

COMPUTER SCIENCE

Computer science deals with the theory of computation, the design of computers, and the study of automating algorithmic processes. It studies the theory, experimentation, and engineering that form the basis for the design and use of computers. Computer science involves programming as a tool for problem solving, but it goes well beyond this.

A good comprehension of computer science helps us design better programs and understand how to use computational tools to solve new problems and to understand complex systems.

Advanced Placement

- A student who gets a 4 or 5 on the Computer Science A AP exam receives immediate credit for CSC-111 Intro to Programming.
- A student who gets a 4 or 5 on the Computer Science Principles AP exam receives immediate credit for CSC-101 Intro to Computer Science.

Student Learning Goals

To give all students who take computer science courses a sense of the nature of computer science and its place in society.

To give computer science majors an understanding and appreciation of the fundamental nature of computer science principles, and to prepare them to become effective users of computer science in their careers.

To give our computer science minors an understanding of computer science, its nature, and uses, to prepare students to become effective users of computer science in their careers.

To give students interested in pursuing graduate study in computer science (or related disciplines) an adequate preparation to succeed in that study.

To prepare students to excel in their academic pursuits beyond computer science. This includes students who take CS distribution courses, computer science minors, and students with additional majors, who will gain deeper insights into their other majors.

Requirements for the Computer Science Major

Code	Title	Credits
Introductory Courses		2
CSC-111	Intro to Programming	
CSC-211	Intro to Data Structures	
Core Courses		3
Take at least 3 of the core courses (the remaining one may be used as an elective)		
CSC-241	Introduction to Machine Organization	
CSC-242	Theory of Programming Languages	
CSC-243	Algorithm Design and Analysis	
CSC-244	Theory of Computing	
Electives		3
Take 3 elective credits (may take the remaining core course as an elective)		
CSC-235	Stochastic Simulation	
CSC-271	Special Topics in Computer Science	
CSC-337	Numerical Analysis	

CSC-338	Topics in Computational Math	
CSC-361	Database System Design	
CSC-362	Operating Systems	
CSC-363	Compiler Design	
CSC-364	Parallel Programming	
Senior Capstone		1
CSC-400	Senior Capstone	
Total Credits		9

Code	Title	Credits
Collateral Requirement		
MAT-108	Intro to Discrete Structures	1
	or MAT-219 Combinatorics	
MAT-111	Calculus I	1

Students may transfer at most one credit from outside Wabash to satisfy the three course elective requirement for the Computer Science major.

Requirements for the Computer Science Minor

The requirements for a minor in computer science are five courses in computer science and a corequisite of one course in mathematics. The computer science courses must include

Code	Title	Credits
Core Requirements		
CSC-111	Intro to Programming	1
CSC-211	Intro to Data Structures	1
One course from the following		1
CSC-241	Introduction to Machine Organization	
CSC-242	Theory of Programming Languages	
CSC-243	Algorithm Design and Analysis	
CSC-244	Theory of Computing	
CSC Electives; except 101 or 106		2
Total Credits		5

Code	Title	Credits
Collateral Requirement		
MAT-108	Intro to Discrete Structures	1
	or MAT-219 Combinatorics	

Potential computer science minors should consult with one of the department members who teaches computer science. Computer Science minors should take CSC-111 Intro to Programming by the fall of the junior year.

CSC-101 Intro to Computer Science

An introduction to the field of computer science: the study of algorithmic processes and the machines that implement them. Students will study the history of computing as well as ethical issues raised by computing and automation. Students will study fundamental areas of the discipline, including basic digital circuits, computer hardware and architecture, data representation, issues of computability, and algorithm design and analysis. Students will also engage in hands-on activities involving basic digital circuits, hardware and programming.

Prerequisites: none

Credit: 1

Distribution: Quantitative Literacy

Equated Courses: APCR

CSC-106 Topics in Introduct Comp Sci

A reflective examination of basic ideas in contemporary computer science. Through participation and discovery, students will consider an articulation of computer science that focuses on procedural units, algorithms, and abstractions. Topics will vary, but could include programming in various contexts, history of computing, etc. This course does not count toward the major or minor in computer science. This course will suffice as a pre-requisite for CSC-111. Topics vary with each scheduled offering. Topics vary with each scheduled offering. Refer to Student Planning's section information for descriptions of individual offerings, and applicability to distribution requirements.

Prerequisites: none

Credits: 0.5-1

Distribution: Quantitative Literacy

CSC-111 Intro to Programming

This course provides an introduction to programming and problem solving in a higher-level, general-purpose language. Programming topics include primitive data types, simple data types such as arrays, program constructs such as conditionals, loops, and functions, and the fundamentals of object-oriented programming. (Note: CSC-111 does not count as a laboratory science.)

Prerequisites: CSC-101, CSC-106, or MAT-112; or permission of the instructor.

Credit: 1

Distribution: Quantitative Literacy

CSC-171 Special Topics in Comp. Sci.

Topics vary with each scheduled offering. Refer to Student Planning's section information for descriptions of individual offerings, and applicability to distribution requirements.

Prerequisites: none

Credit: 1

Distribution: Quantitative Literacy

CSC-187 Independent Study

Individual research projects. The manner of study will be determined by the student in consultation with the instructor. Students must receive written approval of their project proposal from a department Chair before registering for the course.

Prerequisites: none

Credits: 0.5-1

CSC-188 Independent Study

Individual research projects. The manner of study will be determined by the student in consultation with the instructor. Students must receive written approval of their project proposal from a department Chair before registering for the course.

Prerequisites: none

Credits: 0.5-1

CSC-211 Intro to Data Structures

This course studies structured ways of storing and organizing data and algorithms designed for these structures. Attention is given to both theory and practical implementation of data structures and algorithms. Analytical techniques will be developed to study algorithm complexity, comparisons between iterative and recursive algorithms, and theory for searching, sorting, and traversing data. Computational studies will provide practical validation of analytical results and will develop an intuition for understanding tradeoffs between competing methods. Data structures covered include lists, stacks, queues, trees, hash tables, graphs, and related data types.

Prerequisites: CSC-111 with a minimum grade of C-

Credit: 1

CSC-235 Stochastic Simulation

Interesting real world phenomena often involve randomness at some level, and this course develops mathematical and computational tools for studying these systems. In particular, students will study and implement computer simulation models of continuous and discrete stochastic processes with potential applications in physics, economics, epidemiology, networks, sports, elections, and industrial engineering. Specific topics for study include: basic probability models, pseudo-random number generation, queueing models, discrete event simulations, Poisson processes, random walks, Markov chains, Monte Carlo methods, and statistical analysis of simulated data.

Prerequisites: MAT-112 and CSC-111

Credit: 1

CSC-241 Introduction to Machine Organization

This course studies the various levels at which a computer can be studied, both in hardware and software. These levels include transistor level digital circuits, higher-level architectural circuits, and the hierarchy of machine code, assembly code, and high-level programming languages. Students will comparatively study different modern and historical computer architectures, including examples of both RISC and CISC architectures. Students will become proficient in programming in a modern assembly language (e.g. ARM64 or x86-64). This course is offered in the fall semester.

Prerequisites: CSC-111 with a minimum grade of C-

Credit: 1

Distribution: Quantitative Literacy

Equated Courses: CSC-311

CSC-242 Theory of Programming Languages

A study of the paradigms of programming languages, including procedural languages such as Pascal or 'C', object-oriented languages such as C++ or Smalltalk, functional languages such as ML or Scheme, logic-oriented languages such as Prolog, and concurrency such as in Ada. Consideration of how concepts are implemented, such as modules, parameter passing, function evaluation, data types and type checking, memory management, exception handling, and threads. This course is offered in the spring semester.

Prerequisites: CSC-111

Credit: 1

Equated Courses: CSC-321

CSC-243 Algorithm Design and Analysis

This course studies how algorithms are designed, analyzed, implemented and proven to work correctly. Common algorithmic design paradigms will be examined -- divide and conquer, dynamic programming, greedy, as well as the strategy of reducing from one type of problem to another. Standard techniques for studying algorithmic efficiency will be utilized throughout the course, including asymptotic analysis and recurrence relations. Additional specialized topics may be surveyed such as graph algorithms, linear programming, parallel algorithms, approximation algorithms, randomized algorithms, computational geometry and lower bound analysis. This course is offered in the spring semester.

Prerequisites: Take MAT-111 or equivalent., Take CSC-211.

Corequisites: Take either MAT-108 or MAT-219. MAT-108 must be taken previously; MAT-219 can be taken either previously or concurrently.

Credit: 1

Equated Courses: CSC-331

CSC-244 Theory of Computing

How do we know if a problem is computationally hard to solve? In this course, computational problems will be studied as formal languages and classified according to their solvability under various theoretical computation models and resource constraints. The models to be studied will include finite-state automata, pushdown automata, linear-bounded automata and Turing Machines. Alternative characterizations of these models will also be examined, such as those of grammars, circuits, restricted programming languages and Kleene-Godel-Herbrand functions. Complexity classes (e.g., L, P, NP, Co-NP and PSPACE) will be introduced to study time and space constraints, along with the notions of complete problems, efficient reductions and hierarchy theorems. Along the way, many difficult open problems that continue to vex theoretical computer scientists will be explored, such as the infamous P versus NP problem. This course is offered in the fall semester.

Prerequisites: CSC-111 with a minimum grade of C-; CSC-243 with a minimum grade of C-; either MAT-108 or MAT-219 with a minimum grade of C-.

Credit: 1

Equated Courses: CSC-341

CSC-271 Special Topics in Computer Science

Topics vary with each scheduled offering. Refer to Student Planning's section information for descriptions of individual offerings, and applicability to distribution requirements.

Prerequisites: CSC-111 or permission of the instructor

Credits: 0.5-1

CSC-287 Independent Study

Individual research projects. The manner of study will be determined by the student in consultation with the instructor. Students must receive written approval of their project proposal from a department Chair before registering for the course.

Prerequisites: none

Credits: 0.5-1

CSC-288 Independent Study

Individual research projects. The manner of study will be determined by the student in consultation with the instructor. Students must receive written approval of their project proposal from a department Chair before registering for the course.

Prerequisites: none

Credits: 0.5-1

CSC-337 Numerical Analysis

Advanced-This course will address topics such as numerical solution of non-linear equations in one variable, interpolation, approximation, differentiation, integration, difference equations, differential equations and their applications, boundary value problems, linear systems, matrices, and optimization. This course is offered in the fall semester of even-numbered years.

Prerequisites: CSC-111 and MAT-223

Credit: 1

CSC-338 Topics in Computational Math

An advanced course to develop mathematical and computational techniques in areas of mathematics or interdisciplinary study in which computation plays a central and essential role. Topics vary by semester but may include computational geometry, computer algebra, scientific computing, and symbolic computation. Refer to the Course Descriptions document on the Registrar's webpage for topics and descriptions of current offerings. This course is typically offered in the fall semesters of odd-numbered years.

Prerequisites: CSC-111 and MAT-112

Credit: 1

CSC-361 Database System Design

Database management is a central component of a modern computing environment. This course introduces the fundamental concepts of database design and database languages. Topics include relational databases, SQL, formal relational query languages, the E-R model, relational database design, storage and file structures, indexing and hashing, query processing, transactions, and data warehousing and mining.

Prerequisites: Take CSC-211 with a minimum grade of C-

Credit: 1

CSC-362 Operating Systems

At age 21, Linus Torvalds had created his own operating system. Do you want to follow in his footsteps? This course explores the design and implementation of computer operating systems. Topics may include historical aspects of operating systems development, systems programming, process scheduling, synchronization of concurrent processes, virtual machines, memory management and virtual memory, I/O and file systems, system security, OS/architecture interaction, and distributed operating systems. This course will involve working on projects with large amounts of code written in the C programming language.

Prerequisites: CSC-211 with a minimum grade of C-; CSC-241 with a minimum grade of C-

Credit: 1

CSC-363 Compiler Design

This course explores principles and practices used for designing and implementing compilers and interpreters. Students will build a compiler for a programming language designed for the course. The major stages of compilation will be studied in-depth -- lexical analysis, syntax analysis, semantic analysis, and code generation. Additional topics such as advanced parsing techniques and specific compiler-construction tools may be covered at the instructor's discretion.

Prerequisites: Take CSC-211 with a minimum grade of C-

Credit: 1

CSC-364 Parallel Programming

This course provides an introduction to high-performance computing through the study of different ways that a large problem can be divided into separate tasks which are solved simultaneously by parallel processing elements. Topics include the study of different types of parallel computing, the design and implementation of parallel algorithms, hardware that supports parallelism, and analysis of scalability.

Prerequisites: Take CSC-211 with a minimum grade of C-

Credit: 1

Chad Westphal (chair)

CSC-371 Special Topics in Comp. Sci.

Topics vary with each scheduled offering. Refer to Student Planning's section information for descriptions of individual offerings, and applicability to distribution requirements.

Prerequisites: none

Credit: 1

CSC-387 Independent Study

Individual research projects. The manner of study will be determined by the student in consultation with the instructor. Students must receive written approval of their project proposal from a department Chair before registering for the course.

Prerequisites: none

Credits: 0.5-1

CSC-388 Independent Study

Individual research projects. The manner of study will be determined by the student in consultation with the instructor. Students must receive written approval of their project proposal from a department Chair before registering for the course.

Prerequisites: none

Credits: 0.5-1

CSC-400 Senior Capstone

This course is a senior capstone course, which all computer science majors should take in their senior year. This is a project-based course that develops skills in individual and team software development, including reading, documenting, presenting, and critiquing software systems.

Prerequisites: CSC-211 with a minimum grade of C-

Credit: 1

CSC-487 Independent Study

Individual research projects. The manner of study will be determined by the student in consultation with the instructor. Students must receive written approval of their project proposal from a department Chair before registering for the course.

Prerequisites: none

Credits: 0.5-1

CSC-488 Independent Study

Individual research projects. The manner of study will be determined by the student in consultation with the instructor. Students must receive written approval of their project proposal from a department Chair before registering for the course.

Prerequisites: none

Credits: 0.5-1

Computer Science Faculty

Katie Ansaldi

Qixin Deng

Colin B.P. McKinney

William J. Turner