THE UNIVERSITY OF UTAH

Computer Science

Undergraduate Handbook

2008-2009

Computer Science Undergraduate Handbook

University of Utah
School of Computing
50 S Central Campus Dr.
Merrill Engineering Building (MEB) RM 3190
Salt Lake City, UT 84112-9205

(801) 581-8224 (voice)
(801) 581-5843 (fax)

info@cs.utah.edu
www.cs.utah.edu

2008–2009 Academic Year

The School of Computing offers a Bachelor of Science degree in Computer Science. The undergraduate program begins with a set of four courses that give students a solid background in object-oriented programming while exposing them to the breadth of issues that arise in computer science. Students then take seven core courses in discrete mathematics, software engineering, computer organization, algorithms and data structures, software systems, and theory. They build on this background by choosing seven electives from the breadth of the School's course offerings (which includes advanced courses in theoretical computer science, scientific computing, artificial intelligence, databases, operating systems, computer networks, programming languages, graphics, computer architecture, and digital design). Each student's undergraduate program is capped with a senior project. Along with an in-depth study of computing, the curriculum encompasses a general education in mathematics, science, and the humanities.

The School also offers a combination BS/MS degree for students who wish to complete both the Bachelor and Master of Science degrees in a total of five years, and a minor in Computer Science for students who want to use computers in another field. In addition, selected service courses are offered to provide an introduction to the use of computers as tools for students of many backgrounds and interests.

A Bachelor of Science in Computer Engineering is jointly offered by the School of Computing and the Department of Electrical and Computer Engineering. Information about that program is available in a separate handbook or from www.ce.utah.edu.

The School of Computing also offers a Computer Science Bachelors Degree with an emphasis on Entertainment Arts and Engineering. The focus of this track is to provide interested students with the necessary skills and knowledge to enter the arena of computer gaming and animation.

The University of Utah is committed to policies of equal opportunity, affirmative action, and nondiscrimination. The University seeks to provide equal access to its programs, services, and activities for people with disabilities. Reasonable prior notice is needed to arrange accommodations.

(The latest version of this handbook is available online at www.cs.utah.edu/dept/handbooks/current.)

Faculty

Rajeev Balasubramonian, Ph.D.	Assistant Professor
Adam Bargteil, Ph.D.	Assistant Professor
Martin Berzins, Ph.D.	Professor
Richard Brown, Ph.D.	Professor
Erik Brunvand, Ph.D.	Associate Professor
John Carter, Ph.D.	Associate Professor, Adjunct
Elaine Cohen, Ph.D.	Professor
Sarah Creem-Regehr, Ph.D.	Associate Professor, Adjunct
Hal Daume, Ph.D.	Assistant Professor
Alan L. Davis, Ph.D.	Professor
H. James de St. Germain, Ph.D.	Assistant Professor, Clinical
Sam Drake, Sc.D.	Associate Professor, Research
Matthew Flatt, Ph.D.	Associate Professor
Thomas Fletcher, Ph.D.	Assistant Professor, Research
Juliana Freire, Ph.D.	Associate Professor
Guido Gerig, Ph.D.	Professor
Ganesh Gopalakrishnan, Ph.D.	Professor
Charles Hansen, Ph.D.	Professor
Thomas C. Henderson, Ph.D.	Professor
Lee A. Hollaar, Ph.D.	Professor
John M. Hollerbach, Ph.D.	Professor
Stephen Jacobsen, Ph.D.	Professor Research
Peter Jensen	Assistant Instructor, Clinical
Christopher R. Johnson, Ph.D.	Distinguished Professor
Sneha Kasera, Ph.D.	Assistant Professor
Robert Kessler, Ph.D.	Professor
Mike Kirby, Ph.D.	Associate Professor
Robert McDermott, Ph.D.	Associate Professor, Adjunct
Matthew Might, Ph.D.	Assistant Professor
Chris Myers, Ph.D.	Professor, Adjunct
Steven Parker, Ph.D.	Assistant Professor, Adjunct
Erin Parker, Ph.D.	Assistant Professor, Clinical
John Regehr, Ph.D.	Assistant Professor
Richard F. Riesenfeld, Ph.D.	Professor
Ellen M. Riloff, Ph.D.	Associate Professor
Peter Shirley, Ph.D.	Professor, Adjunct
Kris Sikorski, Ph.D.	Professor
Claudio Silva, Ph.D.	Associate Professor
Konrad Slind, Ph.D.	Assistant Professor
Kenneth Stevens, Ph.D.	Associate Professor, Adjunct
Tolga Tasdizen, Ph.D.	Assistant Professor, Adjunct
William B. Thompson, Ph.D.	Professor
Xavier Trichoche, Ph. D.	Assistant Professor, Adjunct
Suresh Venkatasubramanian, Ph.D.	Assistant Professor
Ingo Wald, Ph.D.	Assistant Professor, Adjunct
Ross Whitaker, Ph.D.	Associate Professor
Joseph L. Zachary, Ph.D.	Professor, Clinical

Emeritus Faculty

David Hanscom, Ph.D.	Emeritus Professor
Robert R. Johnson, Ph.D.	Emeritus Professor
Gary E. Lindstrom, Ph.D.	Emeritus Professor
Kent Smith, Ph.D.	Emeritus Professor
Frank Stenger, Ph.D.	Emeritus Professor

Administration

Martin Berzins	Director	585-1545
Robert Kessler	Associate Director	581-4653
John Carter	Associate Director, External Relations	585-5474
Jim de St. Germain	Director, Undergraduate Studies	585-3352
Joe Zachary	Director, Educational Programs	581-7079
John Regehr	Director, Combined BS/MS Program	581-4280
Erik Brunvand	Director, Graduate Studies	585-1883
Mike Kirby	Director, Computational Engineering and Science	581-8579
Claudio Silva	Director, Graphics/Visualization MS Track	587-7588
John Hollerbach	Director, Robotics MS Track	585-6978

Academic Advising

Kelly Olson	Undergraduate Academic Counselor	581-8225
Karen Feinauer	Graduate Counselor	581-8224

A complete staff listing can be found at www.cs.utah.edu/people.html.

Contents

1 The Computer Science Major and Minor

The School of Computing offers a Bachelor of Science in Computer Science. This is a software-oriented degree whose requirements include 19 computer science courses. A student must be admitted as a computer science major by the School in order to take upper-division courses and pursue the computer science degree. The School also offers a five year combination Bachelors/Masters (BS/MS) degree, as well as a minor in computer science consisting of six computer science courses.

Interested students are encouraged to explore the new Entertainment Arts and Engineering (EAE) track through the computer science degree. This track requires courses which are relevant to computer game designers and computer animators, including drawing courses and film courses. The EAE track results in a computer science degree, with an emphasis on the entertainment arts.

Incoming engineering students should consider the Engineering LEAP (E-LEAP) program organized by the Office of Undergraduate Studies. It is a year-long cohort program set up to allow students to take several classes together during their first year on campus. Two of these classes are seminar courses that satisfy University General Education and Diversity requirements. The program also includes sections of Writing, Calculus, and Physics. For more information on E-LEAP, contact the College of Engineering Advisor at 585-7769.

1.1 Becoming a Computer Science Major

Any student can become a computer science pre-major by informing the University Registrar or the School of Computing Academic Counselor. It is advisable to do this early to ensure receiving information about the major and staying advised of any changes that may be made in degree requirements. Declaration as a pre-major will also enable participation in activities associated with the degree program such as the Undergraduate Student Advisory Committee.

In order to become a full major, a student must first complete the courses required of pre-majors and then apply for full major status. An application should be obtained from the School of Computing web page or office during the semester when the student expects to complete these requirements. One may not pre-register for any upper division classes in Computer Science without first being admitted as a full major or a minor. Applications for admission are reviewed at the end of Spring and Summer semesters.

To be admitted to full major status, a student must have:

1. An average Computer Science GPA of at least 2.8 and a minimum grade of C- in the following classes or their equivalents. None of these classes may be taken on a credit/no-credit basis.

    - Computer Science 1060, 1410, and 2420
    - Mathematics 1210 and 1220 (or Math 1250 and 1260)

2. A grade of CR in CS 1010 (a credit/no-credit class).
3. A cumulative University of Utah grade point average of 2.5 or higher.

If credit is granted for any of the above classes based on advanced placement test scores or courses taken at other schools, appropriate grades will be assigned for our use in this calculation. Check with the Undergraduate Academic Counselor for details.

1.2 Undergraduate Advising

The School of Computing has an Undergraduate Academic Counselor (Kelly Olson, 3190 MEB, ugrad-help@cs.utah.edu, 581-8225) who is available to answer questions regarding schedule plans, registration for Computer Science classes, degree requirements, recent School of Computing actions, or any problems the student may be experiencing. Students should visit the Academic Counselor at least once a year to verify that they are on track for graduation. Please call or email to set up an appointment.

The School also has a Faculty Advisor, H. James de St. Germain (germain@cs.utah.edu), who can answer questions about any of the above, as well as more technical issues, such as career decisions and equivalence of transfer classes. The School of Computing receptionist (MEB 3190, 581-8224) will be happy to set up an appointment for you to meet with either the Faculty Advisor or the Academic Counselor.

1.3 Requirements for the Bachelor of Science Degree

The computer science degree can be completed in four full-time years of study if the student is capable of completing the computer science and calculus pre-major sequences during the freshman year. If a student must instead take preparatory classes as a freshman, more than four years may be required to earn a degree. In any event, it is important to take the required pre-major classes early to allow advancement to full major status as soon as possible.

Below is a list of all the requirements to graduate with a Bachelors of Computer Science. These requirements are summarized in Section 1.3.2.

1. **General Education Requirements**: The General Education requirements are described in the University of Utah General Catalog. The requirements for computer science majors are more specific.

    a. The University Writing requirement is satisfied by either Writing 2010 or ESL 1060 (for students who speak English as a second language).

    b. The Quantitative Reasoning requirement is satisfied by Math 1210 and either Math 1210 or 1250, which are required for computer science pre-majors.

    c. The two-course requirement in physical and life sciences is satisfied by classes required for the major.

    d. Students must take two intellectual explorations courses in each of fine arts, humanities, and social sciences[1]. Two of these six courses must be upper division. One should meet the University Diversity requirement, and one should meet the University International requirement. Students should consult with the Academic Counselor to be sure they select appropriate classes to satisfy these requirements with a minimum number of credit hours[2].

    e. The American Institutions requirement can be satisfied by taking one of Economics 1740, History 1700, Honors 2212, or Political Science 1100.

1 Computer Science students should also consider CS 1050, Computers in Society, as one of their Social/Behavioral Science General Education classes. The course focuses on the social issues that surround the increasingly pervasive roles that computers play in society.

2 It is possible to take some classes that can satisfy several General Education requirements at once. Please ask the Computer Science Advisor about such options.

2. **University Graduation Requirements**: The University graduation requirements for the Bachelor of Science degree are described in the University of Utah General Catalog.

   a. The Communication/Writing requirement: Computer Science majors must take either WRTG 3015 (Professional/Technical Writing), WRTG 3014 (Writing in the Sciences), WRTG 3012 (Writing in the Social Sciences), or Honors 3200 (Writing at a Research University) which may be taken by students participating in the University Honors Program. This class should be taken prior to taking the Computer Science Senior Project course.

   b. The Quantitatively Intensive course requirement is satisfied by CS 3810 and CS 4150, which are required for computer science majors.

   c. The Diversity requirement can be satisfied by taking a course from the approved list as part of the intellectual explorations requirement.

   d. Students must complete a minimum of 122 semester hours of course work. At least 40 of the 122 hours must be upper division classes. Upper division classes are numbered 3000 or above. Credits from two-year colleges will not count toward University upper division hours. At least 30 of the total credit hours and 20 of the last 30 hours must be taken at the University.

3. **Math, Science, and Engineering Requirements**: Seven classes in math, science, and/or engineering are required. A grade of C- or better must be earned in each of these courses:

   a. Mathematics 1210 (Calculus I)

   b. Mathematics 1220 or 1250 (Calculus II)

   c. Physics 2210 (Physics I)

   d. Four additional courses, each of which must be at least three semester hours, chosen from among the following:

      - Any class (other than a computer science class) from the Colleges of Engineering, Mines, or Science that requires Calculus II as a prerequisite or corequisite[3].
      - Physics 2220 (Physics II)
      - Biology 1210 (General Biology)
      - Chemistry 1210 (General Chemistry)

3 Note that Math 2250 covers the same material as Math 2270 and 2280, although in less depth. Hence, if 2250 is used as one of these math/science electives, neither 2270 nor 2280 may be counted. Students should take the prerequisites of computer science electives into consideration when planning how to satisfy this requirement.

4. **Computer Science Requirements**: A minimum of 19 computer science classes must be taken. Section 1.3.1 gives an example four-year degree program leading to a Bachelors Degree in computer science. Section 1.5 summarizes the prerequisites for computer science courses.

    a. Required classes:

        - CS 1010 Introduction to Unix
        - CS 1060[4] Explorations in Computer Science
        - CS 1410 Introduction to Computer Science I
        - CS 2420 Introduction to Computer Science II
        - CS 2100 Discrete Structures
        - CS 3500 Software Practice I
        - CS 3505 Software Practice II
        - CS 3810 Computer Organization
        - CS 4150 Algorithms
        - CS 4400 Computer Systems

    b. Theory restricted elective. One of the following[5] must be completed.

        - CS 3100 Models of Computation
        - CS 3200 Scientific Computing

    c. Electives.

        - Seven additional Computer Science classes numbered 3000 or higher, totaling at least 21 semester hours, must be taken. CS 5010/20 and seminars may not be counted. Only one independent study class of up to 3 credit hours (with special permission) may be counted.

    d. Capstone Requirement. One of the following must be completed.

        - CS 4500 Software Engineering Lab
        - CS 4970 Bachelors Thesis[6]

    e. Exit Survey. Computer Science majors are required to complete an exit questionnaire before they will be signed off for graduation. This should be done during the final semester of undergraduate studies.

    f. Duplication of Credit. No single class may be counted toward more than one of the requirements listed above.

4 CS 1060 may be waived by advanced students. Talk to the Director of Undergraduate Studies for the School of Computing.

5 If both of these classes are taken, the second will be counted as a Computer Science elective.

6 Students choosing the thesis option must get special permission from the faculty. Such approval is usually obtained by finding a faculty advisor in the Spring of the Junior year. This option is intended for students who are considering graduate school. The Bachelors Thesis can also be used toward a portion of the thesis requirement for a BS/MS program in Computer Science. For more information about the joint BS/MS program, please see the director of the BS/MS program in the School of Computing.

5. **Continuing Performance Requirements**: All computer science, mathematics, science, engineering, and writing courses taken to satisfy the requirements listed above must be taken for a grade and must be passed with a C- or better (except for CS 1010, in which a grade of CR is required). A student may repeat courses necessary for the major only one time (pre-major courses may be taken as many times as necessary to gain full-major status).

   To remain in good standing and graduate, a student must maintain a cumulative grade point average at the University of 2.3 or higher, and also maintain a grade point average (GPA) of 2.3 in computer science classes taken at the University. Students whose GPA in either of these categories falls below 2.3 are notified that they are on probation and will be given conditions for a return to good standing. Normally, these conditions must be satisfied during the next two semesters, excluding summers. Students failing to meet their probationary conditions are dropped from the rolls of the major.

   All students admitted as full majors are placed on probationary status. If a student's GPA in either of the above categories is below 2.3 after the first year during which they take upper level computer science classes, the student is dropped from the rolls of the major.

   Students are expected to complete all requirements for their degree within four years of acceptance to full major status. Students not making satisfactory progress toward their degrees may be dropped from the rolls and declared inactive. The determination that a student is not making satisfactory progress is made in one of two ways. Either (1) the student has not completed a computer science course for a period of one year, or (2) there is no reasonable way in which the student can complete all degree requirements by the end of the required period of time.

   In order to be reinstated from inactive status or from being dropped due to low GPA, students must petition the Computer Science Undergraduate Committee. Reinstated students proceed under the latest graduation requirements.

   If personal circumstances prevent completion of all degree requirements within four years of acceptance as a full major, a student may request an extension of a specific duration and submit a revised schedule of completion.

Sections 1.3.1 and 1.3.2 describe a sample plan of study for achieving a degree in four years and show the required courses for a Bachelors of Science in Computer Science.

1.3.1 Example Computer Science Degree Program of Study

	Fall		Spring	
Freshman	CS 1010 (Unix)	(.5)	CS 2420 (Intro to CS II)	(4)
	CS 1060[7] (Explorations in CS)	(3)	Math 1220 (Calc II)	(4)
	CS 1410 (Intro to CS)	(4)	Physics 2210	(4)
	Math 1210† (Calc I)	(4)	Gen Ed*	(3)
	Writing 2010	(3)		
		(14.5)		(15)
Sophomore	CS 3500 (Soft Practice I)	(4)	CS 3505 (Soft Practice II)	(3)
	CS 3810 (Comp Org)	(4)	CS elective	(3)
	CS 2100 (Discrete Structures)	(3)	Math/Sci/Eng elective	(3)
	Writing 3015	(3)	Gen Ed	(3)
			Free elective	(4)
		(14)		(16)
Junior	CS 4400 (Comp Systems)	(4)	CS 4150 (Algorithms)	(3)
	CS 3100[8] (Theory Elective)	(3)	CS elective	(3)
	Math/Sci/Eng elective	(3)	CS elective	(3)
	Gen Ed*	(3)	Math/Sci/Eng elective	(3)
	Gen Ed*	(3)	Gen Ed*	(3)
		(16)		(15)
Senior	CS elective	(3)	CS 4500[9]	(3)
	CS elective	(3)	CS elective	(3)
	CS elective	(3)	Math/Sci/Eng elective	(3)
	Gen Ed*	(3)	Gen Ed*	(3)
	Free elective	(4)	Free elective	(4)
		(16)		(16)

* One GE course should be used to meet the American Institutions requirement.

Table 1: Sample Degree Program of Study

This table gives an eight-semester example program leading to a B.S. in Computer Science. It is meant only as a guide, since the scheduling of electives and General Education classes depends upon which ones are selected. This schedule assumes adequate high school preparation in mathematics; it is not advisable to take Physics 2210 without some previous training in calculus. Note that Math 1210 and CS 1410 must be taken during the fall semester in order to complete the required pre-major classes during the first year, unless the student has advanced placement credit or takes 2420 in during their first summer semester.

7 CS 1060 is required but can be waived for sufficiently advanced students. You must meet with the Director of Undergraduate Studies for the School of Computing. CS 1410 is also offered in the Spring with CS 2420 offered in the Summer. For students new to Computer Science, taking 1060, 1410, and 2420 in consecutive semesters (Fall, Spring, Summer) can ease the transition into the major.

8 Please note, that if you wish to take CS 3200 for your theory elective, the course is offered in the spring and a CS elective should be taken in the Fall.

9 CS 4500 can be taken only by students who are on track to graduate before the next offering of the class.

1.3.2 Requirement Summary Computer Science Major

PRE-MAJOR REQUIREMENTS:
**Take all 6 courses. You must earn a C- or better in each course, and a 2.8 average GPA among the following courses:*

1. CS 1010, Intro to Unix ________________ (.5)
2. CS 1060, Explorations in CS ________________ (3)
3. CS 1410, Comp. Science. I ________________ (4)
4. CS 2420, Comp. Science. II ________________ (4)
5. Math 1210, Calculus I ________________ (4)
6. Math 1220, Calculus II ________________ (4)

GENERAL EDUCATION REQUIREMENTS:

1. Wrtg 2010 or ESL 1050+1060 ____________________
2. Wrtg 3012 or 3014 or 3015 ____________________

3. American Institutions* (AI) ______________________
*(choose one of: Econ 1740, History 1700, Hon 2212, Political Science 1100)

A total of 6 courses must be taken in the following 3 areas: Fine Arts, Humanities, Behavioral/Social Science, take two courses from each area. Two of those six classes must be upper division (3000 level or above), one must satisfy the university Diversity requirement and one must satisfy the university International requirement.

1. Fine Arts (FF): ______________________________
2. Fine Arts (FF): ______________________________

3. Humanities (HF): ______________________________
4. Humanities (HF): ______________________________

5. Social/Behavioral Science (BF): ________________
6. Social/Behavioral Science (BF): ________________

- Upper Division 1 ☐
- Upper Division 2 ☐
- Diversity (DV) ☐
- International (IR) ☐

MATH / SCIENCE / ENGINEERING ELECTIVES:
*(Physics 2210 required. Plus 4 additional electives, each must be at least 3 credits. Choose **from Biology 1210, Chemistry 1210, Physics 2220**, or any non-CS class that has Math 1220 (Calculus II) as a prerequisite or co-requisite.)*

1. Physics 2210, Physics for Scie & Engrs I____ (4)
2. ___________ , ________________________ (   )
3. ___________ , ________________________ (   )
4. ___________ , ________________________ (   )
5. ___________ , ________________________ (   )

The Following Requirements are Restricted to FULL Majors:

CS MAJOR REQUIREMENTS:
* CS 2100 can be taken be pre-majors

1. CS 2100*, Discrete Structures ______________ (3,F)
2. CS 3500, Software Practice I ______________ (4,F)
3. CS 3505, Software Practice II ______________ (3,S)
4. CS 3810, Comp. Organization ______________ (4,F)
5. CS 4150, Algorithms ______________ (3,S)
6. CS 4400, Comp. Systems ______________ (4,F)

CS ELECTIVES:
****Choose seven CS courses, 3000-level or above.***
****Seminars may not be counted. Only one Independent Study class can count in this area.***

1. CS ________ , ______________________________ (   )
2. CS ________ , ______________________________ (   )
3. CS ________ , ______________________________ (   )
4. CS ________ , ______________________________ (   )
5. CS ________ , ______________________________ (   )
6. CS ________ , ______________________________ (   )
7. CS ________ , ______________________________ (   )

THEORY ELECTIVE:

Choose ONE:

CS 3100, Models of Computation ___________________ (3,F)
or
CS 3200, Scientific Computing ____________________ (3,S)

CAPSTONE REQUIREMENT:

Choose ONE:

CS 4500, Software Engin. Lab ____________________ (3,S)
or
CS 4970, Bachelors Thesis ______________________ (   )

1.4 Entertainment Arts and Engineering Track

The School of Computing provides a specialty track through the Computer Science program with an emphasis on Entertainment Arts and Engineering (**EAE**). The EAE track is a joint program between the School of Computing and the College of Fine Arts (currently the fine arts side is housed in the Division of Film Studies). The purpose of this program is to provide an undergraduate, interdisciplinary academic path for those students that wish to have careers in the digital entertainment industry (video games, digital animation, computer generated special effects, etc.). There are currently two specified tracks: video games and animation, differing by the junior year Game/Animation series of classes.

The key feature of this program is its interdisciplinary nature. Students from both CS and Fine Arts will take common classes throughout their undergraduate years, culminating in a year long senior project where students will build a video game or animation from the ground up.

Students in the program can choose either a bachelors degree in Film Studies or Computer Science. The requirements listed below are for the Computer Science Bachelors of Science, with an emphasis on EAE.

The EAE Track culminates in a Bachelors degree in Computer Science. All requirements of the general CS degree must be met and you should review those requirements in section 1.3 of this document. The specifics of the track are listed below where they extend the requirements of the traditional CS degree:

1. **General Education Requirements**:
   - Fine Arts (FF) Requirement: Art 1020 (Drawing for non majors).
2. **University Graduation Requirements**:
   - Writing requirement: FA3600 (Writing for New Media) replaces upper division writing.
3. **Math, Science, and Engineering Requirements**: Seven classes in math, science and/or engineering are required.
   a. Mathematics 1210 (Calculus I)
   b. Mathematics 1220 or 1250 (Calculus II)
   c. Mathematics 2250 (Ordinary Differential Equations and Linear Algebra)[10]
   d. Physics 2210 (Physics for Scientists and Engineers I)
   e. Physics 2220 (Physics for Scientists and Engineers II)
   f. Three additional Courses, each of which must be at least three semester hours, chosen from among:
      1. Math 3070 (Applied Statistics) **suggested.**
      2. Mechanical Engineering 2020 (Particle Dynamics) **suggested.**
      3. See CS Math, Science, and Engineering requirements.
4. **Computer Science**. The following classes are required:
   a. The following classes are required:
      1. CS 1010 (.5) - Introduction to Unix
      2. CS 1060[11](3) - Explorations in Computer Science
      3. CS 1410 (4) - Introduction to Computer Science I (EAE section)
      4. CS 2420 (4) - Introduction to Computer Science II (EAE section)

10 Alternatively, Math 2270 (Linear Algebra) and Math 2280 (Intro to Differential Equations) can be taken as separate courses

11 CS 1060 may be waived by advanced students. Talk to the Director of Undergraduate Studies for the School of Computing.

5. CS 2100 (3) - Discrete Structures
6. CS 3500 (4) - Software Practice I
7. CS 3505 (3) - Software Practice II
8. CS 3810 (4) - Computer Organization
9. CS 4150 (3) - Algorithms
10. CS 4400 (4) - Computer Systems
11. CS 3650 (3) - 3D Modeling for Video Games and Machinima
12. CS 3660 (3) - Machinima
13. CS 5460 (4) - Operating systems
14. CS 5530 (3) - Databases
15. CS 5300 (3) - Artificial Intelligence

One of the following series must be taken (usually in the Junior Year):

1. Computer Animation Series
   a. FILM 4700 (4) - Computer Animation I
   b. FILM 4710 (4) - Computer Animation II
2. Game Development Series
   a. FILM 4750 (4) - Game Development: Historical and Traditional Genres
   b. FILM 4760 (4) - Game Development: Contemporary and Alternative Genres

b. Theory restricted elective. Students must take the following course:

- CS 3200(3) - Scientific Computing

c. Entertainment Arts and Engineering Requirement. The following class is required:

- Film 3710(4) - Film Production I

d. EAE Computer Science Elective. One of the following classes must be taken:

- CS 5530 (3) - Machine Learning
- CS 5600 (3) - Intro to Graphics
- CS 5610 (3) - Interactive Graphics
- CS 5961 (3)* - Networked Game Design
- CS 4550 (3) - Web Software Architecture
- CS 6620 (3) - Introduction to Ray Tracing

e. Additional Suggested Electives[12] (not required):

- FA 3350 (3) - Introduction to 3D Computer Graphics
- FILM 2300 (3) - Survey of Videogame Theory and Design

[12] While not required, there are many courses that would benefit you in your EAE future. When possible, where general education or math/science/engineering requirements have already been met (e.g., through AP credit), additional EAE electives should be considered. Talk to your EAE Advisor for more guidance.

f. Capstone Requirement. Both of the following courses are required:

- CS 4510 (3) Senior Project I
- CS 4515 (3) Senior Project II

g. Exit Survey: Same (as listed on page 4)

h. Duplication of Credit: Same (as listed on page 4)

5. Continuing Performance Requirement:

- Same (as listed on page 5)

1.4.1 Example EAE Program of Study

	Fall		Spring	
Freshman	CS 1010 (Unix)	(.5)	CS 2420 (Intro to CS II)	(4)
	CS 1060[13] (Explorations in CS)	(3)	Math 1220 (Calc II)	(4)
	CS 1410[14] (Intro to CS)	(4)	Writing 2010 (Intermediate Writing)	(3)
	Math 1210 (Calc I)	(4)	Math/Sci/Eng elective	(3)
	Art 1020 (Basic Drawing)	(3)	Gen Ed*	(3)
		(14.5)		(17)
Sophomore	CS 3500 (Software Practice I)	(4)	CS 3505 (Software Practice II)	(3)
	CS 3810 (Computer Org)	(4)	CS 3660 (Machinima)	(3)
	CS 3650 (3D Modeling)	(3)	Film 3710 (Film Production I)	(3)
	Physics 2210	(4)	Physics 2220	(4)
			FA 3600 (Writing for NM)	(3)
		(15)		(16)
Junior	CS 2100 (Discrete Structures)	(3)	CS 4150 (Algorithms)	(3)
	CS 4400 (Comp Systems)	(4)	CS 3200 (Scientific Computing)	(3)
	Math 2250 (ODEs & LA)	(4)	CS 5300 (Artificial Intelligence)	(3)
	EAE series (Animation/Game I)	(4)	EAE series (Animation/Game II)	(4)
			Gen Ed	(3)
		(15)		(16)
Senior	CS 4510 (Senior Project I)	(3)	CS 4515 (Senior Project II)	(3)
	CS 5530 (Databases)	(3)	EAE CS elective	(3)
	CS 5460 (Operating Systems)	(4)	Math/Sci/Eng elective	(3)
	Gen Ed*	(3)	Gen Ed*	(3)
	Gen Ed*	(3)	Gen Ed*	(3)
		(16)		(15)

* One GE course should be used to meet the American Institutions requirement.

Table 2: Sample Degree Program of Study

This table gives an eight-semester example program leading to Computer Science B.S. with a specialty in Engineering Arts and Entertainment. It is meant only as a guide, since the scheduling of electives and General Education classes depends upon which ones are selected. All requirements for a general degree in Computer Science must be met. It is possible to take certain general education requirements during Summer sessions to ease the load.

13 CS 1060 is required but can be waived for sufficiently advanced students. You must meet with the Director of Undergraduate Studies for the School of Computing.

14 CS 1410 is also offered in the Spring with CS 2420 offered in the Summer. For students new to Computer Science, taking 1060, 1410, and 2420 in consecutive semesters (Fall, Spring, Summer) can ease the transition into the major.

1.4.2 EAE Requirement Summary

PRE-MAJOR REQUIREMENTS:

**Take all 6 courses. You must earn a C- or better in each course, and a 2.8 average GPA among the following courses:*

1. CS 1010, Intro to Unix ________________ (.5)
2. CS 1060, Explorations in CS ________________ (3)
3. CS 1410, Comp. Science. I ________________ (4)
4. CS 2420, Comp. Science. II ________________ (4)
5. Math 1210, Calculus I ________________ (4)
6. Math 1220, Calculus II ________________ (4)

GENERAL EDUCATION REQUIREMENTS:
*Complete each requirement.

1. Wrtg 2010 or ESL 1050+1060 ________________ (3)
2. FA 3600, Writing for New Media ________________ (3)

3. American Institutions (AI) ________________ (3)
*(choose one: Econ 1740, History 1700, Hon 2212, Political Science 1100)

A total of 6 courses must be taken in the following 3 areas: Fine Arts, Humanities, Behavioral/Social Science, take two courses from each area. Two of those six classes must be upper division (3000 level or above), one must satisfy the university Diversity requirement and one must satisfy the university International requirement.

1. ART 1020 Drawing for non-majors (FF): ___________ (3)
2. Fine Arts (FF): ______________________________ (3)

3. Humanities (HF): ______________________________ (3)
4. Humanities (HF): ______________________________ (3)

5. Social/Behavioral Science (BF): ________________ (3)
6. Social/Behavioral Science (BF): ________________ (3)

- Upper Division 1 ☐
- Upper Division 2 ☐
- Diversity (DV) ☐
- International (IR) ☐

MATH / SCIENCE / ENGINEERING ELECTIVES:
*(Physics 2210 and 2220 required. Plus 3 additional electives, each must be at least 3 credits. Choose **from Biol 1210, Chem 1210, Phys 2220**, or any non-CS class that has Math 1220 (Calculus II) as a prerequisite or co-requisite.)*

1. Physics 2210, Physics for Scie & Engrs___ (4)
2. Physics 2220, Physics II ______________ (4)
3. Math 2250 , ODEs & Linear Algebra _____ (4)
4. ___________ , ______________________ (   )
5. ___________ , ______________________ (   )

The Following Requirements are Restricted to FULL Majors:

FILM REQUIREMENT:

FILM 3710 – Film Production ________________ (3,F)

CS MAJOR REQUIREMENTS:
* CS 2100 can be taken be pre-majors

1. CS 2100*, Discrete Structures ________________ (3,F)
2. CS 3500, Software Practice I ________________ (4,F)
3. CS 3505, Software Practice II ________________ (3,S)
4. CS 3810, Comp. Organization ________________ (4,F)
5. CS 4150, Algorithms ________________ (3,S)
6. CS 4400, Comp. Systems ________________ (4,F)

EAE REQUIREMENTS:

1. CS 3960, 3D Modeling for VG&M ________________ (3,F)
2. CS 5460, Operating Systems ________________ (4,F)
3. CS 5300, Artificial Intelligence ________________ (3,S)
4. CS 5530, Databases ________________ (3,F)
5. CS 5964, Machinima ________________ (3,S)

EAE CS ELECTIVE:
Choose 1 course, 3000-level or above from approved EAE CS list.

1. CS _________ , ___________________________ (   )

ANIMATION SERIES:
Must complete the Animation Series or the Game Design Series.

1. FILM 4700, Computer Animation I ________________(4,F)
2. FILM 4710, Computer Animation II ________________(4,S)

GAME DESIGN SERIES:
Must complete the Animation Series or the Game Design Series.

1. FILM 4750, Game Design I ________________(4,F)
2. FILM 4760, Game Design II ________________(4,S)

THEORY ELECTIVE:

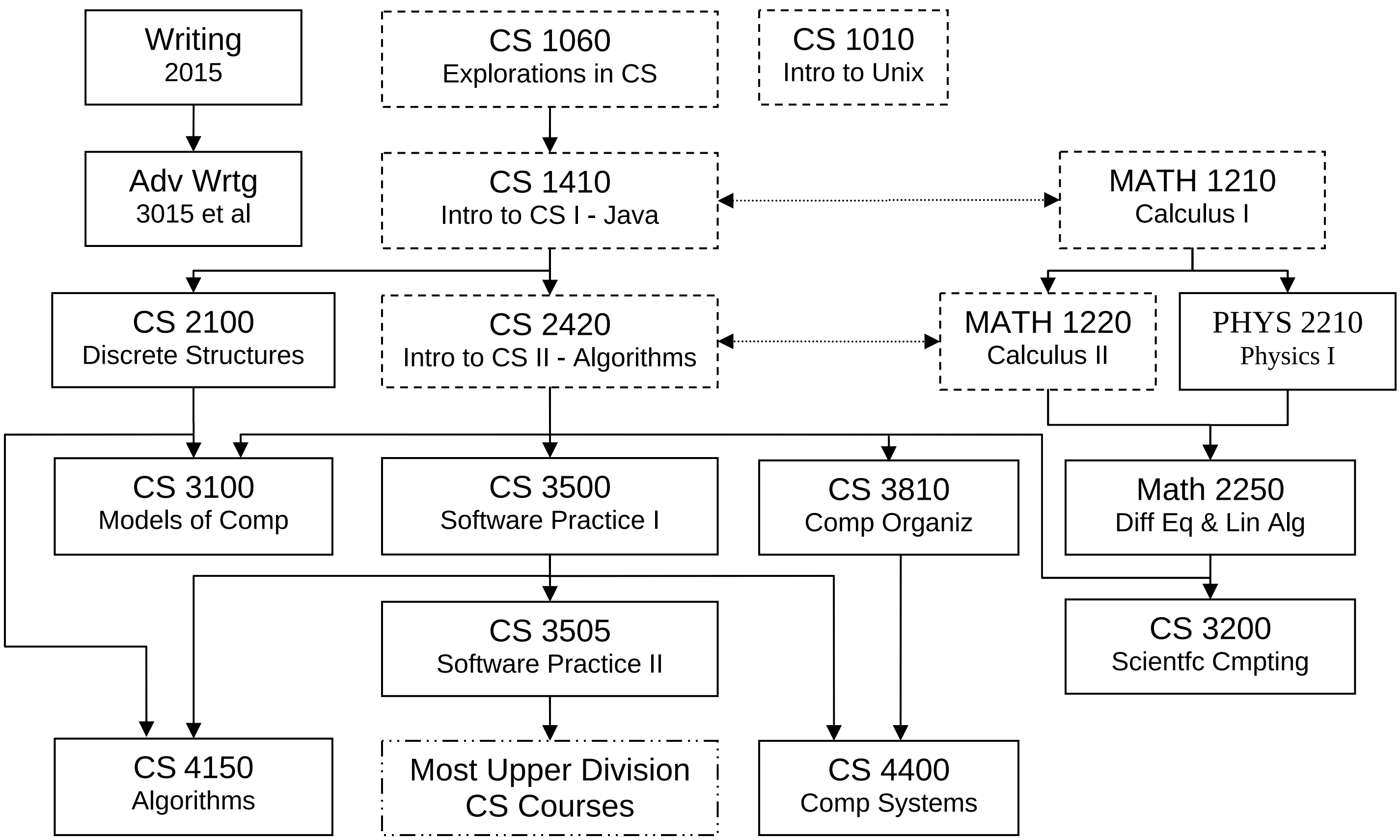
CS 3200, Scientific Computing ________________ (3,S)

CAPSTONE REQUIREMENT:

CS 450x, Senior Project I ________________ (3,F)
CS 450x, Senior Project II ________________ (3,S)

1.5 Computer Science Prerequisite and Course Requirements

The following chart shows the prerequisite path through the Computer Science required courses.



This graph illustrates the order in which classes must be taken to satisfy Prerequisite and corequisite requirements in Computer Science

Students must also satisfy the General Education/University requirements, take four courses that satisfy the math/science electives, complete seven Computer Science electives, and satisfy the Capstone requirement.

- - - - : Required Pre-Major Courses, ◄·········► Corequisites

1.6 *Advanced Placement Exam Course Equivalents for Computer Science*

AP Exam	**U. of U. Class Equivalent**	**CS Required Score**[15]	**Equivalent Grade**[16]
Chemistry	CHEM 1210/1220	5	
		4	
Computer Science A	CS 1410	5	A
Computer Science AB	CS 1410/CS 2420	5	A/A
		4	B+/B-
		3	B-/*
Physics C: Mech	Phys 2210	5	
		4	
Physics C: E & M	Phycs 2220	5	
		4	
Calculus AB	Math 1210	5	A
		4	B-
Calculus BC	Math 1210/1220	5	A/A
		4	B+/B-
		3	B-/*
English	Wrtg 2010	5	
		4	
American History	American Institutions Equiv	3	

* Course not waived and must be taken at the University of Utah

Credit for Advanced Placement tests is given through the University and counts toward the total hours need to graduate. Specific courses in the Computer Science major are waived based on high AP exam scores, independent of these credit hours. The grades listed above are for the purpose of determining the pre-major GPA which is used to determine acceptance into the major. For example, a student who receives a 5 on the CS AB test and a 5 on the Calculus BC test would have CS 1410, CS 2420, Math 1210, and Math 1220 waived, with a pre-major GPA of 3.5 (4.0 + 4.0 + 3.3 + 2.7 divided by 4 classes) and would be allowed directly into the full major as a freshman.

Students should submit a “request for evaluation” to the Admissions Office to have their AP credits officially recorded toward University graduation requirements.

15 This is the score necessary for the School of Computing to waive a requirement. While certain “lower” scores will result in University of Utah credit, they will not satisfy School of Computing requirements.

16 These grades will only be used to determine acceptance into the Computer Science Major. They are not part of your university GPA. The School of Computing will not accept AP scores of 3 in some areas.

1.7 College of Engineering Honors Degree

The College of Engineering offers an Honors Bachelors Degree. For information on this program see the college web page: www.coe.utah.edu/current/UG/Honors_in_Engineering/index_html.

1.8 Computer Science BS/MS Degree

The School of Computing offers a combination Bachelor of Science/Master of Science degree in Computer Science. This program allows students to earn a BS and MS in approximately five academic years. The BS/MS can combine a BS in either Computer Science or Computer Engineering with an MS in either Computer Science or Computing.

Degree requirements are the same as those for earning a BS and MS separately, but there are two basic advantages:

- Admissions decisions are made before senior year, so students can plan with certainty.
- The synchronization barrier between the degrees is broken. Hence, students may take graduate classes during their senior year, and undergrad classes during their fifth year. (Note that International students have special regulations and should consult the BS/MS Program Advisor to help them plan their schedules.)

Students wishing to pursue the thesis option for their MS degree must also choose the thesis option for their BS degree. The Bachelors Thesis will normally constitute a portion of the Masters Thesis.

Applications for the BS/MS program are due on January 15th of the junior year. Program details and application materials are available in the BS/MS Handbook and on the School of Computing web page.

1.9 Computer Science as a Minor

The School of Computing offers a minor for students who desire to gain sufficient background to use and program computers in another field.

In order to be admitted as a computer science minor, a student must have a declared major in another department and be making progress in that major. The admission process is similar to that for majors and is carried out at the same times. Students are admitted to the minor if their average grade in Math 1210, CS 1410, and CS 2420 is 2.8 or higher.

The minor consists of a minimum of 18.5 semester hours of computer science classes.

- Required. The following classes must be taken.
  1. CS 1010 Introduction to Unix
  2. CS 1410 Introduction to Computer Science I (formerly 2010)
  3. CS 2420 Introduction to Computer Science II (formerly 2020)
  4. CS 2100 Discrete Structures
  5. CS 3500 Software Practice
- Elective. Students must take at least one additional CS class at the 3000 level. CS 3505 is recommended.

Computer science minors are guaranteed admission into only the upper division classes that comprise their minor.

1.10 Students Pursuing a Second Degree or a Double Major

Some students may wish to earn a degree in Computer Science as their second B.S. degree. This is possible as long as the requirements for both degrees are met. In some cases, fewer additional class hours are needed because of overlaps in the two degrees. This is especially true of students whose other degree is in computer engineering, electrical engineering, or mathematics, where upper level classes may serve as computer science electives. Students pursuing a double major must notify the Academic Counselor.

1.11 Undergraduate Scholarships

The School of Computing awards several scholarships each year. Recipients are selected based upon academic performance, rather than financial need. Most are awarded to Computer Science and Computer Engineering full majors, or to those students who will become full majors during the following academic year. Applications for these scholarships are available on the web at http://www.cs.utah.edu/students/undergradinfo.shtml or from the School of Computing office. They must be submitted to the School's office by March 15 of the preceding year.

> **Tuition Waiver Scholarships**. These awards are available to students majoring in Computer Science or Computer Engineering who are residents of the state of Utah. They cover up to 18 credit hours of resident tuition for two semesters. To be eligible, students must take at least 12 credit hours per semester.
>
> **School of Computing Scholarships**. These are awards available to all Computer Science and Computer Engineering majors. They range in value from $500 to $3,000, and are made possible by generous donations from the School of Computing faculty, the Eccles Foundation, Kiri Wagstaff, Dave Hanscom and others. To be eligible, students must take at least 9 credit hours per semester.

The College of Engineering also awards several scholarships (Kennecott, Ariel Berrier, Simon Ramo, and others) to the top students in the college. Students may also apply for financial aid from the College, which each year awards a number of Josephine Beam Educational Scholarships. These are worth approximately $500 and are based on need. Information and applications are available on the web at http://www.coe.utah.edu/current/FA Scholarship or from the Office of the Dean of Engineering (WEB 1650). Applications must be submitted by February 15 of the preceding year.

1.12 Employment Opportunities

The School of Computing now offers an internship/co-op program which allows qualified students to work in their fields of interest for all or part of their junior and/or senior years. This can be done on a full or part time basis, either in Salt Lake City or elsewhere.

The benefits of such experience include exposure to ideas which could help with career decisions, making contacts which may be useful sometime in the future, and valuable experience in an area that is pertinent to current studies. Among the corporations participating are IBM, Hewlett Packard, L-3 Communications Systems, Intel, and Micron. Many of our majors take advantage of this valuable opportunity. Students seeking employment should register with the University Office of Career Services (www.careers.utah.edu). Students should contact the Director of Undergraduate Studies for more information on this program.

The School of Computing employs a number of junior and senior students as computer operators and as teaching assistants. These jobs involve no more than 20 hours of work per week at an appropriate hourly wage. Appointments are made each semester based on student applications, which should be submitted prior to the start of each term. These applications are available on the web[17]. In addition, general inquiries are received periodically from local industry and from University research groups for students who are interested in working part or full time. These are emailed to all Computer Science and Computer Engineering majors. More information may be obtained from the Faculty Academic Counselor.

Students seeking employment upon graduation should contact the University Office of Career Services in order to be included on a list supplied to employers. Students not planning to work towards an advanced degree should register with Career Services during their junior year, since most companies begin interviewing during the fall semester.

1.13 Student Participation in School Affairs

Opportunities for students to develop their organizational and leadership abilities are available through participation in the Undergraduate Student Advisory Committee (UgSAC), which plays an active role in the School by coordinating the following:

1. Course and faculty teaching evaluations.
2. Representation (one student) at faculty meetings.
3. Announcements to all declared pre-majors and majors.
4. Representation on the College Student Advisory Committee.
5. Organization of Engineering Week activities.
6. Organization of lunch meetings for pre-majors and majors.
7. Organization of university and high school programming contests.
8. Feedback on issues affecting students, such as scheduling, curriculum changes, and graduation requirements.
9. GRE preparation classes.

Anyone interested in joining this organization should contact UgSAC at ugsac@cs.utah.edu. Participation, suggestions, and criticisms are solicited.

17 www.cs.utah.edu/dept/webforms/current/ugrad-TA.html

2 Computer Science Courses

The number and title of each course is followed by the number of semester hours it carries, the semester(s) during which it is taught (F=fall, S=spring, U=summer), its prerequisites, its co-requisites, and any courses with which it is cross-listed.

Where a course has both a 5000-and 6000-level number, the 5000-level version is intended for undergraduates, and the 6000-level version is for honors and graduate students. The two versions of the class will meet together, but extra work will be expected of honors and graduate students. Additional credit toward the Bachelors Degree will not be given for taking a 6000 level course after taking the 5000 level version.

Courses that have only 6000-level numbers may be taken by graduate and advanced undergraduate students.

Some elective classes are not offered every year. Check the on-line schedule or the Computer Science Advisor to see which classes will be offered in upcoming semesters.

1000 Engineering Computing (3, FS) Coreq: CS 1010, MATH 1210

Introduction to programming principles and engineering problem solving via computational means using MATLAB (during the first half of the semester) and C (during the second half of the semester). Decomposition of programs into data representations, functions, and control structures. Clean programming practices are emphasized. The MATLAB portion of the course focuses on the implementation of physically-based models, data visualization via plotting, and selected numerical techniques. The C portion of the course introduces basic syntax and special features of the language for engineering implementations. Laboratory practice.

1001 Engineering Computing using MATLAB (1.5, FS) Coreq: CS 1010, MATH 1210

Introduction to programming principles and engineering problem solving via computational means using MATLAB. Decomposition of programs into data representations, functions, and control structures. Focus on the implementation of physically-based models, data visualization via plotting, and selected numerical techniques. Clean programming practices are emphasized. (This is a half-semester course that meets with CS 1000.) Laboratory practice.

1010 Introduction to Unix (0.5, FSU)

An introduction to the Unix workstations used in the College of Engineering CADE Lab. Topics include the X Windows system, Unix shell commands, file system issues, text editing with Emacs, accessing the World Wide Web, and electronic mail. Self-paced course using online teaching aids.

1020 Introduction to Programming in C++ (3, U)

An introduction to essential programming concepts using C++. Laboratory practice.

1021 Introduction to Programming in Java (3, FU)

An introduction to essential programming concepts using Java. Laboratory practice emphasizes object-oriented techniques and web-based application design.

1040 Creating Interactive Web Content (3, FS)

Introduction to the essentials of web page design and object-oriented programming through the use of HTML and JavaScript to create interactive web pages. It is appropriate for any student who is comfortable using a computer to write a paper and browse the Web. This is a 100% online course that can be completed on any computer equipped with a recent version of most web browsers.

1050 Computers in Society (3, FS) Social/Behavioral Science Exploration Course

Survey of the social issues that surround the increasingly pervasive roles that computers play in society. Topics include privacy of personal information, encryption and interception of communications, risks posed by unreliable computer systems, freedom of speech in cyberspace, intellectual property as it relates to downloadable media, computer-based crime, and computers in the workplace. Case studies will focus on computer-related issues of current public interest. No background in computer technology is required or assumed beyond the ability to use a computer to send e-mail, browse the web, and write papers.

1060 Explorations in Computer Science (3, FS) Applied Science

Applications of modern computing have transformed the ways that people communicate, govern, learn, play, shop, socialize, and work. This course introduces seven computational principles (representation, abstraction, algorithms, digital circuits, stored programs, programming languages, and networks) and shows how they make possible such disparate technologies as web search engines, computer-animated movies, computer games, digital music, and artificial intelligence. The course assumes no background in computing beyond the ability to use a computer to send e-mail, browse the web, and write papers. The course is appropriate for students seeking an understanding of computational principles that will complement their major field of study.

1410 Introduction to Computer Science I (4, FS) Coreq: MATH 1210, CS 1010

The first course required for students intending to major in computer science and computer engineering. Introduction to the engineering and mathematical skills required to effectively program computers, and to the range of issues confronted by computer scientists. Roles of procedural and data abstraction in decomposing programs into manageable pieces. Introduction to object-oriented programming. Extensive programming exercises that involve the application of elementary software engineering techniques.

2000 Introduction to Programming in C (4, F) Coreq: MATH 1210, CS 1010

Introduction to essential programming concepts using C. Decomposition of programs into functional units; control structures; fundamental data structures of C; recursion; dynamic memory management; low-level programming. Some exposure to C++. Laboratory practice. (Intended for non-CS/CE majors).

2100 Discrete Structures (3, F) Prerequisite: CS 1410

Introduction to propositional logic, predicate logic, formal logical arguments, finite sets, functions, relations, inductive proofs, recurrence relations, graphs, and their applications to Computer Science.

2420 Introduction to Computer Science II (4, SU) Prerequisite: CS 1410

The second course required for students intending to major in computer science and computer engineering. Introduction to the problem of engineering computational efficiency into programs. Classical algorithms (including sorting, searching, and graph traversal) and data structures (including stacks, queues, linked lists, trees, hash tables, and graphs). Analysis of program space and time requirements. Extensive programming exercises that require the application of elementary techniques from software engineering.

3010 Industry Forum (1, F): Credit/No-Credit

Meets with 3011. The Industry Forum is designed to expose students to topics that are not discussed in depth as part of the normal curriculum, but that are likely to be important after they graduate. Each week one or more guest speakers, typically local and national business leaders, will give a talk and answer questions on a topic of interest to them. Topics will run the gamut from the highly career oriented (e.g., how to write a resume and interview or how to decide if graduate school is right for you) to the highly technical (e.g., how video special effects are generated or how software development organizations manage complex system development). 3010 **cannot** be applied toward a CS elective course requirement.

3011 Industry Forum (1, F):

Meets with 3010. 3011 is offered for a letter grade and can be applied towards a CS elective course requirement. This course can be taken up to three times.

3100 Models of Computation (3, F) Quantitatively Intensive. Prerequisite: CS 2420, CS 2100

Models of sequential computation, including finite-state automata, push-down automata, and Turing machines.

3200 Scientific Computation (3, S) Prerequisite: CS 2420, MATH 2250

Scientific computation relevant to computational science and engineering, with emphasis on the process of modeling, simulation, visualization and evaluation. Possible topics related to the four areas include: (modeling) continuous and statistical modeling; (simulation) solving and linear and non-linear systems, interpolation and approximation, numerical differential equations; (visualization) scalar and vector field visualization techniques; (evaluation) connection of results back to case-studies of interest from areas such as physics, biology, etc. Basic knowledge of programming, linear algebra, and differential equations is assumed.

3500 Software Practice I (4, F) Prerequisite: CS 2420

Meets with CS 5010. Practical exposure to the process of creating large software systems, including requirements specifications, design, implementation, testing, and maintenance. Emphasis on software process, software tools (debuggers, profilers, source code repositories, test harnesses), software engineering techniques (time management, code and documentation standards, source code management, object-oriented analysis and design), and team development practice. Much of the work will be in groups and will involve modifying preexisting software systems.

3505 Software Practice II (3, S) Prerequisite: CS 3500

Meets with CS 5020. An in-depth study of traditional software development (using UML) from inception through implementation. The entire class is team-based, and will include a project that uses an agile process.

3700 Fundamentals of Digital System Design (4, S) Quantitatively Intensive B.S. Course. Cross-listed as ECE 3700. Prerequisite: CS 1410 or CS 2000, PHYCS 2220

Techniques for reasoning about, designing, minimizing and implementing digital circuits and systems. Combinatorial (logic and arithmetic) and sequential circuits are covered in detail leading up to the design of complete small digital systems using finite state machine controllers. Use of computer-aided tools for design, minimization, and simulation of circuits. Laboratory is included involving circuit implementation with MSI, LSI, and field programmable gate arrays.

3710 Computer Design Laboratory (3, F) c.l. ECE 3710. Prerequisite: CS/ECE 3700, CS/ECE 3810

Working in teams, students employ the concepts of digital logic design and computer organization to design, implement, and test a computing system. Interface I/O devices and develop associated software/firmware. Extensive use of CAD and software tools.

3810 Computer Organization (4, F) Quantitatively Intensive, c.l ECE 3810. Prerequisite: CS 2420 or CS 2000

An in-depth study of computer architecture and design, including topics such as RISC and CISC instruction set architectures, CPU organizations, pipelining, memory systems, input/output, and parallel machines. Emphasis is placed on performance measures and compilation issues.

3950 Independent Study (1–4)

3960 Special Topics (1–4)

Special topics courses are taught every year. Check the on-line schedule for a current listing of offerings.

3991 Computer Engineering Junior Seminar (0.5, F) Cross-listed as ECE 3991. Prerequisite: CE major status

Presentations from faculty and industry representatives to discuss trends in computer engineering, professionalism, ethics, the impact of engineering in global and societal contexts, lifelong learning, and contemporary issues.

3992 Computer Engineering Pre-Thesis/Pre-Project (1, S) Cross-listed as ECE 3992. Prerequisite: CS/ECE 3710 and 3991, CE major status; Coreq: CS/ECE 5780

Fundamentals of project planning (scoping, group selection, risk assessment, scheduling, backup planning, strategy, etc.) are covered in the first half of the course. The second half involves student presentations and critique of proposals in progress. The final result of the course will be an approved project or thesis proposal.

4005 Honors Research Practice (3, S) Prerequisite: CS 3500 and admission to CS Honors track

Techniques for identifying a Computer Science research problem, literature review, research execution, and preparation for publication.

4010 Teaching Introductory Computer Science (1, FS) Prerequisite: Permission of instructor

Issues confronted by undergraduate teaching assistants in introductory computer science courses, including leading lab sections, conducting office hours, grading assignments, communicating with students. Each student must currently be an undergraduate teaching assistant in the School of Computing. This course does count toward CS elective credit and may be taken up to three times.

4150 Algorithms (3, S) Quantitatively Intensive B.S. Course. Prerequisite: CS 2100, CS 3500

Study of algorithms, data structures, and complexity analysis beyond the introductory treatment from CS 2420. Balanced trees, heaps, hash tables, string matching, graph algorithms, external sorting and searching. Dynamic programming, exhaustive search. Space and time complexity, derivation and solution of recurrence relations, complexity hierarchies, reducibility, NP completeness.

4400 Computer Systems (4, F) Prerequisite: CS 3500, CS 3810

Introduction to computer systems from a programmer's point of view. Machine level representations of programs, optimizing program performance, memory hierarchy, linking, exceptional control flow, measuring program performance, virtual memory, concurrent programming with threads, network programming.

4500 Software Engineering Laboratory (3, S) Prerequisite: CS 3505, senior standing in Computer Science

Development of significant software systems by small student groups, with emphasis on applying sound, disciplined software engineering practice.

4540 Web Software Architecture (3, S) Prerequisite: CS 3505

Software architectures, programming models, and programming environments pertinent to developing web applications. Topics include client-server model, multi-tier software architecture, client-side scripting (JavaScript), server-side programming (Servlets and JavaServer Pages), component reuse (JavaBeans), database connectivity (JDBC), and web servers.

4550 Simulation (3, F) Prerequisite: CS 3505

Basic simulation modeling, modeling complex systems, basic probability and statistics for simulation, building valid simulations, random numbers, and output data analysis. Both discrete event and continuous simulation may be covered.

4710 Computer Engineering Senior Project (3, F) Cross-listed as ECE 4710. Prerequisite: CS/ECE 3710, 3992, and 5780

This is the capstone project course for Computer Engineering majors who do not choose to do a thesis. Projects are done in groups and are of the student's choosing. Classroom sessions are devoted to improving presentation skills and serve as peer reviews of the ideas and work done to date. Multiple in-progress oral presentations are required as is a final written project report and a final oral presentation.

4950 Independent Study (1–4)

4960–4964 Special Topics (1–4)

Special topics courses are taught every year. Check the on-line schedule for a listing of offerings.

4970 Computer Science Bachelors Thesis (3) Prerequisite: Senior standing in computer science

Only students who have previously worked with a faculty member in a research group may register for Bachelors Thesis credit, and then only with the permission of the faculty member. An undergraduate thesis is a publication-quality description of work done in previous semesters. At a minimum a thesis must be published as a technical report; ideally, it should be submitted to a conference or journal. A Bachelors Thesis is intended as an alternative to the senior Software engineering Laboratory for students who are headed for graduate school.

4991 Computer Engineering Senior Thesis I (2, F) Cross-listed as ECE 4991. Prerequisite: CS/ECE 3992 and approved senior thesis proposal

Students work on original senior thesis project under the direction of their approved thesis advisor. This course along with CS/ECE 4992 substitute for CS/ECE 4710 (Computer Engineering Senior Project) for students who have chosen to do a thesis.

4992 Computer Engineering Senior Thesis II (2, S) Cross-listed as ECE 4992. Prerequisite: CS/ECE 4991

Students work on an original senior thesis project under the direction of their approved thesis advisor, make an oral presentation at the annual student technical conference, and prepare and submit their senior thesis for approval. This course along with CS/ECE 4991 substitute for CS/ECE 4710 (Computer Engineering Senior Project) for students who have chosen to do a thesis.

5010 Software Practice I (4, F) Prerequisite: CS 2420 and permission of instructor

Meets with CS 3500 (See the description above)

This course is for graduate students from other than the School of Computing.

5020 Software Practice II (3, S) Prerequisite: CS 5010 and permission of instructor

Meets with CS 3505(See the description above)

This course is for graduate students from other than the School of Computing.

5100 Foundations of Computer Science (3, S) Prerequisite: CS 3100, CS 4150

Meets with CS 6100. A survey of topics in theoretical computer science, focusing on computability and complexity. Turing machines, decidability, relative computability, recursion theorem, non-deterministic TMs, complexity measures, time and space hierarchies, P and NP, NP-completeness, program specification and verification. *Undergraduates only*.

5150 Advanced Algorithms (3, F) Quantitatively Intensive. Prerequisites: CS 2100, CS 2420, CS 4150

Advanced study of algorithms: Complexity theory, NP-hardness, reductions, lower bounds and models. Greedy algorithms (and matroids), dynamic programming, network flows, geometric algorithms. Beyond NP-hardness: approximation algorithms. The power of randomization: tail bounds, randomized analysis, hashing. Linear programming. A taste of quantum computing.

5300 Artificial Intelligence (3, S) Prerequisite: CS 3505

Meets with CS 6300. Introduction to field of artificial intelligence, including heuristic programming, problem-solving, search, theorem proving, question answering, machine learning, pattern recognition, game playing, robotics, computer vision. *Undergraduates only*.

5310 Robotics (3, F) Cross-listed as ME EN 5220. Prerequisite: CS 1000, MATH 2250, PHYCS 2210

Meets with CS 6310. The mechanics of robots, comprising kinematics, dynamics, and trajectories. Planar, spherical, and spatial transformations and displacements. Representing orientation: Euler angles, angle-axis, and quaternions. Velocity and acceleration: the Jacobian and screw theory. Inverse kinematics: solvability and singularities. Trajectory planning: joint interpolation and Cartesian trajectories. Statics of serial chain mechanisms. Inertial parameters, Newton-Euler equations, D'Alembert's principle. Recursive forward and inverse dynamics. Undergraduates only.

5320 Computer Vision (3, F) Prerequisite: CS 3505, MATH 2210, MATH 2270

Meets with CS 6320. Basic pattern-recognition and image-analysis techniques, low-level representation, intrinsic images, "shape from" methods, segmentation, texture and motion analysis, and representation of 2-D and 3-D shape. *Undergraduates only.*

5340 Natural Language Processing (3, F) Prerequisite: CS 3505

Meets with CS 6340. Computational models and methods for understanding written text. Introduction to syntactic analysis, semantic analysis, discourse analysis, knowledge structures, and memory organization. A variety of approaches are covered, including conceptual dependency theory, connectionist methods, and statistical techniques. pplications include story understanding, fact extraction, and information retrieval. *Undergraduates only*.

5350 Machine Learning (3, F) Prerequisite: CS 3505; CS 5300 recommended

Meets with CS 6350. Techniques for developing computer systems that can acquire new knowledge automatically or adapt their behavior over time. Topics include concept learning, decision trees, evaluation functions, clustering methods, explanation-based learning, language learning, cognitive learning architectures, connectionist methods, reinforcement learning, genetic algorithms, hybrid methods, and discovery. *Undergraduates only.*

5460 Operating Systems (4, F) Prerequisite: CS 4400

Meets with CS 6460. Characteristics, objectives, and issues concerning computer operating systems. Hardware/software interactions, process management, memory management, protection, synchronization, resource allocation, file systems, security, and distributed systems. Extensive systems programming.

5470 Compiler Principles and Techniques (4, S) Prerequisite: CS 3100, CS 4400

Lexical analysis, top-down and bottom-up parsing, symbol tables, internal forms and intermediate languages, runtime environments, code generation, code optimization, semantic specifications, error detection and recovery. Use of software tools for lexical analysis and parsing.

5480 Computer Networks (3, F) Prerequisite: CS 4150, CS 4400

Meets with CS 6480. A comprehensive study of the principles and practices of data communication and networks. Topics include: transmission media, data encoding, local and wide area networking architectures, internetwork and transport protocols (e.g., IPv4, IPv6, TCP, UDP, RPC, SMTP), networking infrastructure (e.g., routers, name servers, gateways), network management, distributed applications, network security, and electronic commerce. Principles are

put into practice via a number of programming projects. *Undergraduates only.*

5510 Programming Language Concepts (3, F) Prerequisite: CS 3500

Ideas behind the design and implementation of programming languages. Syntactic description; scope and lifetime of variables; runtime stack organization; parsing and abstract syntax; semantic issues; type systems; programming paradigms; interpreters and compilers.

5520 Anatomy of a Modern Programming Language (3, S) Prerequisite: CS 5510

Requirements, challenges, and techniques for designing a modern programming language, currently focusing on Java as a case study. Syntactic and lexical issues, semantic specification, modularity concepts, support for object-oriented programming, types and subtypes, type safety and security, portability, compilability, dynamic linking and loading, program evolvability, use of meta data (reflection), multi-threading, native code generation and linkage, generic types, persistence.

5530 Database Systems (3, F) Prerequisite: CS 3500

Meets with CS 6530. Representing information about real world enterprises using important data models including the entity-relationship, relational and object-oriented approaches. Database design criteria, including normalization and integrity constraints. Implementation techniques using commercial database management system software.

Selected advanced Topics such as distributed, temporal, active, and multi-media databases. *Undergraduates only.*

5540 Human/Computer Interaction (3, F) Prerequisite: CS 3500

Meets with CS 6540. Fundamentals of input/output devices, user interfaces, and human factors in the context of designing interactive applications. Undergraduates only.

5600 Introduction to Computer Graphics (3, S) Prerequisite: CS 3500, MATH 2250

Basic display techniques, display devices, and graphics systems. Homogeneous coordinates, transformations, and clipping. Introduction to lighting models. Introduction to raster graphics and hidden-surface removal.

5610 Interactive Computer Graphics (3, F) Prerequisite: CS 5600

Meets with CS 6610. Interactive 3D computer graphics, polygonal representations of 3-D objects. Interactive lighting models. Introduction to interactive texture mapping, shadow generation, image-based techniques such as stencils, hidden-line removal, and silhouette edges. Introduction to image-based rendering, global illumination, and volume rendering. Undergraduates only.

5630 Scientific Visualization (3, F) Prerequisite: CS 3505; CS 3200 or CS 6210 or MATH 5600

Meets with CS 6630. Introduction to the techniques and tools needed for the visual display of data. Students will explore many aspects of visualization, using a ”from concepts to results” format. The course begins with an overview of the important issues involved in visualization, continues through an overview of graphics tools relating to visualization, and ends with instruction in the utilization and customization of a variety of scientific visualization software packages. *Undergraduates only.*

5710 Digital VLSI Design (4, F) Cross listed as ECE 5710. Prerequisite CS 3700.
Recommended prerequisite CS 3810.

Meets with CS 6710. Introduction to basic concepts of the design of CMOS integrated circuits. Static and dynamic properties of CMOS circuits, composite layout of CMOS circuits, and modeling of transistors. Commonly encountered CMOS circuits. Students complete design, composite layout, and simulation of an integrated circuit project using computer-aided design tools.

5720 Analog Integrated Circuit Design (3, S) Cross-listed as ECE 5720. Prerequisite: ECE 3110

Meets with CS 6720. Design of analog and mixed-signal CMOS integrated circuits. Fundamental building blocks for analog circuits, including the basic principles of op amp, current mirror, and comparator design. The basics of sample-and-hold circuits. Students complete integrated circuit design, simulation, layout, and verification using computer-aided design tools. *Undergraduates only.*

5740 Computer-Aided Design of Digital Circuits (3, S) Cross-listed as ECE 5740. Prerequisite: CS/ECE 3700, CS 4150

Meets with CS 6740. Introduction to theory and algorithms used for computer-aided synthesis of digital integrated circuits. Topics include algorithms and representations for Boolean optimization, hardware modeling, combination logic optimization, sequential logic optimization and technology mapping. *Undergraduates only.*

5745 Testing and Verification of Digital Circuits (3) Cross listed as CS 5745. Prerequisite: ECE/CS 3700.

Study of failure and fault models in digital circuits, stuck-at-faults, transition faults, transistor faults, combinational/sequential circuit ATPG, FSM testing, design fault test, LFSR and BIST, equivalence checking, BDDs, BMDs, canonical representations of Boolean functions.

5750 Synthesis and Verification of Asynchronous VLSI Systems (3) Cross-listed as ECE 5750. Prerequisite: CS/ECE 3700, CS 3505

Meets with CS 6750. Introduction to systematic methods for the design of asynchronous VLSI systems from high-level specifications to efficient, reliable circuit implantations. Topics include specification, protocols, graphical representations, synthesis, optimization using timing information, and verification. *Undergraduates only*.

5780 Embedded System Design (4, S) Cross-listed as ECE 5780. Prerequisite: CS/ECE 3810, CS 2000 or 4400

Meets with CS 6780. Introduction to issues in embedded system design using microcontrollers. Topics include: microcontroller architecture, memory interfacing, serial and parallel I/O interfacing, analog interfacing, interrupt synchronization, and embedded software. *Undergraduates only.*

5785 Advanced Embedded Systems (4, F) Cross-listed as ECE 5785. Prerequisite: CS/ECE 5780

Meets with CS 6785. This class is about building reliable and efficient embedded systems, with a bias toward software issues and a bias toward whole-system issues. Students complete several projects in C running on ARM-based embedded development boards. The course covers a number of special topics such as embedded software architectures, digital signal processing, feedback control, real-time scheduling, verification and validation, wired and wireless embedded networks, and safety-critical embedded systems. *Undergraduates only.*

5830 VLSI Architecture (3, S) Cross-listed as ECE 5830. Prerequisite: CS/ECE 3700, CS/ECE 3810

Meets with CS 6830. Project-based study of a variety of Topics related to VLSI systems. Use of field programmable gate arrays to design, implement, and test a VLSI project. Undergraduates only.

5950 Independent Study (1–4)

5960–5969 Special Topics (1–4)

Special topics courses are taught every year. Check the on-line schedule for a listing of offerings.

6020 Conducting, Publishing, and Presenting Early-Career Research (3) Prerequisite: Graduate standing in Computer Science

This is an independent study offering designed to encourage beginning graduate students to conduct, publish, and present original research early in their graduate careers. A graduate student can earn credit for CS 6020 by having a first-authored paper accepted for publication in a top-tier journal or conference and by subsequently presenting the published work in a one-hour research colloquium. The research must be conducted while a graduate student at Utah; the paper must be accepted within two years of enrolling in the graduate program; the journal or conference must be approved by the student's graduate committee; the colloquium must be presented as soon as possible after the acceptance of the paper; and the student must complete these requirements and register for CS 6020 within three years of enrolling in the graduate program. CS 6020 may not be repeated for credit.

6100 Foundations of Computer Science (3, S) Prerequisite: CS 3100, CS 4150

Meets with CS 5100. Graduate and honors students only. Extra work required.

6110 Formal Methods for System Design (3, S) Prerequisite: CS 5100/6100

Study of methods for formally specifying and verifying computing systems. Specific techniques include explicit state enumeration, implicit state enumeration, automated decision procedures for first-order logic, and automated theorem proving. Examples selected from the areas of superscalar CPU design, parallel processor memory models, and synchronization and coordination protocols.

6210 Advanced Scientific Computing I (3, F) Prerequisite: CS 3200, CS 3505, MATH 3150

An introduction to existing classical and modern numerical methods and their algorithmic development and efficient implementation. Topics include: numerical linear algebra, interpolation, approximation methods and parallel computation methods for nonlinear equations, ordinary differential equations, and partial differential equations.

6220 Advanced Scientific Computing II (3, S) Prerequisite: CS 6210 or MATH 5600

A study of the numerical solution of two and three dimensional partial differential equations that arise in science and engineering problems. Topics include: finite difference methods, finite element methods, boundary element methods, multigrid methods, mesh generation, storage optimization methods, and adaptive methods.

6230 High Performance Parallel Computing (3, S) Prerequisite: Programming in C/C++

Overview of parallel computing; processors, communications topologies and languages. Use of workstation networks as parallel computers. Design of parallel programs: data composition, load balancing, communications and synchronization. Distributed memory and shared memory programming modules; MPI, PVM, threads. Performance models and practical performance analysis. Case studies of parallel applications.

6300 Artificial Intelligence (3, S) Prerequisite: CS 3505

Meets with CS 5300. Graduate and honors students only. Extra work required.

6310 Robotics (3, F) Cross-listed as ME EN 6220. Prerequisite: CS 1000, MATH 2250, PHYCS 2210

Meets with CS 5310. Graduate and honors students only. Extra work required.

6320 Computer Vision (3, S) Prerequisite: CS 3505, MATH 2210, MATH 2270

Meets with CS 5320. Graduate and honors students only. Extra work required.

6340 Natural Language Processing (3, F) Prerequisite: CS 3505

Meets with CS 5340. Graduate and honors students only. Extra work required.

6350 Machine Learning (3, F) Prerequisite: CS 3505; CS 5300/6300 recommended

Meets with CS 5350. Graduate and honors students only. Extra work required.

6360 Virtual Reality (3, S) Prerequisite: CS 5310/6310

Human interfaces: visual, auditory, haptic, and locomotory displays; position tracking and mapping. Computer hardware and software for the generation of virtual environments. Networking and communications. Telerobotics: remote manipulators and vehicles, low-level control, supervisory control, and real-time architectures. Applications: manufacturing, medicine, hazardous environments, and training.

6370 Geometric Computation for Motion Planning (3, F) Prerequisite: CS 1020, MATH 2250

Geometric computation is the study of practical algorithms for solving queries about geometric properties of computer models and relationships between computer models. Robot motion planning uses these algorithms to formulate safe motion through a modeled environment. In addition, algorithms for geometric computation are used in computer animation, simulation, computer-aided design, haptics, and virtual reality. Topics to be covered in this course are spatial subdivision and model hierarchies, model intersection, distance queries and distance fields, medial axis computations, configuration space, and motion planning. The course will rely on lectures, readings, and projects to provide understanding of current practices in the field.

6380 Multiagent Systems (3, S) Prerequisite: knowledge of programming, data structures, processes, language syntax, and either Matlab or C

Covers fundamental notions of (1) software agents, including: autonomy, communication, persistence, and intelligence; and (2) multi-agent systems, including: communication standards, cooperation, competition and coordination.

Methods will be applied to a practical application (usually in Matlab or C).

6460 Operating Systems (4, F) Prerequisite: CS 4400

Meets with CS 5460. Graduate and honors students only. Extra work required.

6470 Advanced Topics in Compilation (3, F) Prerequisite: CS 5470

Compilation of modern languages. Optimization techniques, register allocation and instruction scheduling, garbage collection, exception handling. Linkers and late-stage compilation and optimization.

6480 Computer Networks (3, F) Prerequisite: CS 4150, CS 4400

Meets with CS 5480. Graduate and honors students only. Extra work required.

6490 Network Security (3, S) Prerequisite: CS 5480/6480

Comprehensive introduction to the principles and practices of network security, especially Internet Security. Topics to be covered include: cryptography, authentication, access control, web security, denial-of-service, digital pests, anonymity, and intrusion detection. Existing network security standards will be used for case studies. Includes laboratory practice.

6510 Functional Programming (3, F) Prerequisite: CS 3100, CS 5510

Practical programming with functional language (e.g., Scheme, ML, Haskell) and functional techniques (e.g., fold operators, continuation-passing style, monads, parametric polymorphism). No previous experience with functional language is required. Course work includes writing programs, presenting programs in class, and critiquing peer programs.

6530 Database Systems (3, F) Prerequisite: CS 3500

Meets with CS 5530. Graduate and honors students only. Extra work required.

6540 Human/Computer Interaction (3, F) Prerequisite: CS 3500

Meets with CS 5540. Graduate and honors students only. Extra work required.

6610 Advanced Computer Graphics I (3, F) Prerequisite: CS 5600

Meets with CS 5610. Graduate and honors students only. Extra work required.

6620 Ray Tracing (3, S) Prerequisite: CS 5610/6610

Introduction to ray-tracing. Intersection methods for 3-D objects, reflection and refraction. Introduction to surface and solid texturing. Introduction to continuous-tone pictures and the aliasing problem. Special effects such as soft shadows, depth-of-field, motion-blur, and indirect lighting.

6630 Scientific Visualization (3, F) Prerequisite: CS 3505; CS 3200 or CS 6210 or MATH 5600

Meets with CS 5630. Graduate and honors students only. Extra work required.

6640 Image Processing (3, S) Prerequisite: CS 2420, MATH 2250

An introductory course in processing grey-scale and color images that covers both mathematical fundamentals and implementation. It introduces students to the basic principles of processing digital signals and how those principles apply to images. These fundamentals include sampling theory, transforms, and filtering. The course also covers a series of basic image-processing problems including enhancement, reconstruction, segmentation, feature detection, and compression. Assignments include several projects with software implementations and analysis of real data.

6670 Computer-Aided Geometric Design I (3, F) Prerequisite: MATH 2210, MATH 2250, CS 3505; Coreq: CS 5600

Introduction to current concepts and issues in CAGD systems with emphasis on free-form surface design; mathematics of free-form curve and surface representations, including Coons patches, Bezier method, B-splines, triangular interpolants, and their geometric consequences; classical surface geometry; local and global design tradeoffs and explicit and parametric tradeoffs; subdivision and refinement as techniques in modeling; current production capabilities

compared to advanced research. Laboratory experiments with current CAD systems.

6680 Computer-Aided Geometric Design II (3) Prerequisite: CS 6670

Project based on material covered in CS 6670.

6710 Digital VLSI Design (4, F) Cross listed as ECE 6710. Prerequisite: CS 3700. Recommended prerequisite CS 3810

Meets with CS/ECE 5710. Graduate students only. Extra work required.

6712 Digital IC Projects Testing (1, F) Cross-listed as ECE 6712. Prerequisite: CS/ECE 6710

This course is designed for students who fabricated an integrated circuit in CS/ECE 6710 or 6770. Students will test their chiips independently and report on the experimental results.

6720 Advanced Integrated Circuit Design II (3, S) Cross-listed as ECE 6720. Prerequisite: ECE 3110 Meets with CS 5720. Graduate and honors students only. Extra work required.

6721 Analog Integrated Circuits Lab (1, S) Cross-listed as ECE 6721. Coreq: CS/ECE 6720

Optional lab that accompanies CS/ECE 5720/6720. Students will test and characterize transistors, circuits, and systems on modern CMOS chips.

6722 Analog Integrated Circuits Project Testing (1, F) Cross-listed as ECE 6722. Prerequisite: CS/ECE 6720

This course is designed for students who fabricated an integrated circuit in CS/ECE 5720/6720. Students will test their chips independently and report on the experimental results.

6740 Computer-Aided Design of Digital Circuits (3, S) Cross-listed as ECE 6740. Prerequisite: CS/ECE 3700, CS 4150

Meets with CS 5740. Graduate and honors students only. Extra work required.

6750 Synthesis and Verification of Asynchronous VLSI Systems (3, F) Cross-listed as ECE 6750. Prerequisite: CS/ECE 3700, CS 3505

Meets with CS 5750. Graduate and honors students only. Extra work required.

6760 Modeling and Analysis of Biological Networks (3, F) Prerequisite: Background in molecular or cell biology or formal modeling.

Introduction to methods for modeling and analyzing biological networks such as genetic regulatory networks, metabolic networks, and signal transduction networks. A particular emphasis will be given to methods inspired by models used by engineers for circuit analysis. Other topics include: stochastic analysis using Monte Carlo methods, differential equation models, Bayesian network models, flux balance analysis, learning methods, pathway databases, and synthesized gene circuits.

6770 Advanced Digital VLSI Systems Design (3) Cross-listed as ECE 6770. Prerequisite: CS/ECE 6710 and instructor permission

This course addresses advanced issues in VLSI design, covering the following topics: design methodologies and IP design, CMOS circuit scaling, advanced logic circuit styles, noise sources and signal integrity in digital design, design techniques for dynamic and static power reduction, power supply issues, interconnect analysis, clocking and synchronization, process variation, and performance verification. Students are expected to complete a substantial design project as part of the course, which involves extensive use of CAD tools.

6780 Embedded System Design (4, S) Cross-listed as ECE 6780. Prerequisite: CS/ECE 3810, CS 2000 or 4400

Meets with CS 5780. Graduate students only. Extra work required.

6785 Advanced Embedded Systems (4, F) Cross-listed as ECE 6785. Prerequisite: CS/ECE 5780/6780

Meets with CS 5785. Graduate students only. Extra work required.

6810 Computer Architecture (3, F) Cross-listed as ECE 6810. Prerequisite: CS/ECE 3810

Principles of modern high performance computer and micro architecture: static vs. dynamic issues, pipelining, control and data hazards, branch prediction and correlation, cache structure and policies, cost-performance and physical complexity analyses.

6830 VLSI Architecture (3, S) Cross-listed as ECE 6830. Prerequisite: CS/ECE 3700, CS/ECE 3810

Meets with CS 5830. Graduate and honors students only. Extra work required.

6960–6969 Special Topics (1–4)

Special topics courses are taught every year. Check the on-line schedule for a listing of offerings.

7010 Writing Research Proposals (2, S) Prerequisite: Graduate standing in Computer Science

Fundamental aspects of writing computer science research proposals, including thesis, dissertation, and grant proposals. Form, style, substance, and marketing of effective proposals will be considered. Emphasis is placed on developing and presenting clear and compelling ideas. Substantial writing and class presentations is required of all participants. (This is a half-semester course.)

7120 Information-Based Complexity (3) Prerequisite: CS 3200, MATH 2270, MATH 3210

Analysis of optimal computational methods for continuous problems. Introduction to the general worst case theory of optimal algorithms, linear problems, and spline algorithms as well as selected nonlinear problems. Examples include optimal integration, approximation, nonlinear zero finding, and fixed points.

7240 Sinc Methods (3, S) Prerequisite: CS 6210 or MATH 5600 or MATH 5610

Sinc methods for solving difficult computational problems, such as partial differential and integral equation problems, that arise in science and engineering research. Emphasis on parallel computation. Applications vary, depending on participants in the class. Students are given projects—whenever possible in their areas of research—that lead to publishable research articles.

7250 Advanced Topics in Scientific Computing (3, F) Prerequisite: CS 6220

In-depth study of research topics of current interest in scientific computing. Topics will typically have been surveyed

in CS 6210 and 6220. This course can be repeated for credit since the focus will be changed from semester to semester.

7310 Advanced Manipulation and Locomotion (3) Cross-listed as ME EN 7230. Prerequisite: CS 6310 or ME 6220

This course will examine grasping, rolling, and sliding manipulation from two perspectives; (1) manipulating the pose of an object with an end-effector via grasping, rolling, and sliding manipulation, and, (2) manipulating the trajectory of a mobile robot via the rolling and sliding contact of wheels, feet, or curved exoskeletons and the ground..

7320 System Identification for Robotics (3) Prerequisite: CS 5310/6310 or ME EN 5220/6220

Modeling and identification of the mechanical properties of robots and their environments. Review of probability and statistics. Parametric versus nonparametric estimation. Linear least squares parameter estimation, total least squares, and Kalman filters. Nonlinear estimation and extended Kalman filters. State estimation. Specific identification methods for kinematic calibration, inertial parameter estimation, and joint friction modeling.

7460 Distributed Operating Systems (3) Prerequisite: CS 5460, CS 5480/6480

Practical distributed operating systems concepts from basics through the state of the art. Topics include inter-process communication, client-server systems, distributed shared memory, distributed file systems, distributed databases, portable computing, software fault tolerance, and wide-area (e.g. web) applications. Work includes individual oral presentations, a group project, and a written research report.

7520 Programming Language Semantics (3, S) Prerequisite: CS 3100, CS 5510

Examination of the formal and pragmatic ideas behind programming language design. Imperative, functional, logic, object-oriented, and multi-paradigm languages. Lambda calculus, fixpoints, type systems, and predicate logic. De-notational semantics and models of concurrency.

7650 Realistic Image Synthesis (3, F) Prerequisite: CS 6620, CS 6670, MATH 5010

Using camera and sensor simulation along with physical simulation to generate realistic synthetic images.

7810 Advanced Computer Architecture (3, S) Cross-listed as ECE 7810. Prerequisite: CS/ECE 6810

Issues in the design of modern microprocessors, with emphasis on current research topics in the field. *Offered in alternate years.*

7820 Parallel Computer Architecture (3, S) Cross-listed as ECE 7820. Prerequisite: CS/ECE 6810

Architecture, design, and analysis of parallel computer systems: vector processing, data vs. control concurrency, shared memory, message passing, communication fabrics, case studies of current high performance parallel systems. *Offered in alternate years*.

7960–7969 Special Topics (1–4)

Special topics courses are taught every year. Check the on-line schedule for a listing of offerings.