Computer Science

College of Engineering and Computer Science

Bachelor of Science • Minor •
Master of Science • Certificates

PROGRAM DESCRIPTION

The Bachelor of Science degree in Computer Science is accredited by the Computing Accreditation Commission (CAC) of ABET, Inc., providing majors with a sound educational base in Computer Science. The Minor in Computer Science is available for students majoring in disciplines other than Computer Engineering.

The certificate program in Web Development is available to both majors and non-majors. Majors interested in enhancing their degrees need only take CSC 121 and CSC 123 in addition to major-required courses. Non-majors who complete the series of seven courses will be able to install and maintain a reasonably complex web server, create home pages, and link pages to local databases.

Career Possibilities

Computer Scientist • Management Information Specialist • Computer Systems Analyst • Technical Representative • Teleprocessing Coordinator • Scientific Application Programmer • Data Processing Application Programmer • Computer Operations Manager • Computer Services Coordinator • Database Administrator • Data Communications Manager • Data Processing Manager • Information Specialist • Programmer Analyst • Software Engineer • Systems Manager • Systems Programmer • Technical Control Specialist • Computer Graphics Specialist • Knowledge Engineer • Systems Engineer • Security Specialist • Data Mining Analyst

Faculty

Behnam Arad, Robert Buckley, Senad Busovaca, Weide Chang, John Clevenger, Nikrouz Faroughi, Isaac Ghansah, V. Scott Gordon, Bolan Jiang, Ying Jin, Ju-Yeon Jo, Roxalie Jones, Ted Krovetz, Kwai-Ting Lan, Meiliu Lu, William Mitchell, Jinsong Ouyang, Anne-Louise Radimsky, Ahmed Salem, Chung-E Wang, Cui Zhang, Du Zhang

Contact Information

Du Zhang, Department Chair
Sara Joslin, Administrative Support Coordinator
Riverside Hall 3018
(916) 278-6834
www.ecs.csus.edu/csc

The Department offers three programs in conjunction with other units: with the Mathematics Department, a joint program in Mathematics and Applied Computing (refer to the Mathematics section of this catalog); and with the Electrical and Electronic Engineering Department, a BS and an MS in Computer Engineering (refer to the Computer Engineering section of this catalog).

Special Features

- The research interests of the department faculty span a broad spectrum of Computer Science including active databases, algorithm analysis, autonomic computing, bioinformatics, compilers, computer architecture, computer forensics, computer graphics and computer game design, computer networks and communications, computing theory, cryptography, data models and database management systems, data warehousing and data mining, distributed systems and real-time systems, formal methods, human-computer interfaces, information assurance and security, intelligent and knowledge-based systems, Internet and web technologies, machine learning, network security, operating systems, performance modeling and evaluation, programming languages and methodologies, software architecture, software project management, software requirements engineering, software system engineering, verification and validation, VLSI design.
- A large heterogeneous network of Linux 64-bit, Sun RISC, and Hewlett-Packard RISC servers combined with Linux, SunOS, HPUX, Windows, and MacOS workstations supports the instructional programs. Linux and Windows workstation laboratories support both lower and upper-division instruction. Specialized laboratories support systems, communications and networking, and computer architecture instruction. A graduate laboratory is designed to provide graduate students with access to a variety of advanced workstations. All students have access to the Internet and every conceivable Web resource.
- Majors are urged to join the student chapters of the Association for Computing Machinery or the IEEE Computer Society. Students with high scholastic achievement may be invited to join Upsilon Pi Epsilon, the national honor society for computer science.
- The University's proximity to many California State agencies and major computer corporations provides numerous opportunities for part-time student employment.

Bachelor of Science Educational Objectives

The objectives of this program are to prepare graduates to:

- analyze, design, and implement a computerized solution to a significant problem in a team environment using appropriate tools;
- enter a professional computer science position or an appropriate graduate program;
- communicate effectively through speaking, writing, and graphics;
- pursue life-long learning and continued professional development; and
- be aware of ethical issues and societal concerns relating to computers in society and apply this knowledge in the conduct of their careers.

UNDERGRADUATE PROGRAMS

All students are admitted as pre-Computer Science majors. Registration in upper-division courses numbered 133 and above is restricted to Computer Science and Computer Engineering majors. Other students need to obtain approval from the CSC Department Chair. To change to the Computer Science major, students who have completed the following lower division (pre-major) requirements are required to complete and submit a Change of Major form to the Computer Science Department Office along with transcript copies: CSC 15, CSC 20, CSC 28, CSC 35, CSC 60, and MATH 30 and MATH 31.

Course Repeat Policy: Students may repeat a Computer Science course once at most without permission. Additional repeats must be approved by the instructor and the Department Chair by completing a “Request to Repeat” petition and submitting it to the Department.

Work Experience: Students may receive a limited amount of academic credit for relevant work experience in Computer Science. There are many opportunities for students to work part-time in the federal, state and local governments. A significant number of positions in private industry are also available in both well-established and new companies in the Sacramento area. Such work experiences often lead to permanent positions upon graduation.

Note: Useful information can also be found in the College of Engineering and Computer Science section.

Requirements • Bachelor of Science Degree

Units required for Major: 87-89
Minimum total units required for BS: 129

Grade of “C-” or better required in all courses applied to the Computer Science major.

Note: Additional units may be required to meet the Sacramento State foreign language requirement.

Courses in parentheses are prerequisites.

A. Required Lower Division Courses (15 units)

(3)	CSC 15	Programming Concepts and Methodology I (CSC 10 or programming experience in a high-level programming language)
(3)	CSC 20	Programming Concepts and Methodology II (CSC 15)
(3)	CSC 28	Discrete Structures for Computer Science (MATH 29 and CSC 20; CSC 20 may be taken concurrently)
(3)	CSC 35	Introduction to Computer Architecture (CSC 15)
(3)	CSC 60	Introduction to Systems Programming in UNIX (CSC 20, CSC 35)

B. Required Mathematics Courses (15 units)

(4)	MATH 30	Calculus I (MATH 29 or four years of high school mathematics which includes two years of algebra, one year of geometry, and one year of mathematical analysis; completion of ELM requirement and Pre-Calculus Diagnostic Test)
(4)	MATH 31	Calculus II (MATH 30 or appropriate high school based AP credit)
(4)	STAT 50	Introduction to Probability and Statistics (MATH 26A, MATH 30, or appropriate high school based AP credit)
(3)	Select one of the following:	
	MATH 32	Calculus III (MATH 31)
	MATH 45	Differential Equations for Science and Engineering (MATH 31)
	MATH 100	Applied Linear Algebra (MATH 26B or MATH 31)
	MATH 102	Number Theory (MATH 31)
	STAT 155	Introduction to Techniques of Operations Research (MATH 31; STAT 50, STAT 103, or STAT 115A; MATH 31 may be taken concurrently) **OR**
	Another advanced math course with prior Computer Science department approval.	

C. Required Science Courses (11-13 units)

(4)	PHYS 11A	General Physics: Mechanics (MATH 30, MATH 31 or equivalent certificated high school courses. MATH 31 may be taken concurrently)
(4)	PHYS 11C	General Physics: Electricity and Magnetism, Modern Physics (MATH 31, PHYS 11A)
(3-5)	Select one of the following (the course chosen cannot also be used to satisfy the General Education B2 requirement):	
	BIO 10	Basic Biological Concepts
	BIO 11	Animal Biology (BIO 10)
	BIO 12	Plant Biology (BIO 10)
	BIO 22	Introductory Human Anatomy (BIO 1, BIO 2, BIO 10, BIO 20)
	BIO 102	The Natural History of Plants (A college biology course or instructor permission)
	BIO 103	Plants and Civilization (BIO 10 or equivalent)
	BIO 104	Physiology of Human Reproduction (BIO 1, BIO 2, BIO 10, BIO 20)
	BIO 115	Introduction to Neuroscience (PSYC 1, PSYC 101, physiology and chemistry background strongly recommended)
	BIO 120	Biology of Aging (BIO 1, BIO 2, BIO 10 or BIO 20)
	CHEM 1A	General Chemistry I (High school algebra (two years) and high school chemistry; or equivalent)
	CSC 148	Modeling and Experimental Design (MATH 31, STAT 50, proficiency in a programming language) (If chosen for a science elective, cannot also be used for a computer science elective)

ECON 141	Introduction to Econometrics (ECON 1A, ECON 1B, ECON 140; ECON 100A or ECON 100B recommended)
ENGR 17	Introductory Circuit Analysis (PHYS 11C, MATH 45; either may be taken concurrently)
ENGR 45	Engineering Materials (PHYS 11A, CHEM 1A; CHEM 1A may be taken concurrently)
PHYS 11B	General Physics: Heat, Light, Sound (MATH 31, PHYS 11A)
PHYS 115	Electronics and Instrumentation (PHYS 11C or PHYS 5B, with instructor permission)

Note: To satisfy the requirements of CAC, the Computing Accreditation Commission of ABET, Inc. which accredits Computer Science programs, students must have taken a total of four courses in a scientific discipline and/or quantitative science. The courses in a scientific discipline must be those typically taken by the majors in that discipline. As part of this requirement a two-semester sequence in a laboratory science must be included (PHYS 11A and PHYS 11C satisfies this requirement). Students ordinarily complete one of the remaining two courses by choosing an appropriate course in General Education Category B2 (BIO 10 is recommended). The second course is expected to be chosen from the list above. Hence, students must have taken a total of four courses in this category.

D. Required Upper Division Courses (37 units)

(3)	CSC 130	Data Structures and Algorithm Analysis (CSC 20, CSC 28; CSC 28 may be taken concurrently)
(3)	CSC 131	Computer Software Engineering (CSC 130; may be taken concurrently)
(3)	CSC 132	Computing Theory (CSC 28, CSC 130; CSC 130 may be taken concurrently)
(3)	CSC 133	Object-Oriented Computer Graphics Programming (CSC 130, CSC 131)
(3)	CSC 134	Database Management and File Organization (CSC 130)
(3)	CSC 136	Programming Languages (CSC 35, CSC 132)
(4)	CSC 137	Computer Organization (CSC 28, CSC 35, CSC 130)
(3)	CSC 138	Computer Networks and Internets (CSC 35, CSC 60, CSC 130)
(3)	CSC 139	Operating System Principles (CSC 60, CSC 137; or equivalents)
(2)	CSC 190	Senior Project: Part I (Senior status; passing score on WPE, CSC 130, CSC 131, and four additional 3-unit CSC upper division courses that fulfill the major requirements excluding CSC 192-195, CSC 198, CSC 199)
(2)	CSC 191	Senior Project: Part II (CSC 190)
(3)	PHIL 103	Business and Computer Ethics
(2)	Select two units from the following:	
	CSC 192	Career Planning (1 unit maximum) (CSC 190, may be taken concurrently)
	CSC 194	Computer Science Seminar (Upper division or graduate status in CSC)
	CSC 195	Fieldwork in Computer Science (Instructor permission)
	CSC 195A	Professional Practice
	CSC 198	Co-curricular Activities in Computer Science
	CSC 199	Special Problems

E. Electives (9 units)

In addition to the required lower-division and upper-division Computer Science courses, Computer Science majors must take three additional elective courses, totaling at least nine (9) units, chosen from undergraduate Computer Science courses numbered CSC 140 or above (excluding CSC 192, CSC 194, CSC 195, CSC 195A, CSC 198, CSC 199). A variety of combinations is acceptable, but it is required that these elective courses be chosen with advisor consultation and approval. With advance written approval from their advisor, the course instructor, and the Department Chair, students with a GPA of 3.0 or better may take graduate courses as electives. In any case students must meet the prerequisite stated in the catalog prior to taking any elective course.

Cooperative Education Program (Work Experience)

The Computer Science Department encourages students to participate in the Cooperative Education Program (Co-op) which provides alternate periods of university study and major-related, paid work experience in private industry or government. The experience will enhance the student's employment prospects upon graduation. Most participants in Computer Science will complete the equivalent of two six-month work periods, one in their junior year and one in their senior year. Students must enroll in CSC 195A (B,C,D) and are awarded a certificate upon satisfactory completion of the two work periods. However, the credits for this course do not replace the curricular requirements of the BS Computer Science degree. Students interested in Co-op should apply in the Career Center (LSN 2000). For information call (916) 278- 6231.

Requirements • Minor

Total units required for Minor: 21

Grade of "C-" or better required in all courses applied to the Computer Science minor.

Admission Requirement

Completion of MATH 29 with a grade of "C-" or better, or passing the ELM at a level qualifying for MATH 30.

Courses in parentheses are prerequisities.

A. Required Courses (12 units)

(3)	CSC 15	Programming Concepts and Methodology I (CSC 10, or programming experience in a high-level programming language)
(3)	CSC 20	Programming Concepts and Methodology II (CSC 15)
(3)	CSC 28	Discrete Structures for Computer Science (MATH 29 and CSC 20; CSC 20 may be taken concurrently)
(3)	CSC 130	Data Structures and Algorithm Analysis (CSC 20, CSC 28; CSC 28 may be taken concurrently)

B. Electives (9 units)

(9) Select nine additional units with faculty approval; at least six units must be upper division courses, and only Computer Science courses which are part of the major may be applied to this nine-unit requirement.

Requirements • Minor Information - Security and Computer Forensics

Total units required for Minor: 21

Grade of "C-" or better required in all courses applied to the Computer Science minor.

Courses in parentheses are prerequisites.

Required Courses (18 units)

(3)	CSC 10	Introduction to Programming Logic (MATH 11 or equivalent) **OR**
	CSC 22	Visual Programming in BASIC (Intermediate Algebra)
(3)	CSC 80	Web Development with HTML/XHTML and Tools (CSC 8, or equivalent computer and Internet experience)
(3)	CSC 114	Fundamentals of Information Assurance & Security (CSC 10, CSC 80)
(3)	CSC 115	Internet Security (CSC 114)
(3)	CSC 116	Cyber Forensics (CSC 114)
(3)	CSC 122	Web Database Applications (CSC 10 or CSC 22; CSC 80, or equivalent)

Requirements • Certificate - Web Development

Total units required for Minor: 21 units.

A grade of "C-" or better required in all courses applied to this certificate program.

Note: Requirements for non-majors interested in mastering a set of skills for creating and managing information on a web server.

Courses in parentheses are prerequisites.

(3)	Select one of the following:	
	CSC 8	Exploring the Internet (Basic computer literacy recommended) **OR**
	CSC 8S	Self-Paced Introduction to Internet Technologies (Basic computer literacy recommended)
(3)	CSC 22	Visual Programming in BASIC (Intermediate Algebra)
(3)	CSC 80	Web Development with HTML/XHTML and Tools (CSC 8 or equivalent computer and Internet experience)
(3)	CSC 120	Web Server Administration (CSC 80 or instructor approval)
(3)	CSC 121	Client-Side Web Programming (CSC 22, CSC 80 or equivalent; or PCSC/CSC Major, CSC 60, CSC 130)
(3)	CSC 122*	Web Database Applications (CSC 10 or CSC 22; CSC 80, or equivalent)
(3)	CSC 123*	Web Application Development (CSC 22, CSC 121, CSC 122 or equivalent experience or PCSC/CSC Major, CSC 60, CSC 134)

* With advisor approval an elective course from CSC or Communication Studies (COMS) can be used in place of either CSC 122 or CSC 123, but not both.

Requirements for Majors: CSC 121 and CSC 123 in addition to completing CSC 60, CSC 130, CSC 134, CSC 138, and CSC 139. **Note:** Prerequisites for majors taking CSC 121 are CSC 60 and CSC 130; prerequisites for majors taking CSC 123 are CSC 60 and CSC 134 (CSC 138 and CSC 139 are recommended as prerequisites and required for the final certificate).

Graduate students who are fully classified in either Computer Science or Software Engineering need only take CSC 121 and CSC 123.

GRADUATE PROGRAMS

The Computer Science Department offers Master's Degree programs in Computer Science and Software Engineering, Certificates of Advanced Study for students enrolled in the Computer Science program, and a Master's Degree joint program in Computer Engineering.

The primary goal of each of these programs is to prepare students to serve as effective professional computer specialists in a society which increasingly depends on computer usage and technology.

A secondary goal is to prepare interested students for research, teaching, or further study toward the Ph.D. in Computer Science. The programs also enable individuals with background in other areas to obtain the skills and knowledge necessary to enter and advance in employment in computer-related industries.

The admission requirements for the Master's in Computer Science and the Master's in Software Engineering are identical. Completion of the Master of Science in Computer Science requires advanced course work in a minimum of three of the following areas: computer architecture/computer engineering, database management systems, information assurance and security, intelligent systems, networks and communications, software engineering, and systems software. Completion of the Master of Science in Software Engineering requires advanced course work in the software engineering area.

Teaching associateships are occasionally available for qualified graduate students; these students assist in instruction of undergraduate courses, supervision of laboratory work, and aid faculty members in research projects. Interested persons should apply in the Department office.

Due to the large number of graduate students in Computer Science who are employed, most graduate level courses are offered in the late afternoon or evening.

Admission Requirements

Admission as a classified graduate student requires:

- a baccalaureate degree;
- a minimum 3.0 GPA in the last 60 units attempted;
- GRE general test;
- mathematical preparation including two semesters of calculus and one semester of calculus-based probability and statistics corresponding to Sacramento State courses MATH 30, MATH 31, STAT 50;
- computer science lower-division preparation including programming proficiency, discrete structures, machine organization, and UNIX and PC-based program development environment proficiency corresponding to Sacramento State courses CSC 15, CSC 20, CSC 28, CSC 35, and CSC 60 and as evidenced by a pass on the graduate student placement test or a baccalaureate degree in computer science; and

- computer science advanced preparation as evidenced by a 3.25 GPA in the following Sacramento State upper division computer science courses or their equivalent elsewhere: CSC 130, CSC 131, CSC 132, CSC 134, CSC 137, CSC 138, CSC 139.

Applicants with deficiencies in the admission requirements area are advised to remove any such deficiencies before applying.

Admission Procedures

Applications are accepted February through March for fall semester and August through September for spring semester. All prospective graduate students, including Sacramento State graduates, must file the following with the Sacramento State Office of Graduate Studies, River Front Center 206, (916) 278-6470:

- an online application for admission;
- two sets of official transcripts from all colleges and universities attended, *other than Sacramento State;* and
- official GRE general test scores.

Applications must be received by April 1 (fall semester) and October 1 (spring semester).

Advancement to Candidacy

Each student must file an application for Advancement to Candidacy, indicating a proposed program of graduate study. This procedure should begin as soon as the classified graduate student has:

- removed any deficiencies in admission requirements;
- completed at least 12 units of graduate level (200 series) Computer Science courses with a minimum 3.0 GPA; and
- passed the Writing Proficiency Examination (WPE) or secured approval for a WPE waiver.

Students must have been advanced to candidacy before they can register for Master's thesis or project. Advancement to Candidacy forms are available in the Office of Graduate Studies. The student fills out the form after planning a degree program in consultation with a Computer Science graduate advisor. The completed form must be signed by the Graduate Coordinator or the Department Chair and is then returned to the Office of Graduate Studies for approval.

Requirements • Master of Science Degree - Computer Science

Units required for MS: 30, including at least 21 units of 200-level and 500-level courses

Minimum GPA: 3.0

Note: Only those courses completed within seven years prior to date of graduation will satisfy course requirements. An outline of degree requirements follows:

Cources in parentheses are prerequisites.

A. Required Courses (13 units)

(3) CSC 201 Programming Language Principles (Fully classified graduate status in Computer Science or Software Engineering)

(3) CSC 204+ Data Models for Database Management Systems (Fully classified graduate status in Computer Science or Software Engineering)

(3) CSC 205 Computer Systems Structure (Fully classified graduate status in Computer Science, Software Engineering or Computer Engineering)

(3) CSC 206 Algorithms and Paradigms (Fully classified graduate status in Computer Science or Software Engineering)

(1) CSC 209 Research Methodology (Fully classified graduate status in Computer Science or Software Engineering, passing score on WPE, completion of at least 12 units of 200-level CSC courses)

+Students whose undergraduate preparation has covered a significant amount of the material in CSC 204 or CSC 205 may be given a waiver by the Department from taking one or more of these courses. In this case, for each course waived with department approval, the student must take three additional units of Restricted Electives, described in Section C below.

B. Breadth Requirement (9 units)

Select one course from three of the following areas:

Computer Architecture/Computer Engineering

CSC 237 Microprocessor Systems Architecture (CSC 205)

CSC 242 Computer-Aided Systems Design and Verification (CSC 205)

CSC 273 Hierarchical Digital Design Methodology (CSC 205, CPE 64 or equivalent)

CSC 280 Advanced Computer Architecture (CSC 205, fully classified graduate status in Computer Science or Software Engineering)

Database Management Systems

CSC 212 Bioinformatics: Data Integration and Algorithms (CSC 130, STAT 50, and graduate status. BIO 10 recommended)

CSC 244 Database System Design (CSC 174 or CSC 204)

Intelligent Systems

CSC 214 Knowledge-Based Systems (Fully classified graduate status in Computer Science or Software Engineering)

CSC 215 Artificial Intelligence (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

CSC 219 Machine Learning (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

Networks and Communications

CSC 255 Computer Networks (CSC 138 or CPE 138)

CSC 258 Distributed Systems (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering and CSC 204)

CSC 275 Advanced Data Communication Systems (CSC 138 or CPE 138 or CSC 205)

Software Engineering

CSC 230 Software System Engineering (Fully classified graduate status in Computer Science or Software Engineering; or fully classified graduate status in Computer Engineering and CSC 131)

CSC 231 Software Engineering Metrics (Fully classified graduate status in Computer Science or Software Engineering)

CSC 232 Software Requirements Analysis and Design (Fully classified graduate status in Computer Science or Software Engineering)

CSC 233 Advanced Software Engineering Project Management (Fully classified graduate status in Computer Science or Software Engineering)

CSC 234 Software Verification and Validation (Fully classified graduate status in Computer Science or Software Engineering; or fully classified graduate status in Computer Engineering and CSC 131)

CSC 235 Software Architecture (Fully classified graduate status in Computer Science or Software Engineering)

CSC 236 Formal Methods in Software Engineering (Fully classified graduate status in Computer Science or Software Engineering)

CSC 238 Human-Computer Interface Design (Fully classified graduate status in Computer Science or Software Engineering)

System Software

CSC 239 Advanced Operating Systems Principles and Design (CSC 205)

CSC 250 Computer Security (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

CSC 251 Principles of Compiler Design (CSC 151 or CSC 201)

C. Restricted Electives (3-6 units)

Prior to taking an elective course, students must obtain approval from their advisor, and either the Graduate Coordinator or the Department Chair.

Students should choose their electives according to the following guidelines:

- One of the following upper division courses: CSC 142, CSC 148, CSC 155, CSC 159, CSC 165, CSC 176, CSC 177 as long as they have not been used towards another degree. (A maximum of 6 undergraduate units may be used in any graduate program.)
- Any 200-level CSC courses not already used to satisfy the Breadth Requirement, with the exception of CSC 295 and CSC 299. Students not required to take CSC 204 or CSC 205 must, for each course waived, take an additional three units in this category.
- Related 200-level courses from outside the Computer Science Department may only be taken with prior department approval and may not have been used in another program.

D. Culminating Requirement (2-5 units)

(2-5) Select one of the following:

CSC 500 Master's Thesis (CSC 209; advanced to candidacy) **OR**

CSC 502 Master's Project (CSC 209; advanced to candidacy)

Students are required to make an oral presentation of their master's project or conduct an oral defense of their master's thesis.

The recommended department-level deadline in each semester for submitting an MS project or thesis signed by the committee chair and its members to the Graduate Coordinator's office is 10 weekdays prior to the University deadline.

Requirements • Master of Science Degree - Software Engineering

Units required for MS: 30, including a software engineering project or thesis with a minimum 3.0 GPA.

This degree provides the student with the ability to specialize in the application of software engineering principles to the development of large and complex computer systems.

The program's courses are structured to satisfy two groups of students: (1) those pursuing an MSSE degree and (2) those interested in individual courses. Individuals wishing to pursue a degree must satisfy the Computer Science graduate program entrance requirements. Those enrolling in individual courses must have an undergraduate degree in Computer Science (or related field) or a minimum of one-year's work experience involving some aspect of software engineering.

The MS Degree in Software Engineering offers, in addition to a core curriculum, advanced studies in the software engineering area. This program covers the entire software application development process from problem definition through requirements, design, implementation, testing, operation, and maintenance.

Note: Only those courses completed within seven years prior to date of graduation will satisfy course requirements. An outline of degree requirements follows:

Courses in parentheses are prerequisites.

A. Required Software Engineering Core Courses (16 units)

(1) CSC 209 Research Methodology (Fully classified graduate status in Computer Science or Software Engineering, or Computer Engineering, passing score on WPE, completion of at least 12 units of 200-level courses in Computer Science)

(3) CSC 230 Software System Engineering (Fully classified graduate status in Computer Science or Software Engineering; or fully classified graduate status in Computer Engineering and CSC 131)

(3) CSC 232 Software Requirements Analysis and Design (Fully classified graduate status in Computer Science or Software Engineering)

(3) CSC 233 Advanced Software Engineering Project Management (Fully classified graduate status in Computer Science or Software Engineering)

(3) CSC 235 Software Architecture (Fully classified graduate status in Computer Science or Software Engineering)

(3) CSC 238 Human-Computer Interface Design (Fully classified graduate status in Computer Science or Software Engineering)

B. Software Engineering Electives (6 units)

Select two from the following:

CSC 231 Software Engineering Metrics (Fully classified graduate status in Computer Science or Software Engineering)

CSC 234 Software Verification and Validation (Fully classified graduate status in Computer Science or Software Engineering; or fully classified graduate status in Computer Engineering and CSC 131)
CSC 236 Formal Methods in Software Engineering (Fully classified graduate status in Computer Science or Software Engineering)

C. Restricted Electives (3-6 units)

Prior to taking an elective course, students must obtain approval from their advisor, and either the Graduate Coordinator or the Department Chair. Students should choose their electives according to the following guidelines:

- Any 200-level CSC courses not already used to satisfy Requirements A and B, with the exception of CSC 295 and CSC 299. An additional three units in this category must be taken if a core course is waived.
- Related 200-level courses from outside the Computer Science Department may only be taken with prior department approval and may not have been used in another program.

D. Culminating Requirement (2-5 units)

Select one of the following:
CSC 500* Master's Thesis (CSC 209; advanced to candidacy) **OR**
CSC 502* Master's Project (CSC 209; advanced to candidacy)

Students are required to make an oral presentation of their master's project or conduct an oral defense of their master's thesis. The recommended department-level deadline in each semester for submitting an MS project or thesis signed by the committee chair and its members to the graduate coordinator's office is 10 weekdays prior to the University deadline.

Requirements • Certificate - Advanced Programs

Minimum required GPA for all courses taken in the program: 3.0

Note: The Certificates in Computer Science program are designed to recognize students who have completed the core graduate courses -- CSC 201, CSC 204, CSC 205 and CSC 206 -- plus additional advanced course work in a specialty area.

These certificates are available only for MS Computer Science matriculated students.

Courses in parentheses are prerequisites.

Certificate in Bioinformatics Technology (12 units)

(3) CSC 212 Bioinformatics: Data Integration and Algorithms (CSC 130, STAT 50, and graduate status; BIO 10 recommended)
(3) CSC 215* Artificial Intelligence (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering) **OR**
CSC 219* Machine Learning (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)
(3) Select one of the following:
CHEM 245 Computational Chemistry (CHEM 140A and CHEM 140B or CHEM 142 or instructor permission)
CSC 244 Database System Design (CSC 174 or CSC 204)
CSC 258* Distributed Systems (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering and CSC 204)
(3) BIO 224 Genomics, Proteomics and Bioinformatics (BIO 184, BIO 222, graduate status or instructor permission)

Certificate in Computer Architecture (15 units)

(3) CSC 142 Advanced Computer Organization (CSC 137 or equivalent)
(3) CSC 280* Advanced Computer Architecture (CSC 205 and fully classified gradaute status in Computer Science or Software Engineering)
(3) Select one of the following:
CSC 237 Microprocessor Systems Architecture (CSC 205)
CSC 242 Computer-Aided Systems Design and Verification (CSC 205)
CSC 273 Hierarchical Digital Design Methodology (CSC 205 or CPE 64 or equivalent)
CSC 275 Advanced Data Communication Systems (CSC 138 or CPE 138 or CSC 205)
CSC 288* Special Topics in Computer Science - Computer Architecture/Computer Engineering
(3) Select one of the following:
CSC 239 Advanced Operating Systems Principles and Design (CSC 205)
CSC 251 Principles of Compiler Design (CSC 151 or CSC 201)
(3) Select one of the following:
EEE 285 Micro-Computer System Design I (CPE 185 or EEE 174)
EEE 286 Micro-Computer System Design II (CPE 186 or EEE 285)

Certificate in Computer Engineering (12 units)

(3) CSC 142 Advanced Computer Organization (CSC 137 or equivalent)
(3) Select one of the following:
CSC 237 Microprocessor Systems Architecture (CSC 205)
CSC 275 Advanced Data Communication Systems (CSC 138 or CPE 138 or CSC 205)
CSC 280* Advanced Computer Architecture (CSC 205 and fully classified graduate status in Computer Science or Software Engineering)
CSC 288* Special Topics in Computer Science - Computer Architecture/Computer Engineering
(3) CSC 242 Computer-Aided Systems Design and Verification (CSC 205) OR
CSC 273 Hierarchical Digital Design Methodology (CSC 205, CPE 64 or equivalent)
(3) EEE 285 Micro-Computer System Design I (CPE 185 or EEE 174) **OR**
EEE 286 Micro-Computer System Design II (CPE 186 or EEE 285)

Certificate in Computer Networks and Communications (9 units)

(3) CSC 255 Computer Networks (CSC 138 or CPE 138)

(6) Select two of the following:

CSC 254 Network Security (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

CSC 258* Distributed Systems (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering and CSC 204)

CSC 275 Advanced Data Communication Systems (CSC 138 or CPE 138 or CSC 205)

CSC 288* Special Topics in Computer Science - Network Communications

Certificate in Data Management Systems (9 units)

(9) Select three of the following:

CSC 244 Database System Design (CSC 174 or CSC 204)

CSC 250 Computer Security (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

CSC 258 Distributed Systems (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering and CSC 204)

CSC 288 Special Topics in Computer Science - Database Management

Certificate in Information Assurance and Security (9 units)

(9) Select three of the following:

CSC 236 Formal Methods in Software Engineering (Fully classified graduate status in Computer Science or Software Engineering)

CSC 250 Computer Security (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

CSC 252 Cryptography Theory and Practice (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

CSC 253 Computer Forensics (Fully classified graduate standing in Computer Science, Software Engineering, or Computer Engineering)

CSC 254 Network Security (Fully classified graduate standing in Computer Science, Software Engineering, or Computer Engineering)

Certificate in Intelligent Systems (9 units)

(3) CSC 215 Artificial Intelligence (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

(6) Select two of the following:

CSC 214 Knowledge-Based Systems (Fully classified graduate status in Computer Science or Software Engineering)

CSC 219 Machine Learning (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering)

CSC 288 Special Topics in Computer Science - Intelligent Systems

Certificate in Software Engineering (9 units)

(9) Select three of the following:

CSC 230 Software System Engineering (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering and CSC 131)

CSC 231 Software Engineering Metrics (Fully classified graduate status in Computer Science or Software Engineering)

CSC 232 Software Requirements Analysis and Design (Fully classified graduate status in Computer Science or Software Engineering)

CSC 233 Advanced Software Engineering Project Management (Fully classified graduate status in Computer Science or Software Engineering)

CSC 234 Software Verification and Validation (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering and CSC 131)

CSC 235 Software Architecture (Fully classified graduate status in Computer Science or Software Engineering)

CSC 236 Formal Methods in Software Engineering(Fully classified graduate status in Computer Science or Software Engineering)

CSC 238 Human Computer Interface Design (Fully classified graduate status in Computer Science or Software Engineering)

CSC 288 Special Topics in Computer Science - Software Engineering.

Certificate in Systems Software (9 units)

(9) Select three of the following:

CSC 239 Advanced Operating Systems Principles and Design (CSC 205)

CSC 244 Database System Design (CSC 174 or CSC 204)

CSC 251 Principles of Compiler Design (CSC 151 or CSC 201)

CSC 258 Distributed Systems (Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering and CSC 204)

One of the above may be replaced by one of the following:

CSC 245 Performance Modeling and Evaluation (Fully classified graduate status in Computer Science or Software Engineering)

CSC 250 Computer Security (Fully classified graduate status in Computer Science or Software Engineering)

CSC 288 Special Topics in Computer Science - Systems Software.

Notes:

- A grade of "C-" or higher is required for all course work in both Child Development Majors.
- Students are strongly recommended to see an advisor early in the program and continuously throughout the program.

Lower Division Courses

CSC 1. Introduction to Computer Science. Fundamental concepts of computers, computation and programming; history and principles of computing; problem solving; input, output; data representation, storage, and file organization; computer hardware, networking and data communication; social, economic and ethical implications; computer security and privacy. Students will solve problems using a programming language. Lecture, two hours; technical activity and laboratory, two hours. **Prerequisite:** Intermediate algebra. **Units:** 3.0.

CSC 1A. Introduction to Computer Science for Advanced Students. Same material as covered in CSC 1 but intended for students who already have significant knowledge of the fundamental concepts of computers and/or computer programming. Students must attend the orientation session during the first class meeting. Two placement tests, one on programming and one on concepts, will be scheduled and used to determine student's prior preparation. **Note:** May be taken by those wishing to obtain credit by examination. Please refer to examination credit guidelines in the University catalog. Not open to students who have received credit for CSC 1 or MIS 5. **Graded:** Credit / No Credit. **Units:** 3.0.

CSC 4A. Introduction to the PC Environment. Introduction to computer hardware and software. Based on the Intel chipset (286, 386, 489, Pentium machines). Topics include: components of computer hardware including boards found inside a typical computer, basic DOS command, application software, simple software installation, program management, file/directory organization, and buying your own computer. **Note:** Does not require any prior knowledge of computers. **Units:** 1.0.

CSC 4B. Introduction to Windows. Introduction to Microsoft Windows. Topics include: using the Program Manager, running Windows and DOS programs, organizing the desktop, customizing Windows and installing Windows software. **Prerequisite:** CSC 4A. **Units:** 1.0.

CSC 4C. Configuring Your PC. Provides the student with enough understanding of the hardware and software PC system operating in a Windows environment to be able to upgrade their computer, ask the right questions from vendors, understand the possible sources of hardware and software conflicts, install new hardware and do advanced installation of new software. **Prerequisite:** CSC 4B. **Units:** 1.0.

CSC 5. Personal Computing. Introduction to the role and use of personal computers. Explanation and hands-on experience with the personal computer, emphasizing the use and relevancy of common software for word processing, filing, spreadsheet analysis, graphics, and communications. Examination of the personal computing milieu and the applications environment. Lecture two hours, technical activity and laboratory two hours. **Units:** 3.0.

CSC 6A. Microcomputer Applications - Word Processing. Microcomputer-based introductory level course in word processing on microcomputers. **Units:** 1.0.

CSC 6B. Microcomputer Applications - Spreadsheets. Microcomputer-based introductory level course in spreadsheet concepts and applications. **Note:** Not open to students receiving credit for MIS 1B. **Units:** 1.0.

CSC 6C. Microcomputer Applications - Database Management. Microcomputer-based introductory level course in database management concepts and applications. **Units:** 1.0.

CSC 8. Introduction to Internet Technologies. Internet applications such as email, instant messaging, file transfer, secure communications, the web, and related tools and protocols. Basics of the web-publishing process and methods used to locate authoritative information on the internet. Webpage design, internet security and emerging technologies on the internet. **Prerequisite:** Basic computer literacy recommended. **Units:** 3.0.

CSC 8S. Self-Paced Introduction to Internet Technologies. Covers the same material as CSC 8, Introduction to Internet Technologies. Provided by electronic means in addition to meetings for orientation, laboratory demonstrations, and tests. Internet applications such as e-mail, instant messaging, file transfer, secure communications, the Web, and related tools and protocols. Basics of the Web-publishing process and methods used to locate authoritative information on the Internet. Webpage design, Internet security and emerging/declining technologies on the Internet. **Prerequisite:** Basic computer literacy recommended. **Graded:** Credit / No Credit. **Units:** 3.0.

CSC 10. Introduction to Programming Logic. Introduction to computer science with an emphasis on programming concepts and methodology. Intended to assist students with little or no programming experience to understand the basic principles of programming logic. Topics include computer hardware and software, problem solving and algorithm development, flow of control, modular design using techniques that can be applied to common programming languages. Lecture two hours, technical activity and laboratory two hours. **Prerequisite:** Math 11 or equivalent **Units:** 3.0.

CSC 15. Programming Concepts and Methodology I. Programming concepts using an object-oriented programming language. Introduction to methodologies for program design, development, testing, and documentation. Topics include program design, algorithm design, number systems, classes and objects, methods (functions), control structures, arrays, and interactive input/output. Lecture two hours, technical activity and laboratory two hours. **Prerequisite:** CSC 10, or programming experience in a high-level programming language. **Units:** 3.0.

CSC 15W. Programming Methodology I Workshop. Designed to assist students in developing a more thorough understanding of programming methodology and problem solving techniques. Activity two hours. **Corequisite:** CSC 15. **Graded:** Credit / No Credit. **Units:** 1.0.

CSC 20. Programming Concepts and Methodology II. Application of object-oriented techniques for systematic problem analysis and specification, design, coding, testing, and documentation. Semester-long project approach emphasizing larger programs. Managing program complexity using abstraction. Introduction to algorithm analysis and Big-O notation. Advanced language features. Basic sorting and searching algorithms. Recursion. Lecture two hours, technical activity and laboratory two hours. **Prerequisite:** CSC 15. **Units:** 3.0.

CSC 20W. Programming Methodology II Workshop. Designed to assist students in developing a more thorough understanding of programming methodology and problem solving techniques. Activity two hours. **Corequisite:** CSC 20. **Graded:** Credit / No Credit. **Units:** 1.0.

CSC 21. Freshman Seminar: Becoming an Educated Person. Introduction to the nature and possible meanings of higher education, and the functions and resources of the University. Designed to help students develop and exercise fundamental academic success strategies and to improve their basic learning skills. Development of information competence and computer literacy. Students interact with fellow students and the seminar leader to build a community of academic and personal support. **Units:** 3.0.

CSC 22. Visual Programming in BASIC. Computer Programming using Visual Basic. Topics include the Visual Basic Integrated Development Environment, visual user interface development, concepts of object-oriented programming, variables, control structures, arrays, functions, subroutines, strings, files, and database access. Applications will be created in areas such as business, games, and multimedia. Lecture two hours, technical activity and laboratory two hours. **Prerequisite:** Intermediate Algebra. **Units:** 3.0.

CSC 25. Introduction to C Programming. Topics include: types, operators, control structures, input/output, arithmetic operations, the C library and preprocessor, functions and parameters, arrays, strings, pointers, and structures. Program design and style will be emphasized. Students will use a compiler. **Note:** Students with significant programming experience should take CSC 60 rather than CSC 25. Lecture two hours, technical activity and laboratory two hours. **Units:** 3.0.

CSC 28. Discrete Structures for Computer Science. Introduction to the essential discrete structures used in Computer Science, with emphasis on their applications. Topics include: counting methods, elementary formal logic and set theory, recursive programming, digital logic and combinational circuits, real number representation, regular expressions, finite automata. **Prerequisite:** MATH 29, CSC 20; CSC 20 may be taken concurrently. **Units:** 3.0.

CSC 35. Introduction to Computer Architecture. Internal representation of numeric and non-numeric data, assembly level machine architecture, addressing modes, subroutine linkage, polled input/output, interrupts, high-level language interfacing, macros and pseudo operations. Lecture two hours, technical activity and laboratory two hours. **Prerequisite:** CSC 15. **Units:** 3.0.

CSC 60. Introduction to Systems Programming in UNIX. Features of the C language commonly used in systems programming, application to systems programming in a UNIX environment. Topics include C preprocessor macros, I/O, bit-manipulation facilities, timesharing system concepts, file permissions, shell script programming, make files and source code control, basic system calls like fork and exec, pointers and dynamic memory allocation, libraries and relocation and linking concepts including assembler handling of symbol tables. Prior knowledge of a C like programming language is presumed. **Prerequisite:** CSC 20, CSC 35. **Units:** 3.0.

CSC 80. Web Development with HTML/XHTML and Tools. Hands-on course covering the processes and guidelines for creating and customizing interactive webpages. Emphasis on use of HTML/XHTML, CSS, and tools to create webpages. HTML/XHTML syntax to create, format, and link documents. Use of tables, graphics, styles, forms, multimedia, and other features in webpages. Effective webpage design and website organization. Lecture two hours and technical activity and laboratory two hours. **Prerequisite:** CSC 8 or equivalent computer and Internet experience. **Units:** 3.0.

CSC 96. Experimental Offerings in Computer Science. When a sufficient number of qualified students apply, one of the staff will conduct a seminar in some topic of computer science. **Note:** May be repeated for credit. **Units:** 1.0-4.0.

CSC 98. Co-Curricular Activities in Computer Science. Students will provide technical assistance in labs, assist instructors in grading course work, or assist in other activities related to the subject matter and concerns of the department. **Graded:** Credit / No Credit. **Units:** 1.0-3.0.

CSC 99. Special Problems. Individual projects or directed reading in specified topics in computer science. **Note:** Open only to students who appear competent to carry on individual work; approval of faculty supervisor and advisor required. May be repeated for credit. **Graded:** Credit / No Credit. **Units:** 1.0-3.0.

Upper Division Courses

CSC 114. Fundamentals of Information Assurance and Security. Topics include the security principle of success, architecture and models, business continuity planning, cryptography, application development security, access control, operating systems security, database security, introduction to computer forensics, web security, Internet security protocols, and security management. Includes projects involving practical computer security tools. **Prerequisite:** CSC 10, CSC 80. **Units:** 3.0.

CSC 115. Internet Security. Internet security problems and discussion of potential solutions: network vulnerabilities and attacks, secure communication and use of cryptography, Internet security protocols and tools to defend against network attacks, network intrusion detection, and wireless network security. Survey and use of software tools for network security. **Prerequisite:** CSC 114. **Units:** 3.0.

CSC 116. Cyber Forensics. Fundamentals of computer forensics and cyber-crime scene analysis including laws, regulations, and international standards; formal methodology for conducting security incident investigations; categories of electronic evidence. Projects involving digital forensic tools. **Prerequisite:** CSC 114. **Units:** 3.0.

CSC 120. Web Server Administration. Managing and maintaining Web servers. Administering open source and commercial Web servers, Web hosting alternatives, professional standards and practices of website hosting, Web server installation, configuration, management, and security. Selecting and using technologies to support professional quality websites. Lecture two hours; technical activity and laboratory two hours. **Prerequisite:** CSC 80. **Units:** 3.0.

CSC 121. Client-Side Web Programming. Client-side Web programming using JavaScript, DHTML, and client-side Web technologies. Event-driven programming, dynamic data types, control structures, and introduction to object-oriented programming and program design. Use of cookies and built-in objects. Validation and processing of forms. Basic features of the Document Object Model. **Prerequisite:** CSC 22 and CSC 80 or equivalent. **Units:** 3.0.

CSC 122. Web Database Applications. Fundamentals of building effective database-driven web applications. Particular emphasis on database access via web interfaces. Introduction to database management systems, their structure and usage, SQL, integrating web applications with databases, design and implementation of dynamic web database applications. **Prerequisite:** CSC 10 or CSC 22; and CSC 80 or equivalent. **Units:** 3.0.

CSC 123. Web Application Development. Developing multi-tiered enterprise-level Web applications. Standards of Web services and other current Web technologies, including XML, AJAX, and server-side programming such as Java EE, .NET, or PHP. Development of Web applications such as those used for e-commerce, e-business, and content management. Format consists of two 1.5 hour lecture/lab sessions per week. **Prerequisite:** CSC 22, CSC 121, CSC 122, or equivalent experience. **Units:** 3.0.

CSC 126. 3D Computer Modeling. Techniques and processes to create 3D computer models and environments. Exercises, assignments and projects designed to build skill levels with 3D computer modeling tools. Demonstrations and workshops in the use of 3D computer modeling software. Critiques, discussion and presentations to develop students' conceptual grasp of 3D computer modeling and virtual reality environments. Example applications in art/new media and computer gaming. **Note:** May be taken twice for credit. **Prerequisite:** CSC 10 or ART 97 or equivalent. **Cross-listed:** ART 142. **Units:** 3.0.

CSC 130. Data Structures and Algorithm Analysis. Specification, implementation, and manipulation of complex data structures: linear/lists, stacks, queues, trees, sets, and graphs. Design and analyze algorithms. Recursion and stack-based memory management. Advanced searching and sorting. NP-completeness. **Prerequisite:** CSC 20, CSC 28; CSC 28 may be taken concurrently. **Units:** 3.0.

CSC 131. Computer Software Engineering. Principles of Software Engineering covering the software development life cycle, including software requirements engineering (elicitation, modeling, analysis and specification), software design, software implementation and testing. Main topics include various software development process models, method and techniques for specifying requirements, architectural and detailed design specification, prototyping, top-down and bottom-up software implementation and testing. Topics also include project management, project documentation and the development of communication skills through written documentation and oral presentation. **Prerequisite:** CSC 130; may be taken concurrently. **Units:** 3.0.

CSC 132. Computing Theory. Introduction to computing theory with examples and applications. Automata and formal languages; language recognition and generation; language hierarchy; deterministic and non-deterministic automata; regular expressions; pushdown automata and context-free grammars; properties of regular and context-free languages; Turing machines; computable and noncomputable functions; decidability. **Prerequisite:** CSC 28, CSC 130; CSC 130 may be taken concurrently. **Units:** 3.0.

CSC 133. Object-Oriented Computer Graphics Programming. Introduction to computer graphics and advanced topics in object-oriented (OO) programming. The OO paradigm is used throughout, utilizing computer graphics as the vehicle for solidifying basic OO concepts, studying the implementation of event-driven systems, and for developing a thorough understanding of advanced OO concepts such as inheritance and polymorphism. Topics include fundamental concepts of object-oriented programming, software design patterns, graphic devices, line and surface drawing, simple 2D and 3D representation, and use of User Interface components. **Prerequisite:** CSC 130, CSC 131. **Units:** 3.0.

CSC 134. Database Management and File Organization. File systems, storage structures and access methods; data modeling; Entity-Relationship analysis and data normalization; design of applications using database technology; elements of commercial database management systems; introduction to transaction processing; introduction to SQL; information interchange and XML; database processing on the Web. **Prerequisite:** CSC 130. **Units:** 3.0.

CSC 136. Programming Languages. Evolution and characteristics of programming languages. Scripting, procedural, object-oriented, functional and logic paradigms. Language specification; interpreters and compilers; virtual machines; parsing techniques. Design and implementation of selected features of programming languages. Programming languages used in the development of intelligent systems, with introduction to Artificial Intelligence. Trends in programming languages. **Prerequisite:** CSC 35, CSC 132. **Units:** 3.0.

CSC 137. Computer Organization. Introduction to computer organization and architecture. Topics include combinational devices, sequential and synchronized circuits, memory, bus structures, input/output and interrupt structures, CPU organization, control unit design and organization, and an introduction to modern processor and memory features. Projects include construction of a complete simple system using a schematic simulator and HDL. **Prerequisite:** CSC 28, CSC 35, CSC 130. **Units:** 4.0.

CSC 138. Computer Networks and Internets. Overview of the fundamentals of computer networks and connections between networks, from the physical layer up through peer-to-peer communications at the application level. Lower layer characteristics including serial vs. parallel, capacity issues, high-speed connections, LAN framing and error handling. LAN vs. WAN characteristics, network architecture and the ISO network model. Internetworking components including LANs, repeaters, routers, bridges, and gateways. Internet addresses, TCP/IP, and the Domain Name System. Common Internet client/server application protocols including SMTP and FTP. Client/Server programming involving sockets. World Wide Web characteristics including CGI and HTTP protocol, Web pages, Web browsers, Web servers, and Applets. Introduction to advanced Web issues such as Web security, search engine operations, and Web database operations. **Prerequisite:** CSC 35, CSC 60, CSC 130. **Cross-listed:** CPE 138; only one may be counted for credit. **Units:** 3.0.

CSC 139. Operating System Principles. Contemporary operating system organization and structure. Topics include: process and thread, concurrency, scheduling, interprocess communication and synchronization, deadlock, real and virtual memory management, device management, file systems, network and distributed operating systems, security and protection. **Prerequisite:** CSC 60, CSC 137; or equivalents. **Units:** 3.0.

CSC 140. Advanced Algorithm Design and Analysis. Algorithm design using dynamic programming, randomization and greedy methods. Analysis using recurrence relations and amortization. String and network-flow algorithms. NP-completeness, reductions, and approximation algorithms. Review of divide-and-conquer design and asymptotic notation. **Prerequisite:** CSC 130. **Units:** 3.0.

CSC 142. Advanced Computer Organization. Design and performance issues of computers: CPU, I/O interface, and memory. Design alternatives for arithmetic functions, CPU internal architecture, instruction set, instruction cycle, I/O, interrupt, direct memory access, and bus and memory hierarchy. CAD tools for schematic capture and simulations. Students will design and simulate a microcomputer. **Prerequisite:** CSC 137 or equivalent. **Cross-listed:** CPE 142; only one may be counted for credit. **Units:** 3.0.

CSC 148. Modeling and Experimental Design. Modeling and simulation techniques; Monte Carlo methods; queuing theory; model formulation, data collection and analysis, experimental design; model verification and validation. **Prerequisite:** MATH 31, STAT 50, and proficiency in a programming language. **Units:** 3.0.

CSC 151. Compiler Construction. Practical approach to compiler design and implementation. Organization of a compiler, algorithms for lexical, syntactic and semantic analysis, recursive descent and/or LALR parsing, organization of symbol tables, error detection and recovery, object code generation. Modular design will be emphasized. **Prerequisite:** CSC 136, may be taken concurrently. **Units:** 3.0.

CSC 154. Computer System Attacks and Countermeasures. Introduction to network and computer security with a focus on how intruders gain access to systems, how they escalate privileges, and what steps can be taken to secure a system against such attacks. Topics include: Perimeter defenses, intrusion detection systems, social engineering, distributed denial of service attacks, buffer overflows, race conditions, trojans, and viruses. **Prerequisite:** CSC 138 or CPE 138. **Units:** 3.0.

CSC 155. Advanced Computer Graphics. Modeling, viewing, and rendering techniques in 3D computer graphics systems. Topics include modeling systems and data structures; polygonal and parametric surface representation; transformations, windowing, clipping and projections in 3D; hidden surface removal algorithms; techniques for realism such as shading, shadows, highlights, and texture; fractals and procedural models; introduction to animation; hardware support for computer graphics; and the application of graphics principles to virtual reality systems and 3D games. **Prerequisite:** CSC 133. **Units:** 3.0.

CSC 159. Operating System Pragmatics. Application of operating system principles to the design and implementation of a multitasking operating system. Students will write an operating system for a computer platform. Topics include: scheduling of processes, control and allocation of computer resources, and user interfacing. **Prerequisite:** CSC 139. **Cross-listed:** CPE 159; only one may be counted for credit. **Units:** 3.0.

CSC 165. Computer Game Architecture and Implementation. Architecture and implementation of computer game systems. Topics include game engine architecture; screen management and rendering control; geometric models; algorithms and data structures for spatial partitioning, occlusion, and collision detection; real-time interactive 3D graphics and animation techniques; behavioral control for autonomous characters; simulation of physical phenomena; sound and music in games; optimization techniques; multi-player games and networking; game development tools and environments. Substantial programming and project work. **Prerequisite:** CSC 130, CSC 133, MATH 30, PHYS 11A. **Units:** 3.0.

CSC 170. Software Requirements and Specification. Analysis and specification of functional and non-functional requirements for real-time and non-real-time software systems in the context of a software development lifecycle. Determining customer and user software requirements and ensuring that specifications are correct, complete, and testable. Includes modeling techniques, methods for representing real-time requirements, and the use of Computer-Aided Software Engineering (CASE) tools to illustrate analysis concepts. **Prerequisite:** CSC 131. **Units:** 3.0.

CSC 171. Software Engineering Project Management. Fundamental issues in the management and economics of a software engineering project in the context of the software development lifecycle. Topics include: techniques for project planning (budgeting and scheduling), controlling (including quality assurance and configuration management), organizing, staffing, and directing a software project (leadership and motivation); and contemporary issues in management. **Prerequisite:** CSC 131. **Units:** 3.0.

CSC 174. Database Management Systems. Topics in database analysis and design, and applications; Extended Entity-Relationship and UML modeling; SQL view, query processing, and query optimization; concurrency control, transaction performance and recovery algorithms; integrity constraints and triggers; functional dependencies and normalization algorithms; application generator technologies; performance and security issues in Internet database processing; introduction to data mining; introduction to database administration. **Prerequisite:** CSC 131, CSC 134. **Units:** 3.0.

CSC 176. Advanced Database Management Systems. Advanced object-relational systems, advanced catalog systems, security mechanisms, distributed database processing, advanced schema design-partitioning, introduction to data warehousing and data mining, materialized views, Internet technologies, parallel query processing, system utilities, database tuning, DBA tools and techniques. **Prerequisite:** CSC 174. **Units:** 3.0.

CSC 177. Data Warehousing and Data Mining. Data warehousing involves data preprocessing, data integration, and providing online analytical processing (OLAP) tools for the interactive analysis of multidimensional data, which facilitates effective data mining. Data mining is the automated extraction of hidden predictive information from databases. Data mining applies concepts and techniques from the fields of databases, machine learning, algorithms, information retrieval, and statistics. Topics include: data warehousing, association analysis, classification, clustering, numeric prediction, and selected advanced data mining topics. **Prerequisite:** CSC 134 and STAT 50. **Units:** 3.0.

CSC 179. Software Testing and Quality Assurance. Testing, verification, validation, and control of real-time and non-real-time software systems in the context of a software development lifecycle. Topics include: unit, integration and system testing; verification and validation (V&V), quality assurance, metrics, and configuration management. **Prerequisite:** CSC 131. **Units:** 3.0.

CSC 180. Intelligent Systems. Theory and implementation of a variety of techniques used to simulate intelligent behavior. Expert systems, fuzzy logic, neural networks, evolutionary computation, and two-player game-tree search will be covered in depth. Knowledge representation, pattern recognition, hybrid approaches, and handling uncertainty will also be discussed. **Prerequisite:** CSC 130, CSC 132, MATH 31, STAT 50 **Units:** 3.0.

CSC 190. Senior Project: Part I. First of a two-course sequence in which student teams undertake a project to develop and deliver a software product. Approved project sponsors must be from industry, government, a non-profit organization, or other area. Teams apply software engineering principles in the preparation of a software proposal, a project management plan and a software requirements specification. All technical work is published using guidelines modeled after IEEE documentation standards. Oral and written reports are required. Lecture one hour, laboratory three hours. **Prerequisite:** Senior status, passing score on the WPE, completion of CSC 130, CSC 131 and four additional 3-unit CSC upper-division courses that fulfill the major requirements (excluding CSC 192-195, 198, 199). **Units:** 2.0.

CSC 191. Senior Project-Part II. Continuation of the group project begun in CSC 190. Teams apply software engineering principles to the design, implementation and testing of their software product. All technical work is published using guidelines modeled after IEEE documentation standards along with an appropriate user manual. Oral and written reports are required. Senior project is completed with the successful delivery, installation and demonstration of the software along with all approved documentation. Lecture one hour, laboratory three hours. **Prerequisite:** CSC 190. **Units:** 2.0.

CSC 192. Career Planning. Designed to help students learn more about the labor market and opportunities in the Computer Science field. Students will examine their interests, consider their goals, and learn how to conduct an effective proactive job search. Strategies for long term career growth will be identified. **Prerequisite:** CSC 190; may be taken concurrently. **Graded:** Credit / No Credit. **Units:** 1.0.

CSC 194. Computer Science Seminar. Series of weekly seminars on Computer Science topics. These topics cover subjects not normally taught in the course of a school year and they range from the very theoretical in Computer Science through applications to presentations by industry on working conditions, real world environment and job opportunities. **Note:** May be repeated for credit. **Prerequisite:** Upper division or graduate status in CSC. **Units:** 1.0.

CSC 195. Fieldwork in Computer Science. Directed observations and work experience in computer science with firms in the industry or public agencies. Supervision is provided by the instructional staff and the cooperating agencies. **Note:** Faculty approval required. May be repeated for credit. **Graded:** Credit / No Credit. **Units:** 1.0-4.0.

CSC 195A. Professional Practice. Supervised employment in a professional engineering or computer science environment. Placement arranged through the Career Center. **Note:** Requires satisfactory completion of the work assignment and a written report. **Prerequisite:** Instructor permission. **Graded:** Credit / No Credit. **Units:** 1.0-12.0.

CSC 195B. Professional Practice. Supervised employment in a professional engineering or computer science environment. Placement arranged through the Career Center. **Note:** Requires satisfactory completion of the work assignment and a written report. **Prerequisite:** Instructor permission. **Graded:** Credit / No Credit. **Units:** 1.0-12.0.

CSC 195C. Professional Practice. Supervised employment in a professional engineering or computer science environment. Placement arranged through the Career Center. **Note:** Requires satisfactory completion of the work assignment and a written report. **Prerequisite:** Instructor permission. **Graded:** Credit / No Credit. **Units:** 1.0-12.0.

CSC 195D. Professional Practice. Supervised employment in a professional engineering or computer science environment. Placement arranged through the Career Center. **Note:** Requires satisfactory completion of the work assignment and a written report. **Prerequisite:** Instructor permission. **Graded:** Credit / No Credit. **Units:** 1.0-12.0.

CSC 196. Experimental Offerings in Computer Science. When a sufficient number of qualified students apply, one of the staff will conduct a seminar in some topic of computer science. **Note:** May be repeated for credit. **Units:** 1.0-4.0.

CSC 198. Co-Curricular Activities in Computer Science. Students will serve in leadership roles in computer science activities, provide tutoring or technical assistance in labs, assist instructors in grading course work, or assist in other activities related to the subject matter and concerns of the department. **Graded:** Credit / No Credit. **Units:** 1.0-3.0.

CSC 199. Special Problems. Individual projects or directed reading in specified topics in computer science. **Note:** Open only to students who appear competent to carry on individual work; approval of faculty supervisor and advisor required. May be repeated for credit. **Graded:** Credit / No Credit. **Units:** 1.0-3.0.

Graduate Courses

CSC 201. Programming Language Principles. Notations for the specification of programming language syntax and semantics; attribute, translational, operational, axiomatic, algebraic, denotational, and action semantics. Applications of programming language syntax and programming language semantics. Use of meta languages to generate executable language definitions for language implementation, program transformation, program property analysis, and rapid software prototyping. Principles of logic, functional, and object-oriented programming languages. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 204. Data Models for Database Management Systems. Database management system (DBMS) architecture; database file organizations and access methods; the relational model and relational algebra; SQL query language; introduction to query optimization; concurrent transaction processing and backup and recovery; applications using embedded SQL, object types, and stored procedures; database analysis and design notations: EER, UML, and XML; web database environments; database security and administration throughout course. **Note:** Not intended for students who have completed CSC 174. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 205. Computer Systems Structure. Overview of computer systems structure, covering hierarchical structure from software and hardware points of view. Concepts of relocation, linking, and loading; hardware-software interfaces from both application program and operating system points of view. Various CPU structures, including RISC and CISC machines, survey of tightly and loosely-coupled architecture, introduction to pipelined, distributed, and parallel systems, computer system communication principles including local and wide-area networks concepts, and various CAD tools and methodologies are introduced. **Prerequisite:** Fully classified graduate status in Computer Science, Software Engineering or Computer Engineering. **Units:** 3.0.

CSC 206. Algorithms and Paradigms. Design and analysis of algorithms. Classical design paradigms including greedy, divide-and-conquer, dynamic programming, and backtracking algorithmic methods. Alternative paradigms of computing including parallel and numerical approaches. Theoretical limits of computation. Selected additional topics such as genetic, approximation, and probabilistic algorithms. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 209. Research Methodology. Research methodology, problem formulation, and problem solving. Orientation to the requirements for Master's Thesis or Project. Presentations on various research topics. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering, passing score on the WPE, completion of at least 12 units of 200-level courses in Computer Science. **Graded:** Credit / No Credit. **Units:** 1.0.

CSC 212. Bioinformatics: Data Integration and Algorithms. The application of information technology and computer science to biological problems, in particular to biomedical science issues involving genetic sequences. Algorithms and their applications to DNA sequencing and protein database search; tools and techniques for data integration to transform genetic sequencing data into comprehensible information to study biological processes. **Prerequisite:** CSC 130, STAT 50, and graduate status; BIO 10 recommended. **Units:** 3.0.

CSC 214. Knowledge-Based Systems. Historical perspective of knowledge-based systems and their relationship to artificial intelligence. Concepts of knowledge representation and automated reasoning. Survey of expert systems in a variety of applications in engineering and other fields. Implementation of expert systems and expert system shells. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 215. Artificial Intelligence. Nature of intelligence and possibility of its realization on digital computers via algorithmic and heuristic programming methods. Knowledge representation. Search procedures. Problem-solving paradigms and simulation of cognitive processes. Machine learning. Natural language understanding, expert systems, and knowledge engineering. Image understanding. Future of artificial intelligence and limits of machine intelligence. **Prerequisite:** Fully classified graduate status in Computer Science, Software Engineering or Computer Engineering. **Units:** 3.0.

CSC 219. Machine Learning. Introduction to major paradigms and methods of machine learning. Inductive learning, explanation-based learning, classifier systems and genetic algorithms, analogical reasoning, case-based learning, connectionist learning, data driven approaches to empirical discovery, and basis of learning theory. Focus is on representative systems that have been built. **Prerequisite:** Fully classified graduate status in Computer Science, Software Engineering or Computer Engineering. **Units:** 3.0.

CSC 230. Software System Engineering. Integration of managerial and technical activities in system engineering that control the cost, schedule, and technical achievement of the developing software system. Application of the principles, activities, tasks and procedures of system engineering and software engineering to the development of a software system. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering; OR fully classified graduate status in Computer Engineering and CSC 131. **Units:** 3.0.

CSC 231. Software Engineering Metrics. Software quality and quantity metrics in software engineering. Measurement theory and metrics. Metrics include management metrics, indirect and direct metrics and predictive metrics. Uses of metrics include software cost and schedule estimates, model calibration, and software productivity measurements. Metrics techniques include the Goal-Question-Metric approach, COCOMO, and function point analysis. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 232. Software Requirements Analysis and Design. Software engineering requirements including elicitation, analysis, specification, verification and management. Emphasizes IEEE software engineering requirements and standards and the concept of operations (ConOps) document. Techniques include structured analysis, use cases and object-oriented analysis. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 233. Advanced Software Engineering Project Management. Advanced methods and procedures for managing a software development project. Includes project planning, scheduling, and cost estimation, project organizational types, staffing and training considerations, leading and motivating computer personnel, and methods for measuring and controlling a project. Emphasizes IEEE software engineering management standards and keys to project success. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 234. Software Verification and Validation. Verification and validation (V & V) techniques to identify and resolve software problems and high-risk issues early in the software lifecycle. Application of V & V to all phases of the lifecycle process. Includes planning and reporting on the V & V effort. Topics also include software quality assurance and software testing. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering; OR fully classified graduate status in Computer Engineering and CSC 131. **Units:** 3.0.

CSC 235. Software Architecture. Software architecture styles. Concepts and activities for software architecture design. Notations, models, and specification languages for software architecture design. Engineering discipline and guidance for software architecture design. Techniques, methods, tools for designing, building, analyzing, and evaluating software architecture. Object-oriented approach for software architecture design. Architecture-based software development. Management of software architecture design. Reuse of software architecture design. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 236. Formal Methods in Software Engineering. Introduction of basic concepts of formal methods in software engineering. Students will learn why and how formal methods should be used in the software development process for delivering a quality product. The following will be discussed: formal-methods-based software life-cycle models; languages for software system specification; modeling and abstraction of software systems; analysis and verification of system properties; software system refinement; formal semantics, program verification; object-orientation of formal methods; systems and tools for the application of formal method, advances of formal methods in software engineering. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering **Units:** 3.0.

CSC 237. Microprocessor Systems Architecture. Microprocessor/microcomputer architecture and hardware/software interfacing design. RISC v. CISC architecture in-depth, case studies of several popular commercial advanced 32-bit microprocessors. Microcomputer firmware architecture is discussed and illustrated with detailed examples. Term project in which students specify, design and build the hardware and firmware of a computer system. **Prerequisite:** CSC 205. **Units:** 3.0.

CSC 238. Human-Computer Interface Design. Issues involved in design of interaction between people and computers. Insight and experience in key issues of HCI design. Emphasis on identifying issues and tradeoffs in interaction design. Development and evaluation of alternative solutions to design problems. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 239. Advanced Operating Systems Principles and Design. Advanced concepts of concurrent processes, concurrent programming and operating systems. Virtual memory management systems, deadlock, file systems, operating system performance measurement and evaluation. **Prerequisite:** CSC 205. **Units:** 3.0.

CSC 242. Computer-Aided Systems Design and Verification. Design and verification methodology using hardware description and verification languages (HDVLs). Advances in IC chip design; introduction to HDVLs such as System Verilog; HDVL language basics including data types, arrays, structures, unions, procedural blocks, tasks, functions, and interface concepts; design hierarchy; verification planning and productivity; verification infrastructure; guidelines for efficient verification of large designs; assertion-based verification; comprehensive computer-related design projects. **Prerequisite:** CSC 205. **Units:** 3.0.

CSC 244. Database System Design. Topics in the design and implementation of database management systems. Database system concepts and architectures; query compiler, query processing algorithms, logical and physical query plans, query optimization; recovery, concurrency control; transaction management in centralized database management systems and distributed database management systems. Also exploration of current research directions, issues, and results related to databases and data management. **Prerequisite:** CSC 174 or CSC 204. **Units:** 3.0.

CSC 245. Performance Modeling and Evaluation. Performance and cost measures; software and hardware performance monitors; data reduction and evaluation; analytic and simulation models of hardware and program behavior; performance-cost trade-offs and resource allocation. **Prerequisite:** Fully classified graduate status in Computer Science or Software Engineering. **Units:** 3.0.

CSC 250. Computer Security. Principles and technologies behind computer security. Introduction to encryption and decryption; security mechanisms in computer programs, operating systems, databases, and networks; administration of computer security, and legal and ethical issues. **Prerequisite:** Fully classified graduate standing in Computer Science, Computer Engineering, or Software Engineering. **Units:** 3.0.

CSC 251. Principles of Compiler Design. Programming language translation, lexical analysis, syntax analysis including LR, LALR, and predictive techniques. Semantic analysis including semantic specification. Code generation and optimization including control and data flow analysis. Storage management. Error detection and recovery. **Prerequisite:** CSC 151 or CSC 201. **Units:** 3.0.

CSC 252. Cryptography Theory and Practice. Introduction to design and analysis of cryptographic systems. Symmetric cryptography: block ciphers and secure hash functions. Asymmetric cryptography: key exchange and public-key systems. Authentication and encryption in an adversarial model. Simple cryptanalysis. Protocol design and analysis. **Prerequisite:** Fully classified graduate status in Computer Science, Computer Engineering, or Software Engineering. **Units:** 3.0.

CSC 253. Computer Forensics. Structured security incident investigations internal and external; emphasis on analysis of electronic evidence and proper audit; utilization of scientific aids in obtaining information from computing devices; legal electronic evidence. **Prerequisite:** Fully classified graduate status in Computer Science, Computer Engineering, or Software Engineering. **Units:** 3.0.

CSC 254. Network Security. In-depth study of network security problems and discussion of potential solutions. Topics include: network vulnerabilities and attacks, secure communication, Internet security protocols and tools to defend against network attacks, network intrusion detection, and wireless network security. Survey and demonstration of software tools used for network security. **Prerequisite:** Fully classified graduate status in Computer Science, Computer Engineering, or Software Engineering. **Units:** 3.0.

CSC 255. Computer Networks. Computer networking fundamentals with emphasis on higher level protocols and functions. Network design considerations, software design and layering concepts, interface design, routing and congestion control algorithms, internetworking, transport protocol design, and end-to-end communication, session and application protocols. Specific examples of commercial and international standards. **Prerequisite:** CSC 138 or CPE 138. **Units:** 3.0.

CSC 258. Distributed Systems. Distributed system architectures, distributed object model, component-based design, time and global states, coordination and agreement, distributed transactions and concurrency control, replication, security, distributed multimedia systems, message passing and distributed shared memory, Web services and Service-Oriented Architecture (SOA), Grid computing. Emphasis on scalability, manageability, security, and dependability of distributed systems. **Prerequisite:** CSC 204 and fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering. **Units:** 3.0.

CSC 273. Hierarchical Digital Design Methodology. Advanced logic modeling, simulation, and synthesis techniques. Topics include modeling, simulation, and synthesis techniques, using Hardware Description Language (HDL's), Register Transfer Level (RTL) representation, high-level functional partitioning, functional verification and testing, computer-aided logic synthesis, logical verification and testing, timing and delay analysis, automated place and route process, and design with Application Specific Integrated Circuits (ASICs) and programmable logic. **Prerequisite:** CSC 205, CPE 64, or equivalent. **Cross-listed:** EEE 273; only one may be counted for credit. **Units:** 3.0.

CSC 275. Advanced Data Communication Systems. Fundamental concepts, principles and issues of data communication systems. The ISO/OSI reference model is used as a vehicle for discussion and emphasizes lower layer of the model. Specific topics include: motivation and objectives, layered architectures, physical layer principles and protocols, data link and medium access control principles and protocols, circuit, packet and cell switching, local area network design principles and performance comparisons, high speed networking, introduction to wide area network architectures. Typical examples and standards are cited for point-to-point, satellite, packet radio and local area networks. **Prerequisite:** CSC 138 or CPE 138 or CSC 205. **Units:** 3.0.

CSC 280. Advanced Computer Architecture. Introduction to parallel architecture covering computer classification schemes, fine and coarse grain parallelism, processor interconnections, and performance issues of multiprocessor systems. Includes parallel and pipelined instruction execution, structure of multiprocessor systems, memory hierarchy and coherency in shared memory systems, programming issues of multiprocessor systems, arithmetic pipeline design, and design for testability. **Prerequisite:** CSC 205 and fully classified graduate status in Computer Science or Software Engineering. **Cross-listed:** EEE 280; only one may be counted for credit. **Units:** 3.0.

CSC 288. Special Topics in Computer Science. Contemporary topics in computer science will be offered as needed. Topics offered: **Units:** 3.0.

CSC 295. Fieldwork. Directed observations and work experience in computer science with firms in the industry or public agencies. Supervision is provided by the instructional staff and the cooperating agencies. **Note:** Faculty approval required. May be repeated for credit. **Graded:** Credit / No Credit. **Units:** 1.0-3.0.

CSC 296. Experimental Offerings in Computer Science. When a sufficient number of qualified students are interested, one of the staff will conduct a seminar on some topic of Computer Science. **Note:** May be repeated for credit. **Units:** 1.0-4.0.

CSC 299. Special Problems. Any properly qualified student who wishes to pursue a problem of his own choice may do so if the proposed subject is acceptable to the member of the staff with whom he/she works and to his/her advisor. **Note:** May be repeated for credit. **Graded:** Credit / No Credit. **Units:** 1.0-3.0.

CSC 500. Master's Thesis. Completion of a thesis approved for the Master's degree. **Note:** May be repeated for credit. **Prerequisite:** CSC 209; advanced to candidacy. **Graded:** Thesis in Progress. **Units:** 1.0-6.0.

CSC 502. Master's Project. Completion of a project approved for the Master's degree. **Prerequisite:** CSC 209; advanced to candidacy. **Graded:** Thesis in Progress. **Units:** 1.0-3.0.