

Computer Science (CS)

CS 1030. Problem Solving with Computers. 3 Hours.

For any student interested in how computers are used to solve problems. This course will introduce the use of computers in problem solving including problem decomposition and algorithm construction. Students will be required to complete simple programming projects. Offered based upon sufficient student need. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Read and write small computer programs and a simple web page. 2. Communicate through discussion and writing about data and its effect on daily life. 3. Work with peers in creating, writing, and evaluating computer programs. Course fee required.

CS 1400. Fundamentals of Programming. 3 Hours.

Required of all students pursuing Computer and Information Technology degrees. Open to all students with a general interest in computer programming. Covers structured programming techniques and the syntax of a high level programming language through completion of programming projects of increasing difficulty. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Read and write small computer programs. 2. Use language components such as variables, conditionals, and lists. 3. Decompose small problems. Course fee required. Prerequisites: CS 1030 (Grade C or higher); OR MATH 1010 or higher MATH course (Grade C or higher); OR ACT math score of 23 or higher or equivalent placement score within two years of enrollment in this course. FA, SP, SU.

CS 1410. Object Oriented Programming. 3 Hours.

Required of all students pursuing Computer and Information Technology degrees, open to all students with a general interest in computer programming. Introduces object oriented programming techniques through completion of programming projects of increasing difficulty. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Construct computer programs in a modern development environment using standard tools. 2. Develop solutions using a range of programming constructs, including control structures, functions, input/output, classes and objects, and data collections. 3. Design and implement programs from English descriptions. 4. Demonstrate the use of correct syntax and semantics in a high-level programming language. Course fee required. Prerequisites: CS 1400 (Grade C or higher). FA, SP.

CS 2100. Discrete Structures. 3 Hours.

Required of students pursuing degrees in Computer Science, Software Engineering, and others within the Computing department. Open to any student with an interest in counting theory and applications. Covers mathematical reasoning, combinatorial analysis, sets, permutations, relations, computational complexity, and Boolean logic through homework and programming assignments. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Apply the principles of logic and set theory to solve computational and combinatorial problems. [CS/SE PLO #2] 2. Enumerate discrete structures of a given kind and size via the use of combinations, permutations, and other combinatorial constructs. [CS/SE PLO #2] 3. Solve complex problems by reducing them into simpler sub-problems and finding patterns. [CS/SE PLO #5] 4. Implement software related to discrete math topics, including a modern cryptography system such as RSA. [CS/SE PLO #1, #2, #3] Prerequisites: CS 1410 (Grade C or higher). FA, SP.

CS 2320. Introduction to Machine Learning. 3 Hours.

This course offers a beginner-friendly introduction to the fundamental concepts and applications of machine learning using public libraries. Designed for students with no prior experience in the topic. Open to all students with a general interest in machine learning and artificial intelligence. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Grasp the foundational concepts of machine learning. 2. Explore real-world applications of machine learning using public libraries. 3. Gain hands-on experience by applying foundational machine learning concepts to practical scenarios. 4. Develop the ability to communicate machine learning concepts and results. Prerequisites: CS 1400 (Grade C or higher). SP.

CS 2420. Introduction to Algorithms and Data Structures. 3 Hours.

Required of students pursuing a Computer Science, Software Engineering, and others within the Computing department. Open to any student with a strong interest in computer programming. Covers the design and use of common data structures, lists, stacks, queues, trees, hash tables, and graphs through completion of several challenging programming projects. Introduces computational complexity, algorithm analysis, and NP-Complete using the context of several algorithms including sorting, searching, SAT, Traveling Salesperson, factoring, etc. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Discuss the basic principles of many software data structures, including efficiencies and tradeoffs. 2. Implement and use several data structures, including Binary Search Trees, Hash Tables, Graphs, and more. 3. Demonstrate a working knowledge of Big-O complexity and Algorithm Analysis. 4. Implement several recursive algorithms. 5. Implement and analyze several sorting algorithms. Course fee required. Prerequisites: CS 1410 (Grade C or higher). FA, SP.

CS 2450. Software Engineering. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis, open to any student with a strong interest in computer programming. Covers current software engineering theory and practice through completion of a challenging team project. Dual listed with SE 2450 (students may take only one course for credit). ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Explain the software engineering knowledge, skills, and professional standards necessary to begin practice as a software engineer. 2. Apply and compare appropriate theories, models, and techniques that provide a basis for the software development lifecycle. 3. Construct reliable software artifacts, both individually and as part of a team. 4. Evaluate trade-offs in software engineering practices and determine appropriate balances in project decision making. 5. Employ new models, techniques, and technologies as they emerge and appreciate the necessity of such continuing professional development. Course fee required. Prerequisites: CS 1410 (Grade C or higher). FA, SP.

CS 2500. Data Wrangling. 3 Hours.

This course provides an introduction to Data Science by first giving a high-level overview of the general process and then providing depth in the first phase of the process, including acquiring and preparing large and complex datasets for analysis. Students will apply these concepts to real datasets using state-of-the-art development tools. The course will include class discussions and readings covering case studies relevant to data ethics. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Describe the common phases of data science. 2. Prepare solutions to acquire, clean, and transform raw datasets in preparation for analysis. 3. Experiment with advanced data science topics, such as mining and visualization. 4. Practice learned concepts using professional-grade development tools and libraries. 5. Evaluate case studies related to ethical, bias, and privacy issues in data science. Prerequisites: CS 1410 (Grade C or higher). FA.

CS 2810. Computer Organization and Architecture. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis, open to any student with a strong interest in computer programming. Covers digital hardware design and systems programming, including numeric representations, digital logic, processor architecture, instruction sets, assembly language, and other low-level programming topics. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Convert between number systems including binary, hexadecimal, octal, and decimal. 2. Debate and compare the design of computer instruction sets and assembly languages. 3. Compose low-level solutions to programming problems that interact directly with the operating system. 4. Generate structured assembly language solutions to algorithmic problems. Course fee required. Prerequisites: CS 1410 (Grade C or higher) and can be concurrently enrolled. FA, SP.

CS 3005. Programming in C++. 3 Hours.

For student pursuing degrees in Computer Science and Computer and Information Technologies, or any student with a strong interest in computer programming. Covers syntax and semantics of C++ programming language through completion of hands-on projects. The student must already be fluent in some other programming language. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Construct computer programs in C++, using functions, classes and STL elements. 2. Construct computer programs using stack, heap and static memory. 3. Construct computer programs in a statically typed language. 4. Construct and use unit tests. 5. Use version control to manage code. 6. Use memory checking and debugging tools. 7. Create larger programs than in previous course work. Course fee required. Prerequisites: CS 1410 (Grade C or higher). FA, SP.

CS 3150. Computer Networks. 3 Hours.

A comprehensive introduction to the principles of computer networks from a developer's perspective, with emphasis on the design and implementation of the Internet, its protocols, and applications. Topics include network applications, network programming interfaces, layered network architectures, transport and congestion control protocols, routing and data link protocols, local area networks, and a selection of special topics. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Identify, interpret, and analyze the basic principles of computer networks, including switching, layering and abstraction, routing, and the various protocols that drive network behavior. 2. Explain and implement how applications use networks and the Internet to communicate using network programming interfaces. 3. Assemble limited components of common Internet applications such as email, video streaming, and peer-to-peer applications. 4. Construct portions of the Internet, including transport protocols and routing algorithms, and justify their design. Course fee required. Prerequisites: CS 2420 (Grade C or higher) AND CS 2810 (Grade C or higher). FA, SP.

CS 3400. Operating Systems. 3 Hours.

Can be used to fulfill a requirement for students pursuing a degree or emphasis in Computer Science, and open to other students. Covers operating systems design and implementation, including processes and threads, synchronization, virtual memory, and file systems. Course taught by arrangement. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Design the major components of an operating system. 2. Analyze the trade-offs between competing goals in system software projects, including safety, performance, convenience, and ease of future maintenance. 3. Build and modify complex software projects in teams. 4. Assess and criticize the design of modern and historical operating systems. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher); AND CS 3005 (Grade C or higher). FA.

CS 3410. Distributed Systems. 3 Hours.

Can be used to fulfill a requirement for students pursuing a degree or emphasis in Computer Science, and open to other students. Covers design and implementation of network applications, including message passing, concurrency, synchronization, scalability, and partial failure. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Design and implement software solutions that span multiple computers across a network. 2. Analyze the tradeoffs between competing goals in system software projects, including safety, performance, convenience, and ease of future maintenance. 3. Build and modify complex software projects in teams. 4. Debate and differentiate the approaches and solutions to distributed systems problems taken by modern internet organizations. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher). SP.

CS 3500. Game Development. 3 Hours.

For students pursuing degrees in Computer Science, Software Engineering, and others within the Computing department. Open to any student with an interest in computer 3D Game Design and Development. Covers rapid game development using a modern 3D game development engine. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Construct a game design document. 2. Rapidly implement a new game using a modern 3D game development engine. 3. Find readily available resources to use as game models, environments, animations, sounds, etc. 4. Implement a modern User Interaction and Control system. 5. Create a very basic Virtual Reality (VR) system. Prerequisites: CS 3005 (Grade C or higher). FA.

CS 3510. Algorithms. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis. Covers the analysis and design of algorithms and data structures, including graphs, greedy algorithms, divide and conquer algorithms, and dynamic programming. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Analyze and classify algorithms of various types, including divide-and-conquer algorithms, graph algorithms, dynamic programming, and linear programming. 2. Assess the run-time complexity of algorithms through analysis and measurement. 3. Evaluate and select suitable algorithms for programming problems. 4. Construct algorithmic solutions to complex problems using the full range of algorithmic approaches. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher); AND CS 2100 (Grade C or higher) OR CS 3310 (Grade C or higher). SP.

CS 3520. Programming Languages. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis. Covers the principles and concepts that characterize high-level computer programming languages, including function and data abstraction, and imperative, functional, logic and object-oriented programming techniques. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Compare major programming paradigms and appraising the impact each has on how programming problems are solved. 2. Differentiate between syntax and the underlying semantics that make up modern and historical languages. 3. Compose and construct software solutions from a wide range of fundamental language constructs. 4. Assess new languages as they emerge and determine their suitability for practical programming projects. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher). FA.

CS 3530. Computational Theory. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis. Covers the theory of computation, including finite-state automata, pushdown automata, Turing machines, and equivalent formalisms. Also introduces complexity theory. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Compare the capabilities of various computational models and formulating new models as needed to research new classes of problems. 2. Analyze formal systems with mathematical rigor and the appropriate formal notation. 3. Investigate computational problems and categorizing their algorithmic complexity. 4. Appraise and justify the limits of computational models and the real-world systems that rely on them. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher); and CS 2100 (Grade C or higher) or CS 3310 (Grade C or higher) (can be concurrently enrolled). FA.

CS 3600. Graphics Programming. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis, and open to other interested students. Covers 2-D and 3-D model creation, transformation, and various rendering techniques through completion of programming assignments. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Program 2D and 3D graphics applications using an appropriate API such as OpenGL. 2. Design software that is highly user interactive. 3. Apply modeling, rendering and animation techniques to a variety of problems. 4. Design lighting systems that use the 3D Phong shading model. 5. Render high quality images using texture mapping, ray tracing, and shadows. 6. Program Curves and Surfaces for use in CAD. 7. Program automated systems using Physically Based Modeling. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 3005 (Grade C or higher). SP.

CS 4300. Artificial Intelligence. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis. Introduces the broad field of artificial intelligence in computer software followed by specific applications in computer gaming strategies. Students will complete programming assignments. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Construct solutions for a range of problems using search algorithms. 2. Infer problems that can be solved using propositional logic and build appropriate solutions. 3. Design and implement Bayesian networks. 4. Integrate third-party libraries into solutions for large software projects. 5. Collaborate to solve large and complex problems. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher); AND CS 3005 (Grade C or higher). FA.

CS 4307. Database Systems. 3 Hours.

An introduction to database systems. Topics include the design and implementation of relational databases (B-trees, indexing, query planning and execution, transactions, ACID semantics) as well as data modeling and querying for applications using databases. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Design and implement the major components of a database management system. 2. Analyze the storage needs of a software project and create a data schema and query strategy to address those needs. 3. Construct substantial software that balances requirements of data safety, performance, and complexity. 4. Debate and criticize modern approaches to data management. Course fee required. Prerequisites: CS 2420 (Grade C or higher) AND CS 2810 (Grade C or higher). SP.

CS 4320. Machine Learning. 3 Hours.

For students pursuing degrees in Computer Science or related fields, with an interest in the theory and practice of machine learning. Covers an introduction to supervised and unsupervised learning, including decision trees, neural networks, naive Bayes classifiers and support vector machines. Students will be required to implement machine learning systems. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Use supervised and unsupervised learning techniques. 2. Implement software learning systems. 3. Evaluate quality of learned systems. 4. Implement software utilizing the results of learning systems. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher); AND CS 3005 (Grade C or higher). SP.

CS 4400. Data Mining. 3 Hours.

This course explores the fundamental algorithms and statistics commonly used to extract patterns from large datasets. Students will practice discovering, cleaning, transforming, and exploring real datasets, actively applying the concepts discussed in the course using common programming tools. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Construct automated solutions to discover and extract patterns from large datasets. 2. Describe common data mining algorithms such as clustering and regression. 3. Apply strategies for large datasets such as streaming and sampling. Prerequisites: CS 2420 AND CS 2810 (Grade C or higher). SP.

CS 4410. Data Visualization. 3 Hours.

This course covers the fundamentals of data visualization, including the types of visualizations and techniques to accommodate different types of data. Students will use state-of-the-art programming tools to practice communicating the meaning of patterns, anomalies, and insights found in real datasets. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Compare and contrast common types and approaches of data visualization. 2. Justify the use of appropriate visualization approaches for different types of datasets. 3. Create visualizations that effectively communicate insights from large datasets. Prerequisites: CS 2420 AND CS 2810 (Grade C or higher). FA.

CS 4550. Compilers. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis. Covers compiler design and implementation, including lexical analysis, parsing, symbol table management, and generating code through challenging programming assignments. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Create Grammars and Finite Automata corresponding to Regular Languages. 2. Scan a text file (of programming code) using a Finite Automata to identify the Tokens. 3. Parse a sequence of Tokens using Context Free Grammar to build a Parse Tree. 4. Interpret a Parse Tree to run the original programming code. 5. Write and execute machine code corresponding to a Parse Tree. 6. Build and maintain a Symbol Table to keep track of all variables in programming code. Course fee required. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher); AND CS 3005 (Grade C or higher). SP.

CS 4600. Senior Project. 3 Hours.

Required of students pursuing a Computer Science degree or emphasis. Students will complete an aggressive programming project of software engineering. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Have practical experience in project specification. 2. Have practical experience in project design. 3. Have practical experience in project implementation. 4. Have practical experience in project testing. Course fee required. Prerequisite: Advanced Standing. SP.

CS 4800R. Undergraduate Research. 1-6 Hours.

For Computer Science students who wish to engage in an undergraduate research project. Students will meet weekly with their faculty mentor to discuss progress on their project and receive feedback. At the end of the semester, the student will submit a written defense of their work. Students who expect to apply to graduate school should strongly consider taking this course. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Survey state-of-the-art research. 2. Identify relevant research problems. 3. Articulate research goals by formulating a research problem and developing a research plan. 4. Demonstrate proficiency in Executing a research plan and defending the resulting contributions. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher). FA, SP.

CS 4920R. Internship. 1-3 Hours.

Internship course in Computer Science and Software Development. Variable credit 1.0 - 3.0. Repeatable up to 3 credits subject to graduation restrictions. Offered by arrangement. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Design and implement programming solutions to meet user needs. 2. Use current software development tools and techniques. 3. Develop software in a team environment. 4. Work with an employer. Prerequisites: CS 2420 (Grade C or higher); AND CS 2810 (Grade C or higher); AND CS 3005 (Grade C or higher); AND instructor permission.

CS 4990. Special Topics in Computer Science. 0.5-3 Hours.

For students wishing instruction that is not available through other regularly scheduled courses in this discipline. Occasionally, either students need some type of non-traditional instruction, or an unanticipated opportunity for instruction presents itself. This course may include standard lectures, travel and field trips, guest speakers, laboratory exercises, or other nontraditional instruction methods. Repeatable for credit as topics vary, up to 6 credits. Offered by arrangement. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Develop and build systems using a specific software framework or methodology. 2. Extrapolate the specialized insights and practices of a specific computational system to a wider field of practice. 3. Apply general purpose algorithmic and problem solving skills to a specific problem domain. Course fee required. Prerequisites: Instructor permission.

CS 4991R. Competitive Programming. 0.5 Hours.

For students interested in competing in programming contests. Covers problem analysis and classification, and efficient implementation of solutions. Repeatable up to 6 times for 3 credits. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Network with students interested in competing in programming contests. 2. Compare and contrast different problem types. 3. Survey possible solutions to common problem types. 4. Implement solutions to various contest problems. 5. Transfer the aforementioned skills to real competitions. Prerequisites: CS 1400 (Grade C or higher). FA, SP.

CS 4992R. Computer Science Seminar. 1 Hour.

For students interested in exploring computer science research topics, the Computer Science Seminar will discuss the problems and contributions outlined in a variety of research papers. Participants will read from a selection of assigned research papers and participate in the seminar discussions. Students considering graduate school or who are desirous to participate in undergraduate research are highly encouraged to attend. Repeatable up to 4 times for 4 credits. ****COURSE LEARNING OUTCOMES (CLOs)** At the successful conclusion of this course, students will be able to: 1. Recall common topics in computer science research. 2. Survey impactful research papers and identify the top computer science conferences. 3. Outline and analyze the contributions of a research paper. 4. Classify and discuss research problems along with known solutions to those problems. Prerequisites: CS 2420 (Grade C or higher) OR CS 2810 (Grade C or higher). FA, SP.

CS 4995. Programming for VR/XR Internship 01. 3 Hours.

Course for students seeking a certificate in Computer Science for VR. This is the first of two Seminar Courses in which students will work collaboratively with Design students and Art students on VR/XR projects. **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to, on a small scale project conceived by the instructor: (1) Construct elements of working VR/XR/AR applications in a collaborative environment. (2) Produce coded assets using industry-standard software and practices. (3) Integrate assets from other disciplines into a single usable VR/XR/AR application. Prerequisites: CS 3500 (Grade C or higher)(Can be taken concurrently). FA.

CS 4996. Programming for VR/XR Internship 02. 3 Hours.

Follow-on course from CS 4995. For students seeking a certificate in Computer Science for VR. This is the second of two Seminar Courses in which students will work collaboratively with Design students and Art students on VR/XR projects. Projects in this course may be continuations of larger projects from CS 4995, or they may be new, more advanced projects. **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to, on a larger scale project conceived by the student or by an internship director: (1) Construct elements of working VR/XR/AR applications in a collaborative environment. (2) Produce coded assets using industry-standard software and practices. (3) Integrate assets from other disciplines into a single usable VR/XR/AR application. Prerequisites: CS 4995 (Grade C or higher).