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 statically typed language. 2. Construct and use unit tests. 3. Use version control to manage existing systems, and organizations. 5. Evaluate new development models, techniques, and technologies as they emerge and appreciate the necessity of such continuing professional development. Course fee required. Prerequisites: CS 1400 (Grade C- or higher). FA, SP.

CS 2420. Introduction to Algorithms and Data Structures. 3 Hours.
Required of students pursuing a Computer Science or Information Technology degree or emphasis, 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 and algorithm analysis. **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, and Graphs. 3. Demonstrate a working knowledge of Big-O complexity. 4. Demonstrate a working knowledge of Algorithm Analysis. 5. Parse mathematical expressions into a functioning graphics calculator. 6. Implement several recursive algorithms. 7. Implement and analyze several sorting algorithms. Course fee required. Prerequisite: 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. Introduces current software engineering theory and practice through completion of a challenging team project. Dual listed with WEB 3450 (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. Demonstrate understanding of software engineering knowledge and skills and of the professional standards necessary to begin practice as a software engineer. 2. Apply appropriate theories, models, and techniques that provide a basis for problem identification and analysis, software design, development, implementation, verification, and documentation. 3. Collaborate as part of a team or individually to develop and deliver quality software artifacts. 4. Reconcile conflicting project objectives, finding acceptable compromises within the limitations of cost, time, knowledge, existing systems, and organizations. 5. Evaluate new development models, techniques, and technologies as they emerge and appreciate the necessity of such continuing professional development. Course fee required. Prerequisites: CS 2420 (Grade C- or higher). FA, SP.

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 current software engineering theory and practice through completion of a challenging team project. Dual listed with WEB 3450 (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. Convert between number systems including binary, hexadecimal, octal, and decimal. 2. Debate and compare the design of computer instruction sets and assembly languages. 3. Create larger programs than in previous course work. Course fee required. Prerequisite: CS 1410 (Grade C- or higher). 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. Prerequisite: CS 1410 (Grade C- or higher). FA, SP.
CS 3010. Mobile Application Development for Android. 3 Hours.
For students pursuing degrees in Computer Science, or other students interested in writing applications for modern mobile devices using Google's Android operating system. **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to: 1. Develop modern applications for phones and tablets using the Android SDK and related tools. 2. Design and implement a functional graphical user interface suitable for a mobile application. 3. Structure sophisticated mobile applications using the software architecture and design patterns of contemporary practice. Course fee required. Prerequisites: CS 2420 (Grade C- or higher) AND CS 3005 (Grade C- or higher). SP.

CS 3020. Mobile Application Development: iOS. 3 Hours.
For students pursuing degrees in Computer Science, or other students interested in writing applications for modern mobile devices using Apple's iOS operating system. **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to: 1. Develop modern applications for phones and tablets using the iOS SDK and related tools. 2. Design and implement a functional graphical user interface suitable for a mobile application. 3. Understand the software architecture and design patterns necessary to develop sophisticated mobile applications. Course fee required. Prerequisites: CS 2420 (Grade C- or higher); AND CS 3005 (Grade C- or higher). FA.

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. Prerequisite: CS 2810 (Grade C- or higher). SP.

CS 3200. Web Application Development I. 3 Hours.
For students pursuing a degree in Computer Science or an emphasis in Software Development, or other students interested in writing applications for the modern web. Covers the fundamentals of three-tier web applications, including client-side code for modern browsers, server code using representative languages, and integration with database systems; also covers the protocols that connect these components and the environments in which they run. Daul listed with WEB 3200. Students may only take one course for credit. **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to: 1. Develop modern web applications using both client-side and server-side languages and technologies. 2. Integrate database technologies into the ecosystem of a web application at a fundamental level. 3. Deploy the environments and infrastructure required by web application servers and related systems. 4. Implement the architectures, protocols and standards necessary to interconnect the client-side and server-side components. Course fee required. Prerequisites: CS 1410 (Grade C- or higher) AND WEB 1400 (Grade C- or higher); OR CS 2810 (Grade C- or higher). FA, SP.

CS 3310. Discrete Mathematics. 3 Hours.
For students pursuing degrees in Computer Science, or other students interested 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. 2. Enumerate discrete structures of a given kind and size via the use of combinations, permutations, and other combinatorial constructs. 3. Reduce complex problems into simpler sub-problems. 4. Implement software related to discrete math topics. Course fee required. Prerequisites: MATH 1100 (Grade C- or higher) OR MATH 1210 (Grade C- or higher); AND CS 1410 (Grade C- or higher). FA.

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 3440. Software Practices. 3 Hours.**
For students pursuing degrees in Computer Science, or other students interested in gaining experience in software development practices. Covers practical usage of software development tools, source code control, software debugging, third party libraries and frameworks, and effective team work. **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to: 1. Design software development workflows that use modern tools. 2. Analyze the tradeoffs between different development practices for individual and team projects. 3. Build and deploy systems using professional best practices and methodologies. Course fee required. Prerequisites: CS 3005 (Grade C- or higher).**

**CS 3500. Application Development. 3 Hours.**
For students pursuing degrees in Computer Science or Computer Information Technology, or others with an interest in graphical interface design and implementation. Covers the theory and practice of constructing easy to use interfaces through programming graphical environment projects in a variety of languages and platforms. Course taught by arrangement. **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to: 1. Develop GUI software in a variety of environments. 2. Be proficient in a variety of software development environments that are relevant to our local economy. Course fee required. Prerequisite: CS 3005 (Grade C- or higher).**

**CS 3510. Advanced Algorithms/Data Structures. 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 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 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 3310 (can be concurrently enrolled). 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 3310 (can be concurrently enrolled). FA.**

**CS 3560. 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. Learn the syntax and use of an appropriate graphics API such as OpenGL. 2. Learn raster display concepts, input and output device management, human and camera visual systems, and a computer visual system. 3. Learn about frame buffers, pixel manipulation, lines, circles, text, and other basic primitives. 4. Learn how an RGB color display physically works, and how to program it as desired. 5. Learn viewing transformation, modeling transformation, and the Graphics Pipeline. 6. Learn to program the mouse and keyboard to respond appropriately to user input. 7. Learn modeling, rendering and animation techniques. 8. Learn the 3D Phong shading model. 9. Learn other graphics techniques such as hidden surface removal, texture mapping, and antialiasing. 10. Be briefly introduced to Curve and Surface modeling. 11. Be briefly introduced to Ray Tracing. 12. Be briefly introduced to Physically Based Modeling. 13. Be briefly introduced to Fractals. 14. Create several 2D and 3D graphics applications. Course fee required. Prerequisites: CS 2420 (Grade C- or higher); AND CS 3005 (Grade C- or higher). SP.**

**CS 4200. Web Application Development II. 3 Hours.**
For students interested in writing applications for the modern web. Covers advanced concepts and topics in client-side and server-side web application development. Students will be introduced to a variety of modern software frameworks, languages, architectural patterns, and techniques in order to create interactive, data-centric web applications. Course is dual listed with WEB 4200. Students may only take one course for credit. **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to: 1. Develop modern web applications using both client-side and server-side languages and technologies. 2. Assess the makeup of various client-side and server-side web application frameworks and their constituent components. 3. Create an interactive user experience using a client-side framework and interaction with a web service. 4. Implement the architectural and design patterns used by web application frameworks, and justify how they are used to produce maintainable and scalable web applications. Course fee required. Prerequisites: CS 3200 (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. Discern 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 Design & Management. 3 Hours.
Required of students pursuing a Computer Science degree or emphasis. Covers administration of database management systems, logical database design, implementation of database designs, and application development using a DBMS. Students will design, manage, and implement databases and applications that use databases. Dual listed with IT 4300 (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. 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) AND CS 3005 (Grade C- or higher). SP.

CS 4310. Database Administration. 3 Hours.
This course covers the database architecture and environment. Students will be able to manage user access control. Students will be able to perform backup, restore, and recovery operations. Students will be able control performance and optimization issues. It covers updating and upgrading of a database system. Students will be able to perform the importing and exporting of data to/from a database. Dual listed with IT 4310 (only one course may be taken for credit). **COURSE LEARNING OUTCOMES (CLOs) At the successful conclusion of this course, students will be able to: 1. Manage and organize data into a database. 2. Backup and restore a database. 3. Tune a database for better performance performance. 4. Import/export data to and from a database. Course fee required. Prerequisite: CS 4307 (Grade C- or higher). FA.

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 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. Be further exposed to computer science theory, including Languages, Grammars, and Machines. 2. Study and implement Scanning and Lexical analysis. 3. Study and implement Parsing. 4. Study and implement an Interpreter. 5. Study and implement Machine Code Generation on a modern CPU. 6. Study and implement a Storage and Symbol Management system. 7. Review the intricacies of C++. 8. Implement a very large project using correct Software Engineering principles. 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: Senior status. 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. 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. Effectively compete in programming competitions. Prerequisite: CS 1400 (Grade C- or higher). FA, SP.

CS 4992. Seminar 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 request some type of non-traditional instruction, or an unanticipated opportunity for instruction presents itself. This seminar course provides a variable credit context for these purposes. As requirements, this seminar course must first be pre-approved by the department chair; second, it must provide at least nine contact hours of lab or lecture for each credit offered; and third, it must include some academic project or paper (i.e., credit is not given for attendance alone). This course may include standard lectures, travel and field trips, guest speakers, laboratory exercises, or other nontraditional instruction methods. Note that this course in an elective and does not fulfill general education or program requirements. Fees may be required for some seminar courses and instructor permission will be optional at the request of the instructor. **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.