COMPUTER SCIENCE

Computer science is a branch of science that deals with the theory of computation or the design of computers. It is the study of automating algorithmic processes that scale. It studies the theory, experimentation, and engineering that form the basis for the design and use of computers. The study of computer science involves programming, but it delves beyond just writing more complex or eye-catching programs. A good comprehension of computer science helps the student design better programs and understand how to use a computer to solve new problems.

Goals
The computer science program has the following 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 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 our students interested in continuing to graduate study in computer science or other disciplines that use computational skills an adequate preparation to succeed in that study;
• To prepare students to excel in their majors. This includes students in distribution courses, and mathematics majors and minors.

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.

Requirements for the Computer Science Major

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC-111</td>
<td>Intro to Programming</td>
<td>1</td>
</tr>
<tr>
<td>CSC-211</td>
<td>Intro Data Structures</td>
<td>1</td>
</tr>
</tbody>
</table>

Core Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC-241</td>
<td>Introduction to Machine Organization</td>
<td>1</td>
</tr>
<tr>
<td>CSC-242</td>
<td>Theory of Programming Languages</td>
<td>1</td>
</tr>
<tr>
<td>CSC-243</td>
<td>Algorithm Analysis</td>
<td>1</td>
</tr>
<tr>
<td>CSC-244</td>
<td>Theory of Computing</td>
<td>1</td>
</tr>
</tbody>
</table>

Electives

Two credits from the following: 2

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC-235</td>
<td>Stochastic Simulation</td>
</tr>
<tr>
<td>CSC-271</td>
<td>Special Topics in Computer Science</td>
</tr>
<tr>
<td>CSC-337</td>
<td>Introduction to Numerical Analysis</td>
</tr>
<tr>
<td>CSC-338</td>
<td>Topics in Computational Math</td>
</tr>
<tr>
<td>CSC-361</td>
<td>Database System Design</td>
</tr>
<tr>
<td>CSC-362</td>
<td>Operating Systems</td>
</tr>
<tr>
<td>CSC-363</td>
<td>Compiler Design</td>
</tr>
<tr>
<td>CSC-364</td>
<td>Parallel Programming</td>
</tr>
</tbody>
</table>

Senior Capstone

CSC-400 Senior Capstone 1

Total Credits 9

Collateral Requirement

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAT-108</td>
<td>Intro to Discrete Structures</td>
<td>1</td>
</tr>
<tr>
<td>or MAT-219 Combinatorics</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC-111</td>
<td>Intro to Programming</td>
<td>1</td>
</tr>
<tr>
<td>CSC-121</td>
<td>Intro to Additional Program Language</td>
<td>0.5</td>
</tr>
<tr>
<td>CSC-211</td>
<td>Intro Data Structures</td>
<td>1</td>
</tr>
<tr>
<td>CSC greater than CSC-211</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CSC Electives; except 101</td>
<td>1.5</td>
<td></td>
</tr>
</tbody>
</table>

Total Credits 5

Collateral Requirement

<table>
<thead>
<tr>
<th>Code</th>
<th>Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAT-108</td>
<td>Intro to Discrete Structures</td>
<td>1</td>
</tr>
<tr>
<td>or MAT-219 Combinatorics</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Potential computer science minors should consult with one of the department members who teaches computer science and should read the brochure “How to Minor in Mathematics and Computer Science at Wabash College (https://www.wabash.edu/academics/uploads/math/How_to_Minor_in_Mathematics_or_Computer_Science_at_Wabash_College.pdf).” 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 as the study of algorithmic process. Students will study the history of the field as well as issues currently confronting the computer science community including ethical issues raised by a rapidly changing technology. Students will learn fundamental concepts of computer science such as computer architecture, data representation, and the issues of computability. Students will engage in hands-on algorithm-building activities and some basic programming exercises. Distribution in Natural Science and Mathematics or Quantitative Skills.

Prerequisites: none
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy, Quantitative Skills
Equated Courses: APCR
CSC-111 Intro to Programming
An introduction to programming in a higher-level, general-purpose language (currently Java). Programming topics include primitive data types, simple data types such as arrays, program constructs such as conditionals, loops and procedures, in an object-oriented context. Applications are chosen from areas such as graphics, simulation, and file processing. This course is offered in the fall semester. Distribution in Natural Science and Mathematics or Quantitative literacy. (Note: CSC 111 does not count as a laboratory science.)
Prerequisites: CSC-101 or MAT 112; or permission of the instructor
Credit: 1
Distribution: Natural Science/Mathematics, 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
Distribution: Natural Science/Mathematics, Quantitative Literacy

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 grade of C- or better.
Credits: 0.5
Distribution: Natural Science/Mathematics, Quantitative Literacy

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. This course is offered irregularly.
Prerequisites: none
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy

CSC-187 Independent Study
Enrollment through Instructor and Department Chair approval.
Prerequisites: none
Credits: 0.5-1
Distribution: Natural Science/Mathematics

CSC-188 Independent Study
Enrollment through Instructor and Department Chair approval.
Prerequisites: none
Credits: 0.5-1
Distribution: Natural Science/Mathematics

CSC-211 Intro Data Structures
An introduction to more advanced abstract data types such as lists; sets; trees, including balanced trees; and graphs. Algorithms for traversing, searching, determining connectivity, and so forth. An in-depth study of, and analysis of, the algorithms used to implement these structures. This course is offered in the spring semester.
Prerequisites: Prereq CSC-111 with a grade of C- or better.
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy

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: Prereq of MAT-112 and CSC-111
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy

CSC-241 Introduction to Machine Organization
A study of the various layers at which a machine can be studied, including higher-level languages, assembly language, machine language, and digital circuits. Data representation. A comparison of RISC and CISC architectures. Some programming in a representative assembly language. Issues of cross-language programming. This course is offered irregularly.
Prerequisites: CSC-211
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy

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 irregularly.
Prerequisites: CSC-121
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy

CSC-243 Algorithm Analysis
Advanced topics and problems in analyzing algorithms. Algorithms involving structures such as sequences, sets, and graphs, and topics such as geometric and numeric algorithms. An introduction to the question of P=NP and NP-Complete problems. Parallel algorithms. This course is offered irregularly.
Prerequisites: CSC-211 and MAT-108 or 219
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy

CSC-231 Equated Courses: CSC-311
CSC-244 Theory of Computing  
An introduction to theoretical computer science. Finite state machines and regular expressions. Context-free and push-down automata. Turning machines, effective computability, and the Halting Problem.  
Prerequisites: Take CSC-111 and MAT-108 or MAT-219 with a grade of C- or Better.  
Credit: 1  
Distribution: Natural Science/Mathematics, Quantitative Literacy, Quantitative Skills  
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  
Distribution: Natural Science/Mathematics, Quantitative Literacy  

CSC-287 Independent Study  
Enrollment through Instructor and Department Chair approval.  
Prerequisites: none  
Credits: 0.5-1  
Distribution: Natural Science/Mathematics  

CSC-288 Independent Study  
Enrollment through Instructor and Department Chair approval.  
Prerequisites: none  
Credits: 0.5-1  
Distribution: Natural Science/Mathematics  

CSC-311 Intro Machine Organ  
A study of the various layers at which a machine can be studied, including higher-level languages, assembly language, machine language, and digital circuits. Data representation. A comparison of RISC and CISC architectures. Some programming in a representative assembly language. Issues of cross-language programming. This course is offered irregularly.  
Prerequisites: CSC-211  
Credit: 1  
Distribution: Natural Science/Mathematics, Quantitative Literacy  
Equated Courses: CSC-241  

CSC-321 Programming Lang  
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 irregularly.  
Prerequisites: CSC-121  
Credit: 1  
Distribution: Natural Science/Mathematics, Quantitative Literacy  
Equated Courses: CSC-242  

CSC-331 Analysis Algorithms  
Advanced topics and problems in analyzing algorithms. Algorithms involving structures such as sequences, sets, and graphs, and topics such as geometric and numeric algorithms. An introduction to the question of P=NP and NP-Complete problems. Parallel algorithms. This course is offered irregularly.  
Prerequisites: CSC-211 and MAT-108 or 219  
Credit: 1  
Distribution: Natural Science/Mathematics, Quantitative Literacy  

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  
Distribution: Natural Science/Mathematics, Quantitative Literacy  

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. This course is typically offered in the fall semesters of odd-numbered years.  
Prerequisites: CSC-111 and MAT-112  
Credit: 1  
Distribution: Natural Science/Mathematics, Quantitative Literacy  

CSC-341 Automata, Computability, Formal Language  
An introduction to theoretical computer science. Finite state machines and regular expressions. Context-free languages and push-down automata. Turning machines, effective computability, and the Halting Problem. This course is offered irregularly.  
Prerequisites: CSC-111 and MAT 108 or 219  
Credit: 1  
Distribution: Natural Science/Mathematics, Quantitative Literacy  

CSC-361 Database System Design  
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-  
Corequisites: Take CSC-211 with a minimum grade of C-  
Credit: 1  
Distribution: Natural Science/Mathematics, Quantitative Literacy, Quantitative Skills
CSC-362 Operating Systems
This course explores the design and implementation of computer operating systems. Topics 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.
Prerequisites: Take CSC-211 with a minimum grade of C-
Corequisites: Take CSC-211 with a minimum grade of C-
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy, Quantitative Skills

CSC-363 Compiler Design
This course explores the principles and practices for the design and implementation of compilers and interpreters with the goal of implementing a compiler for a simple object-oriented language. Topics include all stages of the compilation and execution process: lexical analysis, parsing, symbol tables, type systems, scope, semantic analysis, intermediate representations, run-time environments and interpreters, code generation, program analysis and optimization, and garbage collection.
Prerequisites: Take CSC-211 with a minimum grade of C-
Corequisites: Take CSC-211 with a minimum grade of C-
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy, Quantitative Skills

CSC-364 Parallel Programming
This course explores various aspects of parallel computing including parallel architectures, algorithms, systems, programming languages and implementation issues. The focus is on solving real problems on existing parallel machines. Students will participate in significant parallel implementation projects.
Prerequisites: Take CSC-211 with a minimum grade of C-
Corequisites: Take CSC-211 with a minimum grade of C-
Credit: 1
Distribution: Natural Science/Mathematics, Quantitative Literacy, Quantitative Skills

CSC-387 Independent Study
Directed study on special topics for qualified students. May be repeated for credit.
Prerequisites: none
Credits: 0.5-1
Distribution: Natural Science/Mathematics

CSC-388 Independent Study
Directed study on special topics for qualified students. May be repeated for credit.
Prerequisites: none
Credits: 0.5-1
Distribution: Natural Science/Mathematics

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: Take CSC-211 with a grade of C- or better.
Credit: 1

Computer Science (CSC) Faculty
Mark McCartin-Lim Colin B.P. McKinney William J. Turner Chad Westphal (chair)