Computer Science

Bradshaw, Collins, Wahl.

**Major:** Computer Science courses – 110, 223, 225, 361, 461 or 471 (culminating experience); one from each of the following categories: interdisciplinary: 125, 150, 210, 250; development: 324, 329, 336; systems: 234, 340, 345; concepts: 335, 348, Mat 343.
Two additional computer science credits at or above the 300 level.
Cognate course: Mat 143 (or both Mat 220 and Mat 343)
Comprehensive evaluation, with passing grade. Total of 10.5 major courses, plus 1 cognate = 11.5.

**Minor:** Computer Science courses – 110, 223; three additional computer science units, only one of which can be from the following: CS 125, CS 210, CS 250, Mat 143, Mat 220. Total of 5 minor courses.

By-pass credit opportunity: Students placed directly into CS 223, and who earn a grade of B or better for the course, will receive credit for CS 110 upon application to the Registrar.

**CS 110. Fundamentals of Computer Science.** An introduction to problem-solving, program design, testing methodology and programming languages. Students will learn to program in a high level programming language to solve problems from a wide range of topics such as graphics, modeling, information processing and robotics. This class makes use of laboratory experiences and is suitable for students that have never taken a computer science class as well as students with limited programming experience. Satisfies the Abstraction and Formal Reasoning LADR.

**CS 125. Principles of Computer Animation.** Students will learn the principles of animation developed in the 1930’s by traditional animators. These principles involve a study of motion and the mind’s ability to perceive motion. Principles include the notion of timing, squish and squash, arcs of motion, follow through and secondary action. Students will apply the principles of animation with the aid of 3D animation software. At the end of this course each student will develop a short animation clip. Offered alternate years.

**CS 150. Interdisciplinary Topics I.** An interdisciplinary course that bridges the disciplines of computer science and at least one other discipline. May be repeated with permission of instructor.

**CS 160. Special Topics.**

**CS 210. Scientific Modeling.** This course will use real science problems from physical and biological sciences to teach students how to build computational models to explain these problems. Focus is on the use of modeling as part of the scientific process to lead to further discovery. Offered alternate years. Prerequisites: Mat 112 or Mat 121 plus completion of one Natural World LADR course.

**CS 223. Data Structures.** A study of the implementation and evaluation of advanced data structures. Data structures include linked lists, stacks, queues, trees, graphs and tables with the use of recursive algorithms to perform operations. Prerequisites: 110.

**CS 225. Algorithmic Analysis.** A systematic study of creating and evaluating algorithms, the process of solving problems. Brute force, divide-and-conquer, and problem transformation techniques are used to solve problems. Big Theta notation, best-average-worse case analysis,
and notions of NP completeness are used to evaluate problems. Recommended by end of junior year. Prerequisites: 110 and one of Mat 143 or Mat 220.

**CS 234. Computer Organization.** Fundamentals of computer organization, virtual machines, gate logic, binary arithmetic, elementary architecture, microprocessors, memory organization, machine language, assemblers and I/O implementation. Prerequisite: 110.

**CS 250. Interdisciplinary Topics II.** An interdisciplinary course that bridges the disciplines of computer science and at least one other discipline. May be repeated with permission of instructor.

**CS 260. Special Topics.**

**CS 307. Directed Study.** .50 unit.

**CS 324. GUI & Event-Driven Programming.** Focuses on the development of graphical user interfaces (GUIs) and the programming paradigms entailed by such software. Students will build their skills in problem solving and software development by encountering problems relevant to GUIs. Event handling and GUI APIs are introduced. Offered alternate years. Prerequisite: 223

**CS 329. Database with Web Applications.** Introduction to fundamental concepts of database management systems, including database design, data definition languages, data manipulation languages, and database system implementation. Particular emphasis will be placed on implementation of web-based applications that make use of databases. Offered alternate years. Prerequisite: 223.

**CS 335. Theory of Computation.** A mathematical investigation of the fundamental capabilities and limitations of computers. Finite automata, Turing machines, regular and context-free languages, computability and solvability, the halting problem, Church’s thesis. Offered alternate years. Prerequisite: 225.

**CS 336. Programming Languages.** A comparison of design features for selected procedural, functional, logical, or string oriented languages, syntax and semantics, assignment, control structures, data types, procedures and parameter passing, nesting and scoping. Offered alternate years. Prerequisite: 223.

**CS 340. Computer Networks.** A study of modern communication systems, predominantly the Internet. Topics include: the client-server paradigm, use of existing protocols, creation of new protocols, P2P applications, routing algorithms, distributed sharing of resources and communicating in the face of unreliable networks. Offered alternate years. Prerequisite: 223.

**CS 345. Operating Systems.** Theory of operating systems, including memory management, concurrent processes, scheduling algorithms, device drivers, and file systems. Offered alternate years. Prerequisite: 223 or 234 (preferably both).

**CS 348. Artificial Intelligence.** Knowledge representation, natural language processing, models of reasoning and learning, heuristic search methods, connectionism, and robotics. Offered alternate years. Prerequisite: 223 and 225.

**CS 357. Internship.** Off-campus supervised experience in computer science.

**CS 360. Special Topics.**

**CS 361. Junior Seminar.** An introduction to the advanced study of computer science. Students will learn to access and understand current research materials. Classes will be led by student discussion and presentation. Each student will create a proposal for their culminating experience. Prerequisite: any CS course at 300 and above. .50 unit.

**CS 370. Directed Study.**

**CS 461. Senior Seminar.** A seminar to aid students in the completion of their culminating experience. Students will complete the project they proposed in CS 361 and present their findings in both written and oral form. Prerequisite: 361.
CS 465. Capstone Seminar. Course content will reflect the topic for the annual Capstone. Open to all juniors and seniors and may be repeated once for credit. Students may enroll in only one Capstone seminar in a given term.

CS 471. Independent Study.