# COMPUTER SCIENCE (CSC)

**CSC 101. Introduction to Digital Technologies. 3 Credit Hours.**
The goal of this course is to help students gain a deeper appreciation of the capabilities and limitations of computing. Students will learn core computer science concepts and apply those using tools that they will use as future scientists, data analysts, and decision makers. By understanding software as more than a black box, students are better positioned to understand its value and its results, and make more intelligent decisions about when and how much to rely on sofware results. The course will use tools as Excel, Access, SQL, XML, and NetLogo to demonstrate principles.
Cross-listed Courses: CSC 151, CTS 101

**CSC 151. Introduction to Digital Technologies. 3 Credit Hours.**
The goal of this course is to help students gain a deeper appreciation of the capabilities and limitations of computing. Students will learn core computer science concepts and apply those using tools that they will use as future scientists, data analysts, and decision makers. By understanding software as more than a black box, students are better positioned to understand its value and its results, and make more intelligent decisions about when and how much to rely on sofware results. The course will use tools as Excel, Access, SQL, XML, and NetLogo to demonstrate principles.
Cross-listed Courses: CSC 101, CTS 101

**CSC 155. COBOL Programming. 3 Credit Hours.**
This course is a study of the COBOL programming language, with application of its features for table handling, sorting, sequential and random access file handling and modular programming.
Cross-listed Courses: MIS 326

**CSC 165. Programming in Visual Basic With Visual Basics. 3 Credit Hours.**
This class covers the basics of structured programming using Visual Basic to develop decision support systems or management science applications. The theory and practice of structured programming, logic, systems development are covered in a series of iterative hands-on assignments, which are designed based on practical decision support systems or management science applications. Students can expect to learn how to create and program advanced Excel applications or other equivalent applications. A term project involving the development and documentation of a Visual basic program is required.
Cross-listed Courses: MIS 385

**CSC 170. Introduction to Java Programming Programming. 4 Credit Hours.**
This course will introduce you to programming and object oriented programming using Java. General programming topics include program design, testing strategies, and control structures such as conditionals, iteration. Object oriented topics include creating and using classes, inheritance and interfaces. Students will also learn about basic data structures such as arrays and strings. Students will solve programming problems in weekly lab sessions. Graduate students in the course will also be introduced to event-driven programming using a graphical user interface (GUI), recursion, and 2-dimensional arrays.
Cross-listed Courses: MIS 325

**CSC 175. Introduction to Algorithms and Program Design. 4 Credit Hours.**
This course introduces students to programming with an emphasis on computational problem-solving. Topics include program design and testing strategies, programming language syntax and semantics, scalar data types and an introduction to data structures, control structures, iteration, recursion, file input/output, exceptions as well as an introduction to algorithm analysis. Students will use a high-level programming lanugage to develop programs and reinforce their understanding of topics.
Cross-listed Courses: MIS 175

**CSC 175L. Lab. 0 Credit Hour.**
Cross-listed Courses: MIS 175L

**CSC 179. Transition to Python. 1 Credit Hour.**

**CSC 185. Data Structures & Algorithms. 4 Credit Hours.**
This course studies data structures and algorithm analysis. Topics include data structures such as list, trees, stackes, queues, heaps, hash tables and graphs, and algorithms for searching and sorting, and simple graph algorithms. Students will also learn how to formally analyze algorithms. The emphasis will be on applying data structures to design and implement efficient algorithms. Students will use a high-level programming language to complete several intermediate sized programming projects to reinforce concepts.
Prerequisite: CSC 175 or CSC 179.

**CSC 275. Data Structures and Algorithms. 3 Credit Hours.**
This course studies data structures and algorithm analysis. Topics include data structures such as list, trees, stacks, queues, heaps, hash tables and graphs, and algorithms for searching and sorting, and simple graph algorithms. Students will also learn how to formally analyze algorithms. The emphasis will be on applying data structures to design and implement efficient algorithms. Students will use a high-level programming language to complete several intermediate sized programming projects to reinforce concepts.
Prerequisites: CSC 170 or CSC 176.

**CSC 276. Object Oriented Software Design. 4 Credit Hours.**
This course deals with the general topic of object-oriented software design and introduces an object-oriented programming language. Design strategies (e.g., compositional) and concepts (e.g., functional independence) are discussed in the context of a software design model that contains four elements-architecture, data, interfaces, and components. Different object-oriented software design techniques (e.g., UML), software design metrics (e.g., coupling), and software quality assurance techniques (e.g., review) are discussed and applied to software designs. Also introduced are human-computer interaction, incremental software development using a personal software process, information models and database systems, principles of secure design, defensive programming, threats and attacks, and secure software engineering. Each student will produce design models and a prototype implementation.
Prerequisite: CSC 185.

**CSC 281. Discrete Mathematics. 3 Credit Hours.**
This course covers the fundamental mathematical principles relevant to computer science, applied mathematics, and engineering. Topics included are functions, relations, sets, propositional logic, predicate logic, proof techniques, (with an emphasis on mathematical induction), basics of counting, and discrete probability.
Prerequisite: MTH 145.

Cross-listed Courses: MTH 260

**CSC 289. Bits, Nibbles, and Bytes. 3 Credit Hours.**
This course provides an introduction to computer hardware with an emphasis on architecture and low-level programming. Topics include Boolean logic, design of combinational and sequential circuits, design of computer hardware components such as the CPU, machine level representation of data, and basic machine architecture. Assembly language is used to provide a programming experience where the code directly manipulates the CPU and memory.
Prerequisite: CSC 185.

**CSC 333. Computational Physics. 3 Credit Hours.**
An introduction to computer techniques and simulations emphasizing problem solving in physics and the use of statistical, differential, integral, graphical, and numerical methods. Examples will be drawn from classical, statistical, and quantum mechanics and will include numerical integration, differentiation, and the solution of ordinary and partial differential equations, using programs such as Exel, Maple, Matlab, Mathematica etc.
Prerequisites: MTH 145 and MTH 146.

Corequisite: MTH 245.

Cross-listed Courses: PHY 333

**CSC 346. Software Operating Environments. 3 Credit Hours.**
This course covers operating system principles and design, and focuses on process management, memory management, and device and file management. Performance considerations, including both resource usage and speed, are emphasized. Students will develop design models that describe the functional components of operating systems, and will develop/modify software that simulates selected components of an operating system. (Offered every other spring.)
Prerequisite: CSC 276.

**CSC 355. Programming Languages & Paradigms. 3 Credit Hours.**
This course introduces the theory of programming languages and at least two computing paradigms-functional and logic. Theory topics include syntax, semantics, type systems, program representation, language translation and execution, and memory management. At least one functional and one logic language will be covered, giving students hands-on experience using these paradigms. Choice of programming language and its impact on security will also be discussed.
Prerequisite: CSC 276.

**CSC 361. Cybersecurity for Future Presidents. 3 Credit Hours.**
Future Presidents will need to understand the science, technology, and human considerations behind cyber security well enough to make informed decisions when provided advice and options for action. By adopting the perspective of training future Presidents, this course aims to help all students (whether or not they later seek leadership positions in government or industry) to understand cyber security, privacy, and intellectual property issues from technical and social perspectives. It assumes a basic familiarity with computers including use of modern desktop, mobile and web-based platforms. It is designed for students who have an interest in thinking critically about how technology and cyber security may affect individuals, group, and organizations in 20-30 years. Government by the people dependson a citizenry that understands the issues their leaders must address. This course will provide students the tools to understand and evaluate the actions of future leaders in the area of cyber security.

**CSC 375. Design and Analysis of Algorithms. 3 Credit Hours.**
This course covers at least four major algorithm design techniques: greedy, divide-and-conquer, dynamic programming and network flow algorithms while emphasizing techniques for analyzing algorithms. Designing and analyzing algorithms for parallel machines will also be discussed. Additional topics include computational complexity focusing on NP-completeness, and some algorithmic techniques for dealing with intractable problems including approximation algorithms and local search.
Prerequisites: CSC 185 and CSC 281.

**CSC 390. Independent Study. 1-3 Credit Hours.**
A student who wishes to pursue an independent study project for academic credit must submit, prior to registration, a proposed plan of study that includes the topic to be studied and goal to be achieved, the methodology to be followed, schedule of supervision, end product, evaluation procedure and number of credits sought. The proposal must be approved by the supervising faculty member, the department chair and the academic dean. The proposal will be kept on file in the academic dean's office.

**CSC 411. Introduction of Artificial Intelligence. 3 Credit Hours.**
This course introduces the concepts and uses of artifical intelligence. Topics include heuristic search techniques, branch and bound, game-playing, neural nets, knowledge representation, logic and deduction, planning and an introduction to machine learning.
Prerequisites: CSC 275.

**CSC 421. Numerical Methods. 3 Credit Hours.**
The development of algorithms for and error analysis of: solutions of equations, interpolation and approximation, numerical differentiation and integration, numerical solutions of differential equations. Also, knowledge of a high level programming language. (Offered every other fall.)
Prerequisites: Grade of C- or better in MTH 245 and MTH 261 or permission of the department chair.

Cross-listed Courses: MTH 421

**CSC 425. Introduction to Graphics. 3 Credit Hours.**
This course introduces the concepts and uses of computer graphics. Topics include roster and vector graphics, graphics processing unit, graphics programming libraries, and graphic capabilities found in game engines.
Prerequisite: CSC 276.

**CSC 431. Introduction to Robotics. 3 Credit Hours.**
This course introduces the concepts and uses of wheeled robots. Topics include navigating a robot in a physical setting using various types of sensors (e.g., LiDAR, image, and infrared proximity) and an introduction to the different types and uses of robots.
Prerequisite: CSC 276.

**CSC 441. Secure Software Development in Mobile And Cloud Environments. 3 Credit Hours.**
This course introduces secured software development in two environments - mobile and cloud - with an emphasis on design, construction and testing. The course will also reinforce human-computer interaction, information models and database systems. Each student will produce design models and at least one prototype implementation.
Prerequisites: CSC 275.

**CSC 445. Networks & Cloud Secure Software Development. 4 Credit Hours.**
This course covers net=centric computing by focusing on client-server computing and the internet protocol stack, with emphasis on protocols in the application, transport, and network layers. Also covered are human-computer interaction, agile software development using a personal software process, principles of secure design, defensive programming, threats and attacks, and secure software engineering. Each student will produce design models, a simple client-side application, and a robust server-side component used to illustrate client-server communication. Topics related to server deployment as a cloud service, peer-to-peer computing, and distributed data will also be introduced.
Prerequisites: CSC 276.

**CSC 460. Managing Systems Projects. 3 Credit Hours.**
This course focuses on introductory project management processes, technology and tools, utilizing the Project Management Institute's (PMI) Project Management Body of Knowledge (PMBOK) and the Software Engineering Institute's (SEI's) Capability Maturity Model Integration (CMMI) processes and nomenclature. Students examine the processes and theory of project management as well as industry case studies, and will utilize project management software in support of their management activities. Guest speakers and field research provide students with access and information from industry and academia. Students are engaged in a semester-long project. Initially, they are required to identify the project scope and team charter for their project; subsequent assignments require them to prepare a business case, work breakdown structure, cost estimate, and final project documentation for their project.
Cross-listed Courses: MIS 460, MGT 460, RMI 462

**CSC 471. Models of Computation. 3 Credit Hours.**
While most computer science courses discuss problems which are able to be solved by computers, this course will look at both the capabilities and the limitations of computers. We start by analyzing simple models of computation, including finite state automata, and push down automata, and build up to Turing machines, which are powerful enough to model modern computers including multicore parallel machines. The course explores where the boundary lies between what is possible and impossible to compute on each model to draw conclusions about the nature of computation.
Prerequisites: CSC 375.

**CSC 480. Database Management Systems. 3 Credit Hours.**
This course provides an overview of the concepts and principles of database management systems, blending technical with managerial topics. Students will study the principles of database structures, the database development process, entity-relationship and object-oriented database models, logical and physical database designs, SQL, as well as distributed and object-oriented databases. Students will also examine data warehouses, as well as the challenges of global electronic data management, electronic commerce and ethical issues associated with the increasing integration and complexity of large-scale data sets. Students will complete a database design project during the semester.
Prerequisites: CSC 275 or permission of the instructor.

Cross-listed Courses: MIS 480

**CSC 481. Database Theory. 1 Credit Hour.**
This course introduces students to the theory behind database technologies. Student's knowledge of SQL will be used to discuss relational algebra, relational calculus, normalization, and functional dependencies. Indexing structures(e.g., b-tree, hashing) and their associated performace characteristics and transaction processing (i.e., commits and concurrency issues) will also be discussed. A student must concurrently take CSC 480(MIS 480).

**CSC 490. Internship in Computer Science. 1-6 Credit Hours.**
Participation in a field learning experience in some area of computer science. The student intern reports as required to a supervising faculty member, who will evaluate the internship and its relationship to the student's academic program. May be taken pass/fail only.
Prerequisites: a minimum of four computer science courses; junior or senior standing; and approval of the department.

**CSC 496. Senior Capstone Project. 3 Credit Hours.**
This course, exclusively for senior computer science and software applications and systems development majors, involves the completion and presentation of a software engineering or research project. A small team of students works on a software engineering project while self-managing their project. The project team produces software engineering artifacts, presentation materials, and a prototype implementation. A research project is typically done by students individually and results in a paper, presentation materials, and a prototype implementation or code to support the research agenda.

**CSC 497. Senior Capstone Project. 3 Credit Hours.**
This course, exclusively for senior computer science and software applications and systems development majors, involves the completion and presentation of a software engineering project. A small team of students works on a software engineering project with guidance from a faculty member. The project team produces software engineering artifacts, presentation materials, and a prototype implementation.
Prerequisite: Senior status.

**CSC 498. Cooperative Education Experience. 12 Credit Hours.**