75 in the last 60 units and
2. have completed the following undergraduate prerequisite courses or equivalents with a minimum grade point average of 3.0: CSCI 40, 41, 60, 112, 113, 115, 117, 119, 144, MATH 75, 76.

Applicants who do not meet the requirements 1 and 2 above may be admitted to conditionally classified standing to complete the remaining prerequisites at California State University, Fresno. Approved coursework up to a maximum of 10 units of the 30 units required for the M.S. can be taken concurrently with prerequisite courses by a student with conditionally classified standing.

To attain classified standing from conditionally classified standing, a student must complete the remaining prerequisite courses with a minimum grade point average of 3.0 and have earned a minimum grade point average of 3.0 in all coursework taken toward the M.S. in Computer Science.

---



## **Master of Science Degree Requirements**

The Master of Science degree requires a minimum of 30 units after the completion of the baccalaureate degree according to the criteria below. At least 21 units of the total must be taken in 200-level courses in computer science. The undergraduate courses used toward the bachelor's degree or toward fully classified status may not be used toward the master's degree.

**Required courses (10 units)**

CSCI 174 or 188, 200, 213, 217

**Electives (9 units)**

Three of the following: CSCI 226, 230, 244, 250, 252, 253, 272, 274, 282, 284

**Approved electives (5-8 units)****Culminating experience (3-6 units)**

CSCI 298 or 299

**Total (30 units)**

In order to be eligible for advancement to candidacy in the M.S. in Computer Science program, all students must pass CSCI 200 with a grade of B or better. In addition, all students must demonstrate competence in graduate-level writing prior to being advanced to candidacy. Students may fulfill this requirement by passing the writing component of CSCI 200. Please see the graduate program coordinator for further information.

***COMMON PREREQUISITES TO THE MASTERS PROGRAM***

The Graduate Coordinator will determine the courses individual students need to take as prerequisites to the M. S. Program in Computer Science. Prerequisite requirements are set based on a policy that ensures proper preparation for the Master in Computer Science. Course prerequisite decisions are determined after a thorough review of individual student transcripts and an assessment of each individual's prior educational experiences.

**CSCI 40. Introduction to Programming and Problem Solving (4)**

Prerequisites: Students must take the ELM exam; students who do not pass the exam must record a grade of C or better in a college-taught intermediate algebra course; trigonometry. Introduction to problem solving, algorithm development, procedural and data abstraction; program design, coding, debugging, testing, and documentation; a high-level programming language. (3 lecture, 2 lab hours)

**CSCI 41. Introduction to Data Structures (4)**

Prerequisite: CSCI 40. Programming methodology, program correctness. Review of data types. Data structures: linear and nonlinear structures, files. Implementation of data structures. Recursion. Searching and sorting. (3 lecture, 2 lab hours)

**CSCI 60. Foundations of Computer Science (4)**

Prerequisites: CSCI 40 (may be taken concurrently.) Abstraction, iteration, induction, recursion, complexity of programs, data models, and logic. (3 lecture, 2 lab hours)

### **CSCI 112. Introduction to Computer Systems (4)**

Prerequisite: CSCI 41, 60. Computer arithmetic. Von Neumann architecture. Instruction sets, data types, formats, addressing. Register and ALU organization. Memory hierarchy. I/O. Bus organization. Study of one or more assembly languages. Basics of implementation of higher-level languages. (3 lecture, 2 lab hours)

### **CSCI 113. Introduction to Computer Organization (4)**

Prerequisite: CSCI 41. Fundamental issues of computer design at register-transfer level. Logical design of basic combinational and sequential modules. Organization and design of major functional blocks: ALU, CPU, memory, cache, input/output, hard-wired and microprogrammed control. Simulation of computer organization. Introduction to high-performance superscalar computer organization. (3 lecture, 2 lab hours)

### **CSCI 115. Algorithms and Data Structures (4)**

Prerequisites: CSCI 41, 60; MATH 75. Review of basic data structures. Graph, search paths, and spanning trees. Algorithm design and analysis of sorting, merging, and searching. Memory management, hashing, dynamic storage allocation. Integration of data structures into system design. (3 lecture, 2 lab hours)

### **CSCI 117. Structures of Programming Languages (4)**

Prerequisites: CSCI 41, 60. General concepts and paradigms of programming languages; scope and binding rules, applications and implementations of language concepts. Languages selected from: ADA, ICON, Miranda, ML, MODULA 2, OCCAM 2, PROLOG, LISP, Scheme, SmallTalk. (3 lecture, 2 lab hours)

### **CSCI 119. Introduction to Finite Automata (4)**

Prerequisites: CSCI 41, 60. Strings, languages, and fundamental proof techniques. Regular expression, regular grammar, regular languages, finite automata, their interrelationship, and their properties. Introduction to context-free languages. (3 lecture, 2 lab hours)

### **CSCI 144. Introduction to Operating Systems (4)**

Prerequisites: CSCI 41 and CSCI 112 or ECE 118. Operating system history and services. File systems. Memory management. Process management -- concurrent processes, communication, semaphores, monitors, deadlocks. Resource management -- processor and disk scheduling. Security and protection mechanisms. (3 lecture, 2 lab hours)

**Math 75 and 76-** Math Analysis I-Calculus and/or Math Analysis II Calculus

---

*The following undergraduate courses are eligible to be used as approved electives in the master's program. Students are allowed to include a maximum of 6 units from those listed below but must get the approval from their project (C Sci 298) advisors to include them in their master's program of study.*

**CSCI 124. Introduction to File Processing (3)**

Prerequisite: CSCI 115. Definition of file components, access methods, and file operations. Algorithms for efficient implementation of data structures; characteristics of bulk storage media for mainframe and microcomputers. Introduction to database management systems.

**CSCI 126. Database Systems (3)**

Prerequisite: CSCI 124. Database concepts; hierarchical and relational network models; object-oriented data models. Data normalization, data description languages, data manipulation languages, and query design.

**CSCI 130. Web Programming (3)**

Prerequisite: CSCI 115. Programming for the World Wide Web. Web servers and clients, Internet and Web protocols, and mark-up languages. Client-side scripting, including both gateway and filter-based approaches. (2 lecture, 2 lab hours)

**CSCI 134. Compiler Design (3)**

Prerequisites: CSCI 112, 115, 119. Syntax and semantics of programming languages. Lexical analysis, parsing techniques, parser generator, SLR and LALR parsing. Introduction to symbol table organization and semantic routines. Compiler generators.

**CSCI 146. Systems Architecture (3)**

Prerequisites: CSCI 113, 144. An in-depth analysis of one or more operating systems -- system data structures, hardware architecture, shell and kernel functions, I/O routines, interrupt handling. Other topics may include parallel hardware architectures, performance analysis.

**CSCI 148. Systems Programming (3)**

Prerequisites: CSCI 113, 144. Topics include implementation of operating system components and modification of existing systems. Device drivers, memory management, communication networks, and file systems will be examined. Projects will be emphasized.

**CSCI 152. Software Engineering (3)**

Prerequisite: CSCI 150. In-depth examination of techniques for specification, design, implementation, testing, and verification of software. Human-computer interfaces. Formal methods of software development. Use of software engineering tools for the development of substantial software projects. (2lecture 2 lab hours)

### **CSCI 154. Simulation (3)**

Prerequisites: CSCI 41, 60; MATH 75. Simulation as a tool for the study of complex systems in computer science, statistics and operations research. Generating random variables. Review of principles behind and examples of simulation languages.

### **CSCI 156. Internetworking Systems and Protocols (3)**

Prerequisite: CSCI 144 or permission of instructor. Review of underlying network technologies. Application-level interconnections, network architectures, addressing, mapping abstract addresses to physical addresses, routing datagrams, error and control messages, protocol layering, gateways, subnets. Client-server interactions. Upper layers of protocol stacks. (3 lecture hours)

### **CSCI 164. Artificial Intelligence Programming (3)**

Prerequisite: CSCI 117. Introduction to problem-solving methods from artificial intelligence. Production systems. Knowledge-based systems. Machine learning. Topics chosen from fuzzy logic, neural network models, genetic algorithms. Verification, validation, testing.

### **CSCI 166. Principles of Artificial Intelligence (3)**

Prerequisite: CSCI 164. Analysis of knowledge-based and neural models, including self-organization, sequential learning models, neurally inspired models of reasoning and perception. Integration of different paradigms.

### **CSCI 172. Computer Graphics (3)**

Prerequisites: MATH 76, CSCI 41, and (CSCI 112 or ECE 118). Hardware devices, raster graphics, device independence, graphic data structure and representations, interactive techniques, and algorithms for the display of two- and three-dimensional objects, graphic transformations, graphics standards, modeling, animation, VRML and scientific visualization. (3 lecture hours)

### **CSCI 173. Advanced Computer Graphics (3)**

Prerequisite: CSCI 172. Visible surface algorithms, lighting and shading, textures, curves and surfaces, computer-aided design, advanced modeling techniques, solid modeling, advanced raster graphics architecture, advanced geometric and raster algorithms, user interface, ray tracing, animation techniques, and fractals. (2 lecture, 2 lab hours).

### **CSCI 174. Design and Analysis of Algorithms (3)**

Prerequisites: CSCI 115, 119. Models of computation and measures of complexity, algorithms for sorting and searching, set representation and manipulation, branch and bound, integer and polynomial arithmetic, pattern-matching algorithms, parsing algorithms, graph algorithms, NP-complete problems.

### **CSCI 176. Parallel Processing (3)**

Prerequisites: CSCI 113, 144. Characteristics, and classification of computer systems. Notion and realization of parallelism. Pipeline design techniques. Vector processing.

Array processing. Multiprocessing. Multiprocessing vs. multicomputers. Shared memory vs. message-passing, problem solving, and parallel programming. Architectural trends.

**CSCI 177. Distributed Computer Systems (3)**

Prerequisites: CSCI 113, 144. Characteristics and design of distributed systems. Application and network interconnectivity. Enterprise computing. Distributed data and transaction management. Distributed operating systems. Distributed problem solving and programming.

**CSCI 186. Formal Languages and Automata (3)**

Prerequisite: CSCI 119. Introduction to formal language theory. Context-free grammars, context-sensitive grammars, unrestricted grammars; properties of context-free languages, push-down automata.

**CSCI 188. Introduction to Computability (3)**

Prerequisite: CSCI 119. Introduction to computability and complexity. Turing machines, recursive functions, reduction, undecidability, classes P and NP, and intractable problems.

**CSCI 191T. Proseminars** Prerequisite: permission of instructor.

Presentation of selected topics in computer science.

**GRADUATE CURRICULUM**

**CSCI 174. Design and Analysis of Algorithms (3)**

Prerequisites: CSCI 115, 119. Models of computation and measures of complexity, algorithms for sorting and searching, set representation and manipulation, branch and bound, integer and polynomial arithmetic, pattern-matching algorithms, parsing algorithms, graph algorithms, NP-complete problems.

**CSCI 188. Introduction to Computability (3)**

Prerequisite: CSCI 119. Introduction to computability and complexity. Turing machines, recursive functions, reduction, undecidability, classes P and NP, and intractable problems.

**CSCI 200. Introduction to Research in Computer Science (1)**

Prerequisite: classified standing in computer science. Orientation to the graduate program, introduction to research methodology, and discussion of possible project and thesis topics.

**CSCI 213. Computer Organization (3)**

Prerequisites: CSCI 112 and 113 or permission of instructor. Organization of memory, I/O, and processors. Computer busses. Microprogramming and instruction execution.

Interrupts. Data communications.

### **CSCI 217. Programming Language Principles (3)**

Prerequisite: CSCI 117 or permission of instructor. Advanced topics in programming languages: concurrency, exceptions, types, procedures, execution models. Introduction to the formal specification of programming languages: syntax specification, semantic specification.

### **CSCI 226. Advanced Database Systems (3)**

Prerequisites: CSCI 126 and 144. Implementation of database systems on modern hardware systems. Operating system design issues, including buffering, page size, prefetching, etc. Query processing algorithms; design of crash recovery and concurrency control systems. Implementation of distributed databases and database machines.

### **CSCI 230. Advanced Web Application Development (3)**

Prerequisite: CSCI 130 or permission of instructor. Application development for the World Wide Web. Three-tier architecture; authentication, capability, and session management; versioning and open-source development. Case studies and project work. (Formerly CSCI 291T)

### **CSCI 244. Operating Systems (3)**

Prerequisite: CSCI 144. Operating system functions. Performance monitoring and fine-tuning. Network operating system design. Concurrency, analysis of deadlock. Selected topics from current research.

### **CSCI 250. Advanced Software Engineering (3)**

Prerequisite: CSCI 150 or permission of instructor. Theoretical and practical aspects of software engineering emphasizing requirements analysis, specification, design, coding, testing, correctness, maintenance, and management. Examination of reliability, performance, and software metrics.

### **CSCI 252. Software Development Environments (3)**

Prerequisite: CSCI 150. Overview of modern software engineering environments including structured editors, programmer's assistants, and tools for software cost estimation, testing, scheduling, specification, and verification. Relationship between artificial intelligence and software engineering.

### **CSCI 253. Human Computer Interaction (3)**

Prerequisite: CSCI 150. Software engineering approach to human-computer interaction. Design, evaluation, and implementation of user interfaces and experiences. Modeling, prototyping, inspection, and usability testing. Relationship of user interface characteristics to attention, errors, and efficiency.

### **CSCI 272. Computer Graphics (3)**

Prerequisite: CSCI 172 or permission of instructor. 3-D transformations, visible surface algorithms, shading, textures, curves and surfaces, computer-aided design, advanced

modeling techniques, solid modeling, advanced raster graphics architecture, advanced geometric and raster algorithms, user interface, ray tracing, animation techniques, and fractals.

**CSCI 274. Combinatorial Algorithms (3)**

Prerequisite: CSCI 174. Design and analysis of efficient algorithms for combinatorial problems. Network flow theory, matching theory, augmenting-path algorithms, branch-and-bound algorithms, data structure techniques for efficient implementation of combinatorial algorithms, analysis of data structures, application of data structural techniques to sorting, searching, and geometric problems.

**CSCI 282. Theory of Computation (3)**

Prerequisite: CSCI 188 or permission of instructor. General models of computation, recursive functions, undecidable problems, propositional calculus, predicate calculus, complexity classes, NP-complete problems.

**CSCI 284. Automata Theory (3)**

Prerequisite: CSCI 186 or permission of instructor. Formal languages, abstract machines, algebraic approach to automata, term rewriting systems, formal power series, cryptography, parallel computation.

**CSCI 290. Independent Study**

Prerequisite: approval of department. See Academic Placement -- [Independent Study](#). Approved for RP grading.

**CSCI 291T. Seminar**

Prerequisite: approval of instructor. Special topics in computer science of current interest and importance.

**CSCI 298. Research Project (3)**

Prerequisite: advancement to candidacy. Get [Criteria for Project](#) from your advisor. Independent investigation of an advanced topic as the culminating requirement for the master's degree. Approved for RP grading.

**CSCI 299. Master's Thesis (3-6; max total 6)**

Prerequisite: advancement to candidacy. See [Criteria for Thesis and Project](#) in the regular catalog or see your Thesis Advisor. Preparation, completion, and submission of an acceptable thesis for the master's degree. Approved for RP grading.

[PDF to Word](#)