BACHELOR OF SCIENCE-COMPUTER SCIENCE

A Bachelor of Science degree in Computer Science (BSCS) prepares students to excel in an increasingly complex technical world. Computer Science majors study computers, their organization, and the software that runs them. They will study algorithms for solving different real-world problems, methods of algorithm design and analysis. Computer Science majors learn programming languages, methods of software engineering, and the modern approaches to computer programming. They will study discrete mathematics, and other mathematical disciplines, which are essential for the algorithm design, modeling and solving a variety of real-world problems. With a degree in Computer Science from Texas A&M University-Texarkana, students can pursue careers in software development, database administration, computer engineering, systems analyst, computer network architect, web development, information system management, and many other computer technology careers.

Concentrations in Computer Science:

- Cyber Security (http://catalog.tamut.edu/business-engineering-technology/computer-science/cyber-security-concentration/)
- Software Engineering (http://catalog.tamut.edu/business-engineering-technology/computer-science/software-engineering-concentration/)

Undergraduate Courses in Computer Science

COSC 1315. Introduction to Computer Science. 3 Hours.
This course provides a breadth first introduction to the discipline of Computer Science, including substantial coverage of at least one major programming language including coding assignments. Topic coverage will include: an introduction to algorithms, digital logic, computer architecture, networks, information security, compilers and language translation, families of programming languages, simulation and modeling, e-commerce, databases, and artificial intelligence. Students completing the course successfully will be well prepared in the major of Computer Science.

COSC 1321. Discrete Structures. 3 Hours.
This course covers mathematical mechanisms, which are widely used in the computer modeling and simulations. A discrete nature of a digital computer requires considering discrete rather than continuous models. Since to solve any problem using a computer, a proper model must be developed first, discrete structures and corresponding mathematical tools are very important. Thus the following topics are considered in this course: propositional logic and its role in algorithm design and computer programming, sets and operations on sets, relations and functions, mathematical induction, modular arithmetic and its applications, particularly in encryption, graphs, tress, binary search trees, and Boolean functions.

COSC 2318. Engineering Mathematics. 3 Hours.
This course provides the basic concepts of engineering mathematics including, but not limited to, the review of college algebra, elements of linear algebra, probability and statistics, and differential equations. Prerequisite: COSC 1321.

CS 305. Data Structures. 3 Hours.
This course emphasizes the organization of information; the implementation of common data structures such as lists, stacks, queues, trees, and graphs; and techniques of data abstraction, including encapsulation and inheritance. Instructors administer mini-labs and programming assignments. Assignments will focus on the design, implementation, testing, and evaluation of various data structures. Prerequisite: CS 332.

CS 310. Analysis of Algorithms. 3 Hours.
This course introduces basic elements of the design and analysis of computer algorithms. Topics include methods of algorithms description, proving of their correctness, asymptotic notations and analysis, recursion, divide and conquer, and examples of the efficient algorithms design in signal processing. For each topic, besides in-depth coverage, students will discuss one or more representative problems and their algorithms. In addition to the design and analysis of algorithms, students must gain substantial discrete mathematics problem-solving skills essential for computer engineers. Prerequisite: COSC 1321 or MATH 2305.

CS 316. Web Design and Programming I. 3 Hours.
The course provides the student with an understanding of web page creation using HTML5, CSS, JavaScript, and Ajax. Students will learn how to create hyperlinks, headings, lists, tables, formatting, and images using HTML5 and CSS. Students also learn how to validate form, control cookies, make special effects using JavaScript, and apply Ajax technology to create user interaction. Prerequisite: COSC 1315.

CS 332. C++ Programming. 3 Hours.
This course introduces students to C++ programming language, a dominant language in the industry today. Students will be taught the fundamentals of programming. These concepts are applicable to programming in any language. Topics covered include basic principles of programming using C++, algorithmic and procedural problem solving, program design and development, basic data types, control structures, functions, arrays, pointers, and introduction to classes for programmer-defined data types. Frequent homework and lab assignments will be given during class.

CS 352. Java Programming. 3 Hours.
This course teaches the basics of Java programming, the foundations of object-oriented programming, and the process of building a project in a modular fashion. Java programming provides an overview of programming concepts, design, and an introduction to coding using the Java language. This course has a focus on creating working computer programs in Java. It will address fundamental concepts of analysis, design, and testing and code development. These include flowcharts, Boolean logic, control flow, data types and structures, variable arrays, functions, and pointers. This course will prepare students for focused studies in any programming language. The student will also learn how to enter, compile, link, and run a computer program using the Java language in a Windows or equivalent environment. Instructors will introduce structured programming through techniques for solving business, engineering and scientific problems. Laboratory exercises will provide practice in writing programs and will reinforce basic programming concepts, logic flow, and structured design.

CS 353. Advanced Object-Oriented Programming. 3 Hours.
This course provides an overview of advanced object-oriented programming concepts, design and to coding using the C++ language. It has a focus on creating working computer programs in Visual C++. It addresses advanced concepts of analysis, design, testing, and code development. These include but are not limited to flowcharts, Boolean logic, control flow, data types and structures, Inheritance, Polymorphism Templates, Exceptions and Operator Overloading Strings, Streams, Files and advanced Data Structures topics. This course prepares students for focused studies in gaming or other advanced programming arenas. The student learns how to enter, compile, link, and run a computer program using the C++ language in a Windows, Linux, or equivalent environment. Structured programming will be introduced through techniques designed to solve mathematical, scientific, and engineering problems. Laboratory exercises provide practice in writing programs and reinforce advanced programming concepts, logic flow, and structured design. Prerequisites: CS 332.

CS 355. Python Programming. 3 Hours.
This course will provide a broad introduction to Python's major built-in object types such as numbers, lists, and dictionaries. Creating and processing objects with Python statements, and Python's general syntax model. Using functions to avoid code redundancy and package code for reuse. Organizing statements, functions, and other tools into larger components with modules. An introduction to classes, Python's object-oriented programming tool for structuring code. Writing large programs with Python's exception-handling model and development tools, and learning advanced Python tools, including decorators, descriptors, metaclasses, and Unicode processing.

CS 360. Artificial Intelligence. 3 Hours.
This course will introduce the basic principles of artificial intelligence (AI) and its applications. The class will begin by discussing ways to represent knowledge about the world through logic and how to reason logically with that knowledge. The students will learn general principles of rule-based expert systems. Instructors will introduce and analyze techniques, which allow reasoning under uncertainty. Students will consider Bayesion networks and other probabilistic reasoning models. Students will observe basic principles of the learning theory and consider real world applicaitons of AI, such as expert-based systems and natural-language representation. Prerequisite: COSC 1315.

CS 361. Database Systems and Design. 3 Hours.
This course provides the basic concepts of management of database systems. The cousre emphasizes understanding the various database management functions and providing database support for the organization. Topics include types of database models, database design, entity-relationship diagrams, normalization, database-management systems, administration of database security, error recovery, concurrency control, and distributed-database systems. This course focuses on the design of a database starting from the conceptual design to the implementation of a database schema and user interfaces to the database. The course is heavily design oriented. In most of the projects, students have to design and implement a database using a commercial database management system and associated development tools. Students will learn the database query language SQL and the development of applications using PL/SQL. Students use Oracle 10g (SQL, PL/SQL) and SQL Server 2005 database software in this course. Laboratory exercises provide practice in writing programs and reinforce concepts. Cross-listed with MIS 361. Credit for both MIS 361 and CS 361 cannot be awarded.

CS 362. Systems Analysis and Design. 3 Hours.
Study of the methodology for analysis and design of a business information system. Emphasis on critical analysis of existing systems and design of computer based systems. A systems analysis project is required. Cross-listed with MIS 362. Credit for both CS 362 and MIS 362 cannot be awarded. Prerequisite: Computer Literacy, or consent of instructor.

CS 363. Neural Networks and Machine Learning. 3 Hours.
This course provides the basic concepts of neural networks and machine learning including but not limited to biological foundations of neuronal morphology, machine learning concept and its fundamentals, basics of neural information processing, artificial neuron and its activation functions, multilayer feed forward neural networks and back propagation learning, Hopfield neural networks and associative memories, neuro-fuzzy and kernel-based networks, and support vector machines. Laboratory exercises provide experience with design and utilization neural and other machine-learning algorithms using MATLAB and solving real-world classification, prediction, and pattern recognition problems. This will help students to accomplish specified challenges as they build problem-solving skills. Prerequisite: COSC 1315 or ENGR 1201.

CS 367. Software Engineering. 3 Hours.
This course will offer a wide perspective on software design, stages of software development, design of software documentation, and development including requirements analysis, technical design, estimating, programming style, testing and quality, management, and maintenance. A part of the course is a software project, which students shall design.

CS 370. Programming Language Design. 3 Hours.
This course explores the design of high-level languages, criteria for language selection, specification techniques for syntax and semantics, trends in high-level language design, and introduction to programming in LISP. Prerequisite: CS 332.

CS 380. Automata Theory. 3 Hours.
This course is a study of the basic types of abstract languages and their acceptors, the Chomsky hierarchy, solvability and recursive function theory, and application of theoretical results to practical problems. Prerequisite: COSC 1321.

CS 390. Ethics in Technology. 3 Hours.
This course examines ethical issues and moral problems that engineers, computer scientists, and information technology professionals face. This course covers issues such as moral and ethical relevance, professional responsibilities, privacy, intellectual property, risks, and liabilities. Students review case studies of ethical conflicts in work environment and resolve theoretical situations through the application of ethical codes.

CS 410. Operating Systems. 3 Hours.
This course covers the principles and concepts that govern the design of modern computer operating systems. This course covers managing computing resources such as the memory, the processor, and the Input/Output devices. The course also covers algorithms for CPU scheduling, memory and general resource allocation, process coordination and management, and case studies of several operating systems. Operating systems also manage the authentication, accounting, and authorization aspects in a multi-user system. Students will explore issues and limitations imposed on a computing environment by the choice of different operating systems. Prerequisite: CS 332.

CS 420. Computer Networks. 3 Hours.
Students learn the basic computer networking concepts including ISO/OSI and TCP/IP reference model for networking protocols. The topic covers network architectures, communication protocols, physical media, error control, data link control, medium access control, local area networks, network layer, congestion control, and introduction to virtual circuit and datagram network. The course will also include the case studies and lab assignments for existing networks and network architecture. Prerequisite: COSC 1315 or ENGR 1201.

CS 430. Mobile App Development. 3 Hours.
The course provides the student with a strong foundation in Java programming and the confidence to build successful mobile applications. Students will learn how to use the basic functionalities including user input, variables, operations, decision-making controls, lists, arrays, and Web Browsers. Students also learn how to implement audio, display pictures, and create animation and graphics in Android apps.

CS 431. Internship in Computer Science. 3 Hours.
The internship is a work experience that will allow the student to develop skills, gain hands-on business experience, and test career choices and options. The internship will complement and validate the student's academic training.

CS 465. Computer Security. 3 Hours.
This course will provide a broad introduction to host-based and Internet-based computer security. Topics covered include an introduction to cryptography, authentication protocols, access control, database security, intrusion detection, malicious software such as worms and virus propagation, and techniques to secure the Internet such as firewalls, intrusion detection systems, and Web and IP security.

CS 467. Image Processing and Computer Vision. 3 Hours.
This course provides the basic concepts of image processing and computer vision including but not limited to image sensing and acquisition, visual perception, image enhancement (mostly spatial domain image enhancement, but some essential elements of the frequency domain enhancement will be considered), image filtering in spatial and frequency domain, edge detection and image segmentation, elements of morphological image processing, elements of image restoration, image understanding and recognition, elements of color image processing. Laboratory exercises provide experience with design and utilization image processing algorithms using MATLAB and solving real-world problems in medical and satellite image processing, in old images restoration and in digital photography. Students will program different algorithms and use their programs for processing real images. This will help students to accomplish specified challenges as they build problem-solving skills. Prerequisite: COSC 1315 or ENGR 1201.

CS 471. Network Security and Policy. 3 Hours.
This course will provide a broad introduction to attack strategies in the cyber security kill chain, learning how to enhance defensive strategies by improving security policies, hardening networks, implementing active sensors, and leveraging threat intelligence. Learning how to perform an incident investigation, gaining an in-depth understanding of the recovery process, understanding continuous security monitoring and how to implement a vulnerability management strategy. Learning how to perform log analysis to identify suspicious activities.

CS 472. Digital Forensics, Law, and Ethics. 3 Hours.
This course will provide a broad introduction to a comprehensive and integrative introduction to cybercrime. It provides an authoritative synthesis of the disparate literature on the various types of cybercrime, the global investigation and detection of cybercrime and the role of digital information, and the wider role of technology as a facilitator for social relationships between deviants and criminals.

CS 474. Computer Game Programming. 3 Hours.
An introduction to game development for students interested in the technical aspects of making computer games. This course provides the technical and mathematical background to develop a 2D and 3D style games using JavaScript and HTML5, with brief introduction to other high-level languages and their applicability to game development. During the course, students will be introduced to modern game platforms, and the effect of their differences, evolution, and limitations, on game programming. In addition, students will learn the rudiments of game design and the common work flow practices within the industry. Prerequisites: COSC 1315 and CS 310.

CS 480. Innovation Lab. 1 Hour.
This lab course explores the creative approaches of recent and historic innovations in computer science, business, and technology. Through a case study approach, this course cultivates intentional and systematic competencies in students in order to develop innovation leaders capable of solving problems in technology and business settings. Students will draw insights from the most innovative and successful organizations to explore their approaches. Students will also examine the role of failure in innovations throughout history using foundational creative-thinking concepts.

CS 481. Software Project Management. 3 Hours.
This course will provide a broad introduction to basic principles of software project management: planning and estimating, measuring and controlling, leading and communicating, and managing risk. Also covered are relevant topics from CMMI-DEV-v1.2, IEEE/ISO Standards 12207, IEEE Standard 1058, and the PMI Body of Knowledge.

CS 482. Parallel Modeling and Simulation. 3 Hours.
This course will provide a broad introduction to mathematical/computational modeling and analysis developed in the emerging interdisciplinary field of Complex Systems Science. Complex systems are systems made of a large number of microscopic components interacting with each other in nontrivial ways. Many real-world systems can be understood as complex systems, where critically important information resides in the relationships between the parts and not necessarily within the parts themselves.

CS 483. User Design Methodology. 3 Hours.
This course will provide a broad introduction to principles, techniques, and best practices needed to build user experiences for the web, mobile devices, and desktop environments. Coverage includes the entire process, from user personas and stories through wireframes, layouts, and execution. Also addressed are key issues such as telemetry and security implicit in User Design. Resources and artifacts covered include case studies, sample design documents, and UX testing plans.

CS 484. Software Metrics. 3 Hours.
This course will provide a broad introduction to software metrics. The course will cover material relevant to object-oriented design, design patterns, model-driven development, and agile development processes. It includes coverage of causal models and Bayesian networks and their application to software engineering. Recent research incorporating findings relevant to the latest software metrics activities, industrial case studies, and standards will be covered.

CS 485. Capstone in CS. 4 Hours.
The aim of the capstone project in the senior year of Computer Science majors is to familiarize them with the process of solving real-world computational problems as practiced in industry. This course requires students to develop a project based on the knowledge and skills acquired in earlier coursework and integrate their technical knowledge through practical design effort. The work can be performed as a team work or can be performed as an individual project design.

CS 489. Individual Study. 3 Hours.
This course provides individual instruction. Students may repeat the course when topics vary.

CS 490. CS Senior Design I. 3 Hours.
This course is taken by seniors as the first part of the senior design experience in the semester before CS 491. Projects may involve the design of an algorithm, or a software and/or hardware system and topics covered may include the design process, project planning and management, and project costs, and includes aspects of ethics in computer science design, safety, environmental considerations, economic constraints, liability, manufacturing, and marketing. Projects are carried out using a team-based approach and selection and analysis of a design project to be continued in CS 491 is carried out. Written progress reports, a proposal, a final report, and oral presentations are required. Cross-listed with EE 490 and MGT 490. Credit can only be awarded for one course. Prerequisite: Junior or Senior classification.

CS 491. CS Senior Design II. 3 Hours.
Projects involving the design of a device, circuit system, process, or algorithm that have started in the previous semester will be completed. Team solution to an computer science design problem as formulated and initiated in CS 490 will continue to take place. Written progress reports, a final report, design manuals, and oral presentations are required. Cross-listed with MGT 491 and EE 491. Credit can only be awarded for one course. Prerequisite: CS 490; open only to Computer Science majors.

CS 497. Special Topics. 3 Hours.
Instructors will provide an organized class designed to cover areas of specific interest. Students may repeat the course when topics vary. Prerequisite: Instructor permission.

CS 499. Independent Research. 1-6 Hours.
Independent research in Computer Science conducted by a student under the guidance of a faculty member of his or her choice. The student is required to maintain a research journal and submit a project report by the end of the semester and potentially make an oral presentation on the project. SCH and hours are by arrangement and, with a change in content, this course may be repeated for credit. Prerequisite: Consent of instructor.

Faculty

Dr. Michael Pelosi
Associate Professor
Email: michael.pelosi@tamut.edu

Rong Ren
Lecturer
Email: RRen@tamut.edu