

# COMPUTER SCIENCE (CSC) COURSES

---

Some special topics courses listed below may have individual offerings that will apply to distribution requirements. See the [Curriculum Outline \(http://bulletin.wabash.edu/curriculum/curriculum-outline/\)](http://bulletin.wabash.edu/curriculum/curriculum-outline/) section of this Bulletin for more information.

## 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

## 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. Please refer to the Registrar's page for course description.

**Prerequisites:** none

**Credits:** 0.5-1

## 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-112 Advanced Programming

A variety of topics that are important in developing large-scale software. Object oriented programming in a language such as C++. Dynamic data structures such as lists, queues, and stacks. An introduction to a rigorous analysis of the efficiency of an algorithm. Advanced algorithms such as Quicksort, mergesort, and the use of hash tables. An introduction to using the Unix operating system and Unix tools for software development such as Make.

**Prerequisites:** CSC-111

**Credit:** 1

## CSC-121 Intro to Additional Program Language

An introduction to one or more additional programming languages. Students will build on their previous knowledge of a programming language to learn one or more additional languages. Languages vary by semester but may include any programming paradigm. For a given semester the course content and other particulars will be announced before registration for that semester. This course may be taken multiple times, for credit for each different language.

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

**Credits:** 0.5

## CSC-171 Special Topics in Comp. Sci.

This course is designed for the treatment of material outside the regular offerings of the department. For a given semester, the course content and other particulars will be announced before registration for that semester. Refer to the Course Descriptions document on the Registrar's webpage for topics and descriptions of current offerings.

**Prerequisites:** none

**Credit:** 1

## 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:** MAT-111, CSC-211, and MAT-108 or MAT-219

**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**

This course is designed for the treatment of material outside the regular offerings of the department. For a given semester, the course content and other particulars will be announced before registration for that semester. This course is offered irregularly. Refer to the Course Descriptions document on the Registrar's webpage for Topics and Descriptions of current offerings.

**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 Introduction to 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:** CSC-211 with a minimum grade of C-

**Corequisites:** 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:** CSC-211 with a minimum grade of C-

**Corequisites:** 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:** CSC-211 with a minimum grade of C-

**Corequisites:** Take CSC-211 with a minimum grade of C-

**Credit:** 1

**CSC-371 Special Topics in Comp. Sci.**

This course is designed for the treatment of material outside the regular offerings of the department. For a given semester, the course content and other particulars will be announced before registration for that semester.

Refer to the Course Descriptions document on the Registrar's webpage for topics and descriptions of current offerings.

**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