

Description of Courses

CSCI 121. Introduction to Programming (1/2).

Introduction to computer programming for students with no or limited experience in programming, emphasizing programming constructs (loops, conditionals, recursion) and object-oriented programming. Students work in pairs on increasingly complex programs to create software solutions to problems of general interest. (2A) *Offered each semester, first module.*

CSCI 123. Great Ideas in Computer Science

(1/2). Introduction to computer science in the liberal arts tradition. Topics include social and ethical implications of computer science, privacy, historical perspective, Moore's law, how information is stored, robots, artificial intelligence, and the Internet. *Offered occasionally.*

CSCI 125. Introduction to Computer

Hardware (1/2). Introduction to the organization of the physical components of a computer (hardware) and the interface between the hardware and the programs/instructions (software) that results in a functioning computational machine. Features an introduction to binary numbers, digital logic, and elementary programming at the raw binary level and the assembler level (in terms of the instruction set for the central processing unit (CPU)). Includes study of the other critical components of the computer: memory and its hierarchy, buses, and their communication with one another and the outside world. *Offered fall 2012 and each fall thereafter, first module. Prerequisite: some prior computer programming experience is desirable.*

CSCI 131. Object-Oriented Java Programming

(1/2). Programming in Java for students with substantial programming experience in some language. Basic programming constructs, with emphasis on standard techniques for specifying and documenting programs, plus object-oriented programming, including the use of standard Java objects and the creation of new objects. Testing techniques are emphasized throughout. (1S) *Offered each semester, second module. Prerequisite: Computer Science 121 or experience with programming.*

CSCI 161. Data Visualization (1). Framed in an interdisciplinary perspective, this introductory course focuses on the aesthetic and computational principles and techniques (including programming techniques) for visualizing data and information from different contexts: molecular, statistical, image, graphics, and network/connection-based. Programming and implementation are based on scripting languages, program modules (many based on easy-to-implement matrix operations), and complete software packages. Students also use specialized projection equipment to view images in 3D. (Also listed as Interdisciplinary Studies 161.) (1S) *Offered odd years, spring semester. Prerequisite: Computer Science 121 and Mathematics 110 or consent of instructor.*

CSCI 170. General Topics in Computer

Science (1/4 - 1). Special topics applicable to a general audience. Course title and content vary, and the course may be repeated for credit when the title and content change. *Offered occasionally. Prerequisite: Varies with topic.*

CSCI 204. Data Structures and Algorithms (1).

Practical coverage of data structures with opportunities for software problem-solving. Covers linked-lists, skip lists, general and balanced trees, hashing, and graphs, together with algorithms and standard tools for their implementation, plus algorithms for diverse sorting methods and complexity analysis of algorithms. Students learn how to use abstractions of data structures in designing software for applied problems, to implement the details of algorithms in writing programs, and to analyze the tradeoffs in choices of data structures and algorithms. *Offered each spring. Prerequisite: Computer Science 131.*

CSCI 211. Threaded Programming (½). Basic techniques for threaded programs (“threads”), processes in a computer system or program that function together, such as remote users accessing a common site, a program running in one thread while buttons and menus wait for possible commands (“events”) in other threads, or a graphical animation program with several screen objects moving simultaneously. This course introduces basic techniques for creating threaded programs, communicating between threads, and handling concurrency problems. Topics are discussed in a general context, using event-driven interfaces and graphical animation as concrete programming examples. *Offered odd years, fall semester, first module. Prerequisite: Computer Science 204.*

CSCI 221. Programming Language Paradigms (½). Explores language families outside the traditional imperative and object-oriented languages, including functional languages and logic programming. Offers perspectives on choice of language to suit a problem context. *Offered fall 2012, first module. Prerequisite: Computer Science 204.*

CSCI 227. Databases and the Web (½). Relational database systems are studied with an eye toward practical application. Topics include the relational model, entity-relationship (ER) diagrams, and basic SQL. Also covered are effective web interfaces to databases including scripting access (e.g., PHP or Ruby on Rails). Students also create a full database server. An important aspect of this course is a team project to design and implement a solution to a database application chosen based on student interests. The creation of this project runs the entire semester and is done jointly with students in CSCI 327, but most formal class sessions are in the first half of the semester. This course offers students practical experience with web programming; it can be useful to students in disciplines outside computer science who wish to gain a practical understanding of databases. *Offered even years, fall semester. Prerequisite: Computer Science 204.*

CSCI 231. Computer Networks (½). Introduction to the concepts, design, and implementation of computer networks, presenting both a service model and a layered-architecture model. The course examines the Internet and its services and protocols at the application, transport, network, and physical layers. It also discusses models of network-based computing, with an emphasis on client/server socket-based models. *Offered even years, spring semester, second module. Prerequisite: Computer Science 121 and 125.*

CSCI 241. Computer Architecture (½). In-depth study of the hardware constituting a modern von Neumann central processing unit (CPU) in terms of its functional subunits (registers, arithmetic and logic unit, data path and control) and their interconnections, as well as in terms of its interface to memory and the external world. Includes formal study of digital logic, instruction set architectures, advanced assembly language, and simulation and study of a CPU formed from subunits constructed using digital logic. The course also explores alternative processor architectures and multiprocessing. *Offered odd years, fall semester, second module. Prerequisite: Computer Science 125.*

CSCI 261. Principles of Computation and Modeling (1). Surveys principles of computational science—basic techniques for the application of computer science in different disciplines—in an interdisciplinary fashion, using examples from art, biology, chemistry, economics, and other disciplines. Discusses fundamental algorithms and packaged implementations of numerical methods needed for modeling and visualization: matrix operations, numerical solutions of differential equations, and graphical and image processing. The course explores computational modeling of fractals, chaos, and complex adaptive systems, based on Wolfram’s thesis that a small set of rules can produce complex behavior. Additionally, the course features global simulations (systems dynamics) and local simulations (cellular automata). (Also listed as Interdisciplinary Studies 261.) *Offered even years, spring semester. Prerequisite: Computer Science 204 and Mathematics 110 or consent of instructor.*

CSCI 270. Intermediate Topics in Computer Science (1/4 - 1). Selected aspects of computer science reflecting particular interests and experience of the instructor. Course title and content vary, and the course may be repeated for credit when the title and content change. *Offered occasionally. Prerequisite: varies with topic.*

CSCI 301. Topics in Algorithms (1/2). Investigation of general techniques for the design, comparison, and analysis of different major classes of algorithms. The precise topic varies but possible offerings include: geometric algorithms; parallel and distributed algorithms; net-centric algorithms; analysis of algorithms; graph algorithms; and genetic algorithms. May be repeated for credit if topic is different. *Offered 2012, spring semester, first module. Prerequisite: Computer Science 204; Mathematics 160 or 200.*

CSCI 311. Operating Systems (1/2). Surveys the principles on which modern operating systems are based, including concurrency mechanisms, scheduling, memory management, file systems, and security, with examples from major contemporary operating systems. *Offered in odd years, fall semester, second module. Prerequisite: Computer Science 125 and 211.*

CSCI 327. Databases (1). Relational database systems are studied with a balance of practical application with core database algorithms and implementation details. The first part of the course introduces the practical use of databases including the relational model, entity-relationship (ER) diagrams, and basic SQL. Also covered are effective web interfaces to databases including scripting access (e.g., PHP or Ruby on Rails). Students also create a full database server. An important aspect of this course is a team project to design and implement a solution to a database application chosen based on student interests. The creation of this project runs the entire semester and is done jointly with students in CSCI 227. The second part of the course introduces the topics of normalization, B+-trees, hashing and external sorting, query optimization, transactions and file storage. Students should note that the algorithmic areas covered offer a nice study of advanced algorithms as applied to slow storage devices, and this course is a good complement to Computer Science 204. *Offered even years, fall semester. Prerequisite: Computer Science 204.*

CSCI 331. Graphical Algorithms (1/2). Geometric algorithms and modeling, animation, 3D graphics and rendering. Combines these techniques to consider ways to implement virtual realities, including discussion of future directions of virtual reality. *Offered even years, spring semester, first module. Not offered spring 2012. Prerequisite: Computer Science 211; Mathematics 160 or 200.*

CSCI 341. Object-Oriented Analysis and Design (1/2). Object-oriented design of large programs, including the graphical user interface to a program and emphasizing principles of human-computer interaction. Treats extraction of objects, design of their interactions, and markup language techniques for specifying object designs. The course concludes with validation techniques and the specification of the behaviors of objects, stopping short of writing code to implement designs. *Offered odd years, fall semester. Prerequisite: Computer Science 204.*

CSCI 370. Advanced Topics in Computer Science (1/4 - 1). Selected aspects of computer science reflecting particular interests and experience of the instructor. Course title and content vary, and the course may be repeated for credit when the title and content change. *Offered occasionally. Prerequisite: varies with topic.*

CSCI 381. Software Development Projects (1/2). In this developmental course, students learn from one another as well as from the instructor. Students work in teams to enhance an ongoing software project through design, implementation, testing, and documentation; teams regularly present ideas, progress reports, and designs. Programming is done in pairs, pairing a more experienced student with a less experienced student. Students learn current design and programming tools and give presentations on topics of current professional interest, including ethical considerations. Computer Science majors are expected to take this course each semester in which they are in residence during their junior and senior years. May be taken up to 4 times. *Offered each semester. Prerequisite: junior standing. Computer Science 204; Computer Science 211 is recommended.*

CSCI 390. Special Projects (1/4 - 1). Individual, guided investigation of a problem or topic in computer science. *Prerequisite: sophomore standing.*

CSCI 395. Teaching Assistant (1/2). Work with faculty in classroom instruction. *Graded credit/no credit.*