CH462, Biochemistry IIA—Central Metabolism: Topics include carbohydrate and lipid metabolism, electron transport and oxidative phosphorylation, and biological membranes (structure, transport, and signal transduction). Three hours of lecture and one two-hour computer laboratory per week. Prerequisite: CH361 with a grade of C or better. (U)(4) Spring

CH463, Biochemistry Laboratory II: Advanced biochemistry laboratory focusing on inquiry-based, student driven research projects, biochemical literature, complex data analysis, experimental design, and presentation of research results. Projects and techniques will build on the methodology introduced in CH363. Writing intensive (WAC). Prerequisites: CH363 with a grade of C or better. (U)(2) Spring

CH465, Environmental Chemistry: This course focuses on the application of basic chemistry principles in understanding the source, fate, transport, and transformation of compounds in natural and polluted environmental compartments. This includes the atmosphere, hydrosphere, and lithosphere. The impact of these on a number of issues will be discussed. Prerequisite: CH351. (U)(3) Occasionally

CH469, Special Topics in Biochemistry: Various advanced topics in biochemistry. The subject matter for study will be listed when the course is offered. Prerequisite: One year of biochemistry or permission of the instructor. (U/G)(3) Occasionally

CH471, Physical Chemistry I: The introduction to quantum chemistry, bonding, spectroscopy, and statistical mechanics. Prerequisites: CH352 or equivalent, MA107 or equivalent, and PH202 or equivalent. (U/G)(3) Spring

CH472, Physical Chemistry II: The introduction to chemical thermodynamics and chemical kinetics. Prerequisites: CH352 or equivalent, MA107 or equivalent, and PH202 or equivalent. (U/G)(3) Fall

CH473, Physical Chemistry Laboratory: Introduction to methods for determining physical phenomena of chemistry. The topics covered will vary by section number, and can include thermodynamics, kinetics, spectroscopy, quantum chemistry, and computational methods. Co- or prerequisite: CH471 or CH472. (U/G)(2) Annually, term varies

CH475, Computational Chemistry: The course covers theoretical methods of computing molecular structure. Based on the structure, thermodynamic properties, reactivity, and spectroscopic measurements (ultraviolet, visible, infrared, and nuclear magnetic resonance) can be computed. Computed values are compared to experimental data. Prerequisite: CH352 with a grade of C or better. (U)(2) Occasionally

CH479, Special Topics in Physical Chemistry: Various advanced topics in physical chemistry. The subject matter for study will be listed when the course is offered. Prerequisite: One year of physical chemistry or permission of the instructor. (U/G)(3) Occasionally

CH492, Chemistry Seminar 2: Attendance at and participation in the chemistry departmental seminar series, including an oral presentation on a research project or literature review. Required for university honors. Prerequisites: CH392 and 28 hours of chemistry. (U)(1) Fall & spring

CH493, Undergraduate Research: Individual research under guidance of a faculty member. Required for university honors. Prerequisites: Fourteen hours of chemistry, permission of the instructor, and permission of the head of the department. (U/G)(1) Fall & spring

CH494, Undergraduate Research: Individual research under guidance of a faculty member. Required for university honors. Prerequisites: Fourteen hours of chemistry, permission of the instructor, and permission of the head of the department. (U/G)(2) Fall & spring

CH499, Honors Thesis: (U)(3) Annually, term varies

CH518, Advanced Placement Workshop for Chemistry High School Teachers: The workshop will focus on the topics found on the AP exam and in college general chemistry (atomic structure, periodic properties, bonding, kinetics, thermodynamics, electrochemistry, equilibrium, solutions, acid/base, and solubility). The focus of the workshop will be on problem solving and course content material with some limited laboratory demonstration activities. (G)(3) Summer

Computer Science and Software Engineering

Administration

Jonathan P. Sorenson, PhD, Department Chair

Professors

Zhi-Hong Chen, PhD; Panagiotis K. Linos, PhD; Jonathan P. Sorenson, PhD

Associate Professor

Ankur Gupta, PhD

Department Website

www.butler.edu/computer-science

Why Study Computer Science or Software Engineering?

These disciplines are an excellent choice for students who enjoy problem solving and are good at mathematics and the sciences. Good communication skills are also a plus, as this type of work is typically done in small groups or as part of a team. Computers and software have become essential tools in all disciplines, so you can pair the study of computer science or software engineering with a major or minor in any other area of study, including the sciences, business, fine arts, or the humanities. Employment prospects in computer science and software engineering are excellent.

Why Study Computer Science or Software Engineering at Butler?

- Our Engineering Projects in Community Service Program combines service learning with the practice of software engineering; student-run teams develop software for charity and nonprofit clients.
- Our supercomputer is available for student research projects, with possible support available through the Honors Program and the Butler Summer Institute.
- Our students learn parallel programming theory and techniques both in the first-year-level CS142 course and in the junior/senior-level CS452 course.
- Most of our students add a second major or minor and still graduate on time.
- Quality internships and part-time jobs are plentiful in the Indianapolis area.
- Our alumni advisory board is active and engaged, not only mentoring our students and visiting for pizza lunches with our students, but also advising the department on curriculum and career preparation.

Our software engineering program prepares students for professional careers as software system designers, developers, testers, maintainers, and project managers. Our computer science program is more broad-based and prepares students not only for a variety of careers in the computing disciplines, such as systems or database administration and web development, but also for graduate study either in computer science or a related discipline, or for professional programs such as law school.

Degree Programs

- Major in Computer Science (BS, BA)
- Major in Software Engineering (BS, BA)
- Minor in Computer Science

Computer Science Student Learning Outcomes

- Analyze and solve mathematics-based problems
- Understand and explain the main concepts, principles, algorithms, data structures, and theories of computer science
- Write programs and develop software to solve problems
- Communicate and work effectively in teams
- Articulate one's role in society as a computing professional, including ethical, legal, and social obligations

Requirements for the Computer Science Major

All of the following courses:

- MA106–107, Calculus and Analytic Geometry I, II
- MA215, Linear Algebra
- CS151 and CS252, Foundations of Computing I, II
- CS248, Object-Oriented Programming and Data Structures
- CS382 or CS383, EPICS II
- CS321, Computer Organization
- CS351, Algorithms
- SE361, Object-Oriented Design
- CS333, Theory of Database Systems
- CS452, Parallel Algorithm Design and Programming
- CS485, Computer Ethics
- CS473, Topics in Computer Science
- 1 course numbered CS440–459
- 1 course numbered CS430–439 or SE460 or above

Requirements for the Computer Science Minor

- CS151, Foundations of Computing I
- CS248, Object-Oriented Programming and Data Structures
- 12 additional credit hours of computer science or software engineering electives numbered 250 or above

Software Engineering Student Learning Outcomes

- Analyze and solve mathematics-based problems
- Understand and explain the main concepts, principles, algorithms, data structures, and theories of computer science and software engineering
- Write programs and develop software to solve problems
- Communicate and work effectively in teams
- Articulate one's role in society as a computing professional, including ethical, legal, and social obligations
- Acquire and synthesize new knowledge for software development in specific application domains

Requirements for the Software Engineering Major

All of the following courses:

- MA106 and MA107, Calculus and Analytic Geometry I, II

- MA162, Probability and Statistics
- CS151 and CS252, Foundations of Computing I, II
- CS248, Object-Oriented Programming and Data Structures
- CS382 or CS383, EPICS II
- CS321, Computer Organization
- CS351, Algorithms
- CS333, Theory of Database Systems
- CS452, Parallel Algorithm Design and Programming
- CS485, Computer Ethics
- SE361, Object-Oriented Design
- SE461, Managing Software Development
- SE462, Modernizing Legacy Software
- SE463, Software Testing and Quality Assurance
- SE411, Internship

One of the following two courses:

- CS435, Computer Networks
- CS441, Organization of Programming Languages

Three additional credit hours of computer science or software engineering electives numbered 300 or above.

Core Course Offered by Computer Science

AR220-CS, Robot Programming: This introductory programming course features personal robots that can move, draw, and take digital pictures. Robot behaviors are programmed and controlled remotely using a high-level language such as Python from a desktop or laptop computer. Topics include conditional execution, repetition, defining functions, and using arrays. No prior programming experience required. (U)(3) Fall & spring

Computer Science Courses

CS142, Introduction to Computer Science and Programming: An introduction to programming in a high-level language (assignment, data types, expressions, selection, loops, functions, arrays), including parallel programming for supercomputers. Topics such as AI, software engineering, and databases are also discussed. Prerequisite: MA101 or equivalent. (U)(3) Fall, spring, & summer

CS151, Foundations of Computing 1: Introduction to mathematical problem solving, with emphasis on techniques for designing computer-based solutions. Concepts include problem-solving principles, logic, proof techniques, sets, sequences, functions, relations, and inductive and recursive thinking. Prerequisites: MA101 or MA102 or equivalent, and a declared major or minor in CS/SE. (U)(3) Fall

CS248, Object-Oriented Programming and Data Structures: This course is an introduction to object-oriented programming using Java. Topics include algorithm analysis; recursion; the stack, queue, tree, and heap data structures; sorting algorithms; and GUI programming. A brief survey of computer science is also included: history, software engineering, computer organization, operating systems, networks, programming languages, databases, artificial intelligence, and theory. Prerequisites: CS142 or equivalent and CS151. (U)(5) Spring

CS252, Foundations of Computing 2: As a continuation of CS151, concepts include mathematical logic, formal grammars, algebraic structures, finite-state machines and automata, graph theory, and combinatorics. Prerequisite: CS151. (U)(3) Spring

CS282, EPICS 1 Service Learning: Supervised team software project for a local charity or nonprofit organization. May be repeated for credit. Typically taken by non-majors. Prerequisite: CS142 or SE267 or equivalent. (U)(2) Fall & spring

CS283, EPICS 1 Service Learning: Supervised team software project for a local charity or nonprofit organization. May be repeated for credit. Typically taken by non-majors. Prerequisite: CS142 or SE267 or equivalent. (U)(3) Fall & spring

CS300, Teaching Practicum: Students assist a faculty member teaching a 100- or 200-level CS course by helping students with assignments and laboratory exercises, conducting help sessions, preparing course materials, and setting up laboratory exercises. The student receives regularly scheduled supervision from the faculty instructor. May be repeated once for credit. (U)(3) Occasionally

CS308, Problem Seminar: A survey of basic problem-solving strategies such as recursion, divide and conquer, and backtracking, and applying these strategies to sample problems in mathematics, text processing, graphics, and games. Prerequisite: CS248 or permission of the instructor. (U)(1) Fall & spring

CS321, Computer Organization: Principles of computer architecture are introduced from a layered point of view, beginning at the level of gates and digital logic, and progressing through micro-programming, the machine language execution cycle, addressing modes, symbolic assembly language, and the fundamentals of operating systems. Advanced topics including pipelined and parallel architectures are also covered. Corequisite: CS248. (U)(3) Fall

CS333, Database Systems: An introduction to the theory, design, and use of modern database management systems. Topics include the relational, entity-relationship, and object-oriented data models, query languages such as SQL, file systems, concurrency and deadlock, reliability, security, and query optimization. Prerequisites: CS248 and CS252. (U/G)(3) Fall & spring

CS351, Algorithms: A systematic study of data structures and algorithms with an introduction to theoretical computer science. Topics include lists, stacks, queues, trees, and graph structure, searching and sorting algorithms, mathematical algorithms, time and space complexity, an introduction to the theory of NP-completeness, and an introduction to computability theory. Prerequisite: CS248. (U)(3) Fall & spring

CS382, EPICS 2 Service Learning: Supervised team software project for a local charity or nonprofit organization. May be repeated for credit. Prerequisite: CS248. (U)(2) Fall & spring

CS383, EPICS 2 Service Learning: Supervised team software project for a local charity or nonprofit organization. May be repeated for credit. Prerequisite: CS248. (U)(3) Fall & spring

CS403, Independent Study: Provides an opportunity for qualified students to pursue special topics under the guidance of a department staff member. Prerequisite: Permission of the department. (U/G)(3) Occasionally

CS411, Internship: Supervised work experience directly related to major area of study. Prerequisites: Junior or senior standing and permission of the department. (U)(3) Occasionally

CS431, Theory of Operating Systems: Introduces the major concept areas of operating systems principles, including the study of process, storage, and processor management; performance issues; distributed systems; and protection and security. Prerequisites: CS248, CS252, and CS321. (U/G)(3) Occasionally

CS435, Computer Networks: An introduction to computer networks from a layered point of view, beginning with the physical and data link layers and progressing through the medium access layer, the network layer, the transport layer, and the applications layer. Specific content includes Ethernet, TCP/IP, and the web. Students will write client/server programs that communicate across a network. Prerequisite: CS321. (U/G)(3) Occasionally

CS441, Organization of Programming Languages: Emphasizes the principles and programming paradigms that govern the design and implementation of contemporary programming languages. Includes the study of language syntax, processors, representations, and paradigms. Prerequisites: CS252, CS321, and SE361. (U/G)(3) Occasionally

CS447, Computer Graphics: An introduction to the mathematical basis of computer graphics and graphics algorithms. Topics include 2D and 3D graphics, transformations, perspective, and animation. Prerequisite: CS248. (U)(3) Occasionally

CS451, Theory of Computation: Basic theoretical principles of computer science that are embodied in formal languages, automata, computability, and computational complexity. Includes regular expressions, context-free grammars, Turing machines, Church's thesis, and unsolvability. Prerequisites: CS252, CS321, and CS351. (U/G)(3) Occasionally

CS452, Parallel Algorithm Design and Programming: A study of theoretical and practical paradigms of parallel algorithm design. Topics include model costs, lower bounds, architecture and topology, data-parallelism, synchronization, transactional memory, message passing, and parallel design for sorting, graphs, string processing, and dynamic programming. Prerequisite: CS351. (U)(3) Spring

CS455, Numerical Analysis: Solutions of equations and systems, error analysis, numerical differentiation and integration, interpolation, least squares approximation, and numerical solution of ordinary differential equations. Prerequisites: MA107 and CS142 or equivalent. (U/G)(3) Occasionally

CS458, Introduction to Cryptography and Cryptanalysis: An introduction to classical and public-key cryptographic protocols, including mathematical and algorithmic cryptanalysis. Symmetric key systems include simple substitution and transposition ciphers, the Vigenere cipher, the one-time pad, block ciphers, and quantum cryptography. Public-key systems include RSA, discrete-log systems, Diffie-Hellman and EIGamal, digital signatures, and zero-knowledge protocols. Prerequisites: CS151 or MA205, and CS142 or equivalent. (U)(3) Occasionally

CS473, Topics in Computer Science: In-depth study of special topics not covered in regular courses. Prerequisite: Permission of the department. (U/G)(3) Fall & spring

CS482, EPIS 3 Service Learning: Supervised team software project for a local charity or nonprofit organization. May be repeated for credit. Prerequisite: SE361 and one of CS382 or CS383. (U)(2) Fall & spring

CS483, EPICS 3 Service Learning: Supervised team software project for a local charity or nonprofit organization. May be repeated for credit. Prerequisite: SE361 and one of CS382 or CS383. (U)(3) Fall & spring

CS485, Computer Ethics: Ethical and social issues in computing with emphasis on professional responsibilities, risks and liabilities, and intellectual property. Prerequisites: CS142 and sophomore standing. (U/G)(1) Fall

CS490, Research Methods: An introduction to research methodology in computer science, including an overview of computer science literature and techniques for presenting and evaluating research results. Prerequisites: CS321, CS351, and SE361, or junior standing and permission of the department. (U)(2) Occasionally

CS499, Honors Thesis: (U)(3) Occasionally

Software Engineering Courses

SE267, Business Application Development: Programming in Visual Basic, with applications to business. Topics include data representation, control structures, arrays, functions, and objects. (U)(3) Fall, spring, & summer

SE331, Web-Based Client/Server Programming: Web-based applications programming in a Unix-based environment. Topics include XHTML/CSS and JavaScript on the client side, and PHP and Perl CGI programming on the server side. Other topics include client/server object-oriented software architectures, protection and security related to network programming, and database connectivity. Prerequisite: CS248. (U)(3) Occasionally

SE361, Object-Oriented Design: This course uses the Unified Modeling Language (UML) as a vehicle to introduce the basic principles of object-oriented methodology and design, covering classes, objects, data abstraction, polymorphism, information hiding, and relationships among classes such as inheritance, association, aggregation, and composition. Specific design techniques are covered for object-oriented programming languages such as Java and C++. The course also provides a first exposure to the software development life cycle of object-oriented software applications. A small-team design project is required. Prerequisites: CS248 and CS333. (U)(3) Spring

SE411, Internship: Supervised work experience in software engineering. Prerequisites: SE361, SE461, CS485, and one of SE462 or CS382 or CS383. (U)(3) Occasionally

SE461, Managing Software Development: Techniques, principles, and processes for developing large, complex software systems: systems analysis and specification, modeling, design patterns, implementation, validation and verification, quality assurance, and project management. A team-based software project is required. Prerequisite: SE361. (U/G)(3) Occasionally

SE462, Modernizing Legacy Software: Fundamental concepts, principles, techniques, and tools for the maintenance and evolution of legacy software systems. Software maintenance and evolution process models, reengineering, reverse engineering, and program comprehension tools are covered. A modernization project is required. Prerequisite: SE361. (U/G)(3) Occasionally

SE463, Testing and Quality Assurance: Basic concepts, systematic techniques, and tools involved in testing and QA of software systems. Some topics to be covered include black and white box testing techniques, object-oriented testing, regression testing, system integration testing, and planning and reporting of testing activities. Prerequisite: SE361. (U)(3) Occasionally

SE473, Topics in Software Engineering: In-depth study of special topics not covered in regular courses. Prerequisite: Permission of the department. (U/G)(3) Occasionally

Economics

Administration

Hillary Buttrick, JD, Program Director

Professors

Peter Grossman, PhD, Efroymson Chair of Economics; Robert Main, PhD; William Rieber, PhD

Associate Professor

Kathy Paulson Gjerde, PhD

Adjunct Faculty

Robert Kirk, PhD; Theodore Kuhn, MA; Thomas Litkowski, MA; Timothy Zimmer, PhD

Program Website

www.butler.edu/las/economics

Why Study Economics?

Economics provides you with problem-solving and data-gathering skills to make informed decisions in a variety of settings, e.g., government, law, finance, business, and journalism; and for a variety of job responsibilities—from college intern to a newly hired bank credit analyst at one end of the spectrum, to U.S. senator, Supreme Court justice, or CEO of a successful business on the other.

Besides being a strong major in its own right, economics is a nice complement for students who have interests in philosophy, political science, sociology, history, mathematics, psychology, foreign languages, engineering, and English. Indeed, the Nobel Prize in Economics is often awarded to economists who also have a keen interest in one of the above areas.

Why Study Economics at Butler?

You can study economics as a major in the College of Liberal Arts and Sciences and also as a major in the Lacy School of Business. The student in Liberal Arts and Sciences may have a given career path in mind but often does not. He or she knows that economics offers an array of opportunities and takes economics to learn critical-thinking skills and about the dynamic economic environment in which we live. The career choice will follow. The same applies to a student who majors in the Lacy School of Business, but the career choice in that case is more targeted to the business sector. The economics courses taken in either college are the same; the courses outside of economics, though, are different between the two colleges.

Economics Student Learning Outcomes

Students will learn and discuss issues such as how the Federal Reserve creates money and influences interest rates domestically and around the world, why the euro changes in value against the dollar, and the reason behind Zimbabwe's inflation. You'll learn about the "Wealth of Nations" from rich countries (e.g., the United States) to poor ones (e.g., Bangladesh), about growing countries (e.g., China) and countries transitioning to capitalism (e.g., Russia). You'll learn about the invisible hand of the market and the different roles of government in a mixed economy.

The tools in addressing these questions include supply and demand, international trade and exchange rate determination, monetary and fiscal policy, market structure, and statistical relationships.

Degree Programs

- Major in Economics (BA)
- Minor in Economics

Requirements for the Major

The major consists of 34 hours, including the required foundation (13 hours*), other required courses (9 hours), and electives (12 hours).