Department of Computer Science

(www.cs.uni.edu/)

The Department of Computer Science offers the following undergraduate and program certificates. Specific requirements for these programs are listed within this Department of Computer Science section in the following order:

Undergraduate Majors (B.S.)

- Computer Science (p. 1)
- Networking and System Administration (p. 2)

Undergraduate Major (B.A.)

- Computer Science (p. 2)

Minor

- Computer Science (p. 3)
- Data Science (p. 3) (also listed in Department of Mathematics and Department of Physics)

Program Certificates

- Computer Science (p. 4)
- Computer Science Education (p. 4)

Notes:

1. Undergraduate students who have been admitted to the university provisionally because of non-satisfaction of the high school mathematics requirements may not enroll in any computer science credit course before this requirement has been met.
2. All courses counting toward a major or minor in the Department of Computer Science must be passed with a grade of C- or better.
3. Prerequisite courses in the Department of Computer Science must be passed with a grade of C before taking a subsequent course.
4. All majors in the Department of Computer Science require a project course (marked with asterisk in the degree statements). The course used to meet this requirement is to be taken in the area of specialization, i.e., an area in which at least three courses are taken.
5. All courses in a prerequisite chain to a course are considered regressive to it - students may not take them for credit after passing the later course. Additionally, CS 1120, CS 1130, CS 1160, CS ED 1310, and CS ED 1320 are regressive to CS 1520 and any course having it as prerequisite.
6. A student with a major in the Department of Computer Science cannot also receive a Computer Science minor.
7. A student with a major in the Department of Computer Science cannot also receive a Certificate in Computer Science.
8. A student with a minor in the Department of Computer Science cannot also receive a Certificate in Computer Science.

Bachelor of Science Degree Programs

Computer Science Major

The B.S. Computer Science major requires a minimum of 120 total hours to graduate. This total includes UNIFI/General Education requirements and the following specified major requirements, plus electives to complete the minimum of 120 hours.

Required

<table>
<thead>
<tr>
<th>Computer Science:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1410 Computer Organization</td>
<td>3</td>
</tr>
<tr>
<td>CS 1510 Introduction to Computing</td>
<td>4</td>
</tr>
<tr>
<td>CS 1520 Data Structures</td>
<td>4</td>
</tr>
<tr>
<td>CS 1800 Discrete Structures</td>
<td>3</td>
</tr>
<tr>
<td>CS 2530 Intermediate Computing</td>
<td>3</td>
</tr>
<tr>
<td>CS 3730/5730 Project Management</td>
<td>1</td>
</tr>
</tbody>
</table>

Research:

| CS 4800 Undergraduate Research in Computer Science (topic pre-approved by department) | 1 |

Electives

Mathematics:

<table>
<thead>
<tr>
<th>Select four from the following:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 1420 Calculus I ^, #</td>
<td>13</td>
</tr>
<tr>
<td>MATH 1421 Calculus II #</td>
<td></td>
</tr>
<tr>
<td>MATH 2500 Linear Algebra for Applications</td>
<td></td>
</tr>
<tr>
<td>MATH 3440/5440 Numerical Analysis</td>
<td></td>
</tr>
<tr>
<td>MATH 3530/5530 Combinatorics</td>
<td></td>
</tr>
<tr>
<td>MATH 3752/5752 Introduction to Probability</td>
<td></td>
</tr>
<tr>
<td>STAT 1772 Introduction to Statistical Methods</td>
<td></td>
</tr>
</tbody>
</table>

Computer Science:

| 24 |

Eight courses including:

- A specialty of three courses from the Foundations area
- A specialty of three courses from one other area
- One course from each of the remaining two areas
- One of the specialty areas must include a project course (marked with an asterisk *)

Foundations:

| CS 3530 Design and Analysis of Algorithms |   |
| CS 3540 Programming Languages and Paradigms |   |
| CS 3810/5810 Theory of Computation        |   |
| CS 4550/5550 Translation of Programming Languages |   |
| CS 4880/5880 Topics in Computer Science † |   |

Data and Applications:
Department of Computer Science

Required
Mathematics:
MATH 1420 Calculus I ^ 4
MATH 1421 Calculus II 4

Computer Science:
CS 1410 Computer Organization 3
CS 1510 Introduction to Computing 4
CS 1520 Data Structures 4
CS 1800 Discrete Structures 3

Electives
Two courses selected from among the Computer Science "area" courses and 2000-level or above courses meeting the Mathematics requirements.

Total Hours 62

^ MATH 1420 has prerequisite of MATH 1140, or MATH 1110 and MATH 1130, or equivalent.
* A project course must be taken as one of the three in the specialty area.
# MATH 1420, MATH 1421, and CS 4740/5740 are 4-hour courses. CS 3610/5610 is a 4-hour course if taken with lab.
† CS 4880/5880 may be counted in a specialty area with department approval for the specific topic.

Networking and System Administration Major
The B.S. Networking and System Administration major requires a minimum of 120 total hours to graduate. This total includes UNIFI/General Education requirements and the following specified major requirements, plus electives to complete the minimum of 120 hours.

Required
Computer Science:
CS 1410 Computer Organization 3
CS 1510 Introduction to Computing 4
CS 1520 Data Structures 4
CS 1800 Discrete Structures 3
CS 2530 Intermediate Computing 3
CS 3730/5730 Project Management 1

Electives

Bachelor of Arts Degree Programs

Computer Science Major
The B.A. Computer Science major requires a minimum of 120 total hours to graduate. This total includes UNIFI/General Education requirements and the following specified major requirements, plus electives to complete the minimum of 120 hours.

Required
Computer Science:
CS 1410 Computer Organization 3
CS 1510 Introduction to Computing 4
CS 1520 Data Structures 4
CS 1800 Discrete Structures 3
CS 2530 Intermediate Computing 3
CS 3730/5730 Project Management 1
Mathematics:
Select two of the following: 6

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 1420</td>
<td>Calculus I^,#</td>
</tr>
<tr>
<td>MATH 1421</td>
<td>Calculus II#</td>
</tr>
<tr>
<td>MATH 2500</td>
<td>Linear Algebra for Applications</td>
</tr>
<tr>
<td>MATH 3440/5440</td>
<td>Numerical Analysis</td>
</tr>
<tr>
<td>MATH 3530/5530</td>
<td>Combinatorics</td>
</tr>
<tr>
<td>MATH 3752/5752</td>
<td>Introduction to Probability</td>
</tr>
<tr>
<td>STAT 1772</td>
<td>Introduction to Statistical Methods</td>
</tr>
</tbody>
</table>

Computer Science: 18
Six courses including:
- Three courses from one specialty area
- One course from each of the remaining three areas
  - Speciality area must include a project course (*)

Foundations:
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 3530</td>
<td>Design and Analysis of Algorithms</td>
</tr>
<tr>
<td>CS 3540</td>
<td>Programming Languages and Paradigms</td>
</tr>
<tr>
<td>CS 3810/5810</td>
<td>Theory of Computation</td>
</tr>
<tr>
<td>CS 4550/5550</td>
<td>Translation of Programming Languages</td>
</tr>
<tr>
<td>CS 4880/5880</td>
<td>Topics in Computer Science †</td>
</tr>
</tbody>
</table>

Data and Applications:
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 3140/5140</td>
<td>Database Systems</td>
</tr>
<tr>
<td>CS 3150/5150</td>
<td>Information Storage and Retrieval</td>
</tr>
<tr>
<td>CS 3610/5610</td>
<td>Artificial Intelligence#</td>
</tr>
<tr>
<td>CS 3650/5650</td>
<td>Computational Biology</td>
</tr>
<tr>
<td>CS 4620/5620</td>
<td>Intelligent Systems*</td>
</tr>
<tr>
<td>CS 4880/5880</td>
<td>Topics in Computer Science †</td>
</tr>
</tbody>
</table>

Software Engineering:
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 2720</td>
<td>Software Engineering</td>
</tr>
<tr>
<td>CS 3120/5120</td>
<td>User Interface Design</td>
</tr>
<tr>
<td>CS 3750/5750</td>
<td>Software Verification and Validation</td>
</tr>
<tr>
<td>CS 4740/5740</td>
<td>Real-Time Embedded Systems *,#</td>
</tr>
</tbody>
</table>

Electives 3

Physics:
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHYSICS 4160/5160</td>
<td>Data Visualization, Modeling and Simulation</td>
</tr>
</tbody>
</table>

One course selected from among the Computer Science “area” courses and 2000-level or above courses meeting the Mathematics requirement.

Total Hours 45

^ MATH 1420 has prerequisite of MATH 1140, or MATH 1110 and MATH 1130, or equivalent.
* A project course must be taken as one of the three in the specialty area.
# MATH 1420, MATH 1421, and CS 4740/5740 are 4-hour courses. CS 3610/5610 is a 4-hour course if taken with lab.
† CS 4880 may be counted in a specialty area with department approval for the specific topic.

Minors

Computer Science Minor
A student with a major in the Department of Computer Science cannot also receive a Computer Science minor.

Required
Computer Science:
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1410</td>
<td>Computer Organization</td>
</tr>
<tr>
<td>CS 1510</td>
<td>Introduction to Computing</td>
</tr>
<tr>
<td>CS 1520</td>
<td>Data Structures</td>
</tr>
<tr>
<td>CS 1800</td>
<td>Discrete Structures</td>
</tr>
<tr>
<td>CS 2530</td>
<td>Intermediate Computing</td>
</tr>
</tbody>
</table>

Electives
any Computer Science course that counts toward the Computer Science B.A. major

Total Hours 26

Data Science Minor
The Data Science minor is an interdisciplinary program that integrates computer programming, machine learning, statistics, predictive modeling and visualization to provide students with broad based skills for extracting gainful information from data that originate from a variety of sources. A final project (ideally with corporate or non-profit partnerships) will ensure that students employ their skills to solve a real-world problem.

Statistics:
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAT 1772</td>
<td>Introduction to Statistical Methods</td>
</tr>
<tr>
<td>STAT 4784/5784</td>
<td>Introduction to Machine Learning</td>
</tr>
</tbody>
</table>

Computer Science:
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1510</td>
<td>Introduction to Computing</td>
</tr>
<tr>
<td>CS 2150</td>
<td>Computing for Data Science</td>
</tr>
</tbody>
</table>

or
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1520</td>
<td>Data Structures</td>
</tr>
<tr>
<td>CS 1800</td>
<td>Discrete Structures</td>
</tr>
</tbody>
</table>

Electives
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 3140/5140</td>
<td>Database Systems</td>
</tr>
</tbody>
</table>

Physics:
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHYSICS 4160/5160</td>
<td>Data Visualization, Modeling and Simulation</td>
</tr>
</tbody>
</table>

Required Data Science Project 2-3
Program Certificates

The University of Northern Iowa makes available, in addition to traditional programs, the opportunity for students to earn program certificates. Program certificates provide an alternative to programs leading to a degree, a major, or a minor; they certify that an individual has completed a program approved by the university. For information on the following certificates, contact the Department of Computer Science or the Office of the Registrar, which serves as the centralized registry.

Certificate in Computer Science

A student with a major in the Department of Computer Science cannot also receive a Certificate in Computer Science.

Required

<table>
<thead>
<tr>
<th>Computer Science:</th>
<th>Data Structures</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1520</td>
<td>Media Computation</td>
<td></td>
</tr>
<tr>
<td>one course from the following:</td>
<td>3-4</td>
<td></td>
</tr>
<tr>
<td>CS 1120</td>
<td>Visual BASIC Programming</td>
<td></td>
</tr>
<tr>
<td>CS 1130</td>
<td>C/C++ Programming</td>
<td></td>
</tr>
<tr>
<td>CS ED 1310</td>
<td>Programming Environments for Elementary Education</td>
<td></td>
</tr>
<tr>
<td>CS ED 1320</td>
<td>Fundamentals of Programming</td>
<td></td>
</tr>
<tr>
<td>CS 1510</td>
<td>Introduction to Computing</td>
<td></td>
</tr>
</tbody>
</table>

Two courses, from ONE of the following groups: 6

Group 1:

<table>
<thead>
<tr>
<th>CS 1800</th>
<th>Discrete Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 2530</td>
<td>Intermediate Computing</td>
</tr>
</tbody>
</table>

or Group 2:

<table>
<thead>
<tr>
<th>CS 1410</th>
<th>Computer Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 2420</td>
<td>Computer Architecture and Parallel Programming</td>
</tr>
</tbody>
</table>

Total Hours 13-14

Certificate in Computer Science Education

This Computer Science Education certificate is appropriate for students interested in adding experience in computer science to a teaching license. It is for students with a teaching major in a discipline outside of computer science. It consists of the coursework approved by the Iowa Board of Educational Examiners to qualify for the addition of the state's 5-12 Computer Science endorsement #278 on a state teaching license.

Required:

| CS ED 1320 | Fundamentals of Programming | 3 |
| CS ED 2310 | Foundational Concepts in Computer Science | 3 |
| CS ED 3310/5310 | Teaching and Learning Programming | 3 |

Total Hours 15

Computer Science, B.S.

Goal 1: Students will be able to effectively communicate computing information to colleagues and the public.

Outcome 1: Students will be able to prepare and produce written communications using standard computing style and format.

Outcome 2: Students will be able to prepare and deliver an oral presentation on computing topics.

Goal 2: Students will be able to apply computing knowledge to problems involving data and process.

Outcome 3: Students will demonstrate proficient knowledge and application of computing content.

Goal 3: Students will be able to think critically about computing problems.

Outcome 4: Students will be able to specify a computing module’s interface and design its implementation.

Outcome 5: Students will be conduct a research or development project in which they specify a computing problem, investigate possible solutions, and implement a working system.

Goal 4: Students will demonstrate the skills needed to work on a team successfully.

Outcome 6: Students will work on a team to analyze a computing problem and implement its solution.

Networking and System Administration, B.S.

Goal 1: Students will be able to effectively communicate networking and computer system information to colleagues and the public.

Outcome 1: Students will be able to prepare and produce written communications using standard computing style and format.

Outcome 2: Students will be able to prepare and deliver an oral presentation on networking and computer system topics.

Goal 2: Students will be able to apply networking and computer system knowledge to problems involving data and process.

Outcome 3: Students will demonstrate proficient knowledge and application of networking and computer system content.

Goal 3: Students will be able to think critically about networking and computer system problems.

Outcome 4: Students will be able to specify a computing systems’s interface and design its implementation.

Outcome 5: Students will be conduct a research or development project in which they specify a networking or computer system problem, investigate possible solutions, and implement a working system.
Goal 4: Students will demonstrate the skills needed to work on a team successfully.

Outcome 6: Students will work on a team to analyze a networking or computer system problem and implement its solution.

Computer Science, B.A.

Goal 1: Students will be able to effectively communicate computing information to colleagues and the public.

Outcome 1: Students will be able to prepare and produce written communications using standard computing style and format.

Outcome 2: Students will be able to prepare and deliver an oral presentation on computing topics.

Goal 2: Students will be able to apply computing knowledge to problems involving data and process.

Outcome 3: Students will demonstrate proficient knowledge and application of computing content.

Goal 3: Students will be able to think critically about computing problems.

Outcome 4: Students will be able to specify a computing module’s interface and design its implementation.

Goal 4: Students will demonstrate the skills needed to work on a team successfully.

Outcome 5: Students will work on a team to analyze a computing problem and implement its solution.

Computer Science Education Courses

CS ED 1310. Programming Environments for Elementary Education — 3 hrs.
Introduction to computational thinking and computer programming. Taught as a survey of programming environments used by elementary education teachers. Topics include structure of programming and the study of several programming environments used by students at a variety of age/ability levels. Prerequisite(s): TEACHING 2017. (Fall and Spring)

Introduction to computer programming through a survey of programming environments used by teachers. Topics include structure of programming, study of several programming environments used by students at a variety of age/ability levels, and end-user programming for teachers. (Fall)

CS ED 2310. Foundational Concepts in Computer Science — 3 hrs.
Examine the breadth of the field of computer science beyond the programming of the computers. Topics include data storage and manipulation, computer organization, operating systems, networking and the Internet, software engineering, databases, and artificial intelligence. Give potential teachers a general introduction to commonly taught sub-disciplines in computer science so that they can have informed discussions with students before they study a topic more deeply. Prerequisite(s): TEACHING 2017. (Fall)

CS ED 3310/5310. Teaching and Learning Programming — 3 hrs.
Students will enhance their skill and understanding of programming while examining common programming problems, examples of bad/good code, and common misconceptions/difficulties. Students will examine common paradigms for teaching programming. Students will study broader challenges to learning such as broadening participation, understanding and avoiding bias, and best practices for classroom engagement. Prerequisite(s): CS ED 1320 or equivalent; TEACHING 2017; junior standing. (Spring)

CS ED 3320/5320. Data Structures and Algorithms — 3 hrs.
Introduction to the structure and application of common data structures used in computer science and the algorithms used with/for these structures. Includes an ongoing discussion on algorithm analysis. Also includes significant elements of algorithms, program design, techniques for data storage and retrieval, and data beyond a local text file. Prerequisite(s): CS ED 1320 or equivalent; CS ED 3310/5310; TEACHING 2017; junior standing. (Spring)

CS ED 4330/5330. Methods for Teaching Computer Science — 3 hrs.
Teaching approaches, instructional and assessment strategies, curricular and laboratory materials, and issues in secondary computer science. Students will develop a project, either in a classroom or during an after-school or similar experience that seeks to teach a section of the CS standards. Projects should stress inclusivity and appeal to a diverse group of students. Data will be collected about projects including demographics of participants, aggregate achievement, and analysis of student work samples. Prerequisite(s): CS ED 1320 or equivalent; CS ED 2310; CS ED 3310/5310; CS ED 3320/5320; TEACHING 3128; EDPSYCH 3148; junior standing. (Spring)

Computer Science Courses

Introduction to operation, applications, implications of computers, microcomputers, and network communications. Develops skill in current applications and sensitizes students to societal issues related to computing. (Fall and Spring)

Explores use of computational tools to explore data sets, find patterns, and solve complex problems. Topics include representing problems, modeling data, simulating processes, and validating models, with applications in the sciences, social sciences, humanities, and business. (Fall and Spring)

CS 1100. Web Development: Client-Side Coding — 3 hrs.
Client-side Web development adhering to recent/current Web standards. Includes by-hand Web page development involving basic HTML, CSS, data acquisition using forms, and JavaScript for data validation and simple Web-based tools. (Fall)

CS 1120. Media Computation — 3 hrs.
Introduction to computation, algorithmic thinking, data transformation and processing, and programming in the context of media such as images, sound, and video. (Spring)

Programming using the language Visual BASIC. Broad coverage of language syntax, programming practice, and programming problems appropriate to the novice or end-use programmer using a personal computer. (Fall and Spring)

CS 1160. C/C++ Programming — 3 hrs.
Programming using the C and C++ languages including the object-oriented paradigm. Broad coverage of language syntax and programming practice. Appropriate for developers of general computing applications and systems. Course presumes no prior programming experience. (Fall)
CS 1170. Introductory Programming for Data Science — 3 hrs.
Explores use of software development and tools to explore data sets, find patterns, and solve complex problems in the area of data science. Topics include introductory programming methodologies, data visualization, and an introduction to data mining. (Fall)

Study of computers in terms of their major functional units. Machine representations of data, digital logic, memory, CPUs, buses, and input/output. Instruction set architectures and their implementations, addressing methods, and sequencing. Assembly language programming. Prerequisite(s) or corequisite(s): CS 1510. (Fall and Spring)

CS 1510. Introduction to Computing — 4 hrs.
Introduction to software development through algorithmic problem solving and procedural abstraction. Programming in the small. Fundamental control structures, data modeling, and file processing. Significant emphasis on program design and style. (Fall and Spring)

CS 1520. Data Structures — 4 hrs.
Introduction to use and implementation of data structures such as sets, hash tables, stacks, trees, queues, heaps, and graphs. Additional topics include searching algorithms, sorting algorithms, and algorithmic time and space complexity analysis. Design and implementation of programs using functional decomposition. Prerequisite(s): CS 1510 or department approval. Prerequisite(s) or corequisite(s): CS 1800 or department approval. (Fall and Spring)

CS 1800. Discrete Structures — 3 hrs.
Introduction to logical forms, arguments, predicates, and quantified statements; methods of proof; elementary number theory; counting; sequences; sets; functions; relations; graphs; and Boolean algebra in the context of computer science. Prerequisite(s): CS 1130, CS 1160, or CS 1510. (Fall and Spring)

Development of interactive web sites: server-side scripting; database definition and use; site security; and additional topics as time allows. Prerequisite(s): CS 1100 or consent of instructor. (Fall and Spring)

CS 2150. Computing for Data Science — 3 hrs.
Introduction to basic data and file structures for data science, such as sets, lists, matrices, and maps, and the algorithms that use them. Boolean algebra and its role in reasoning about data. Basics of the software development life cycle with emphasis on requirements and testing. Not available for credit to students with credit in CS 1800 or CS 1520. Prerequisite(s): CS 1510; consent of department. (Spring)

Computer architecture of uniprocessor and multiprocessor systems with an emphasis on parallel programming to achieve high performance. Topics include processor design (pipelining and superscalar), memory hierarchy, interconnection networks, performance metrics, parallel program design, and parallel programming tools for multi-core systems, clusters, and graphics processing units. Prerequisite(s): CS 1410. (Fall)

Intermediate software development in an object-oriented environment. Further experience with object-oriented analysis and design, including modeling languages. Focus on software reuse through frameworks and patterns and on software development methodology and tools. Prerequisite(s): CS 1510; CS 1520; CS 1800; or department approval. (Fall and Spring)

CS 2720. Software Engineering — 3 hrs.
Study of software life cycle models and their phases—planning, requirements, specifications, design, implementation, testing, and maintenance. Emphasis on tools, documentation, and applications. Prerequisite(s): CS 1520; CS 1800. (Spring)

CS 2880. Topics in Computing — 3 hrs.
Topics from recent developments in computing appropriate to students with limited computer science background, often relating to relatively sophisticated or technical use of computing or recent developments in programming. May be repeated on different topic. (Variable)

CS 3110. Web Application Development — 3 hrs.
Development of web applications such as content management systems. Involves HTML and CSS coding, client- and server-side scripting, the creation and manipulation of databases to generate web pages, and secure authentication of users. Prerequisite(s): CS 1520; CS 1100 or COMM 2555. (Variable)

CS 3120/5120. User Interface Design — 3 hrs.
Examination of the theory, design, programming, and evaluation of interactive application interfaces. Topics include human capabilities and limitations, the interface design and engineering process, prototyping and interface construction, interface evaluation, and possibly topics such as data visualization and the World Wide Web. Prerequisite(s): for Computer Science majors: CS 1520; CS 1800; junior standing. Prerequisite(s) for non-Computer Science majors: junior standing; department approval. (Fall)

Study of, and access to, physical databases; data models, query languages, transaction processing, and recovery techniques; object-oriented and distributed database systems; and database design. Prerequisite(s): CS 1520; CS 1800; junior standing. Prerequisite(s) for Data Science minors: CS 2150; junior standing. (Fall)

CS 3150/5150. Information Storage and Retrieval — 3 hrs.
Natural language processing; analysis of textual material by statistical, syntactic, and logical methods; retrieval systems models, dictionary construction, query processing, file structures, content analysis; automatic retrieval systems and question-answering systems; and evaluation of retrieval effectiveness. Prerequisite(s): CS 1520; CS 1800; junior standing. (Spring)

CS 3179. Cooperative Education — 1-4 hrs.
Application of classroom learning to field experience. Credit may not be applied to major or minor. Offered on credit/no credit basis only. May be retaken for up to six credits total. (Fall and Spring)

History and evolution of operating systems; process and processor management; primary and auxiliary storage management; performance evaluation, security, and distributed systems issues; and case studies of modern operating systems. Prerequisite(s): CS 1410; CS 1520; CS 1800; junior standing. (Spring)

Network architectures and communication protocol standards. Topics include communication of digital data, data-link protocols, local-area networks, network-layer protocols, transport-layer protocols, applications, network security, and management. Prerequisite(s): CS 1410; CS 1520; CS 1800; junior standing. Prerequisites for Department of Technology majors: TECH 1037; TECH 2041; TECH 2042; TECH 1160. (Fall)
CS 3510. Topics in Programming — 1 hr.
Quick study of a specified programming language or environment for those with considerable programming experience. Language syntax and semantics, common problems solved using it, and best practices. May be repeated with different topics. Prerequisite(s): CS 1520; CS 1800. (Variable)

CS 3530. Design and Analysis of Algorithms — 3 hrs.
Algorithm design techniques such as dynamic programming and greedy algorithms; complexity analysis of algorithms; efficient algorithms for classical problems; intractable problems and techniques for addressing them; and algorithms for parallel machines. Prerequisite(s): CS 1520; CS 1800. (Fall)

CS 3540. Programming Languages and Paradigms — 3 hrs.
Organization of programming languages; language design issues including syntax, data types, sequence control, and storage management; comparison of language features from object-oriented, imperative, functional, and logical paradigms. Prerequisite(s): CS 1520; CS 1800; junior standing. (Spring)

CS 3550/5550. Introduction to Analysis of Programming Languages — 3 hrs.
Introduction to analysis of programming languages and construction of translators. Prerequisite(s): CS 2530 and one of the following: CS 3500, CS 3540, CS 3810/5810; junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730. (Fall)

CS 3530. Design and Analysis of Algorithms — 3 hrs.
Algorithm design techniques such as dynamic programming and greedy algorithms; complexity analysis of algorithms; efficient algorithms for classical problems; intractable problems and techniques for addressing them; and algorithms for parallel machines. Prerequisite(s): CS 1520; CS 1800. (Variable)

CS 3540. Programming Languages and Paradigms — 3 hrs.
Organization of programming languages; language design issues including syntax, data types, sequence control, and storage management; comparison of language features from object-oriented, imperative, functional, and logical paradigms. Prerequisite(s): CS 1520; CS 1800; junior standing. (Spring)

CS 3550/5550. Introduction to Analysis of Programming Languages — 3 hrs.
Introduction to analysis of programming languages and construction of translators. Prerequisite(s): CS 2530 and one of the following: CS 3500, CS 3540, CS 3810/5810; junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730. (Fall)

CS 3610/5610. Artificial Intelligence — 3 hrs.
Models of intelligent behavior and problem solving; knowledge representation and search methods; learning; topics such as knowledge-based systems, language understanding, and vision. Prerequisite(s): CS 1520; CS 1800; junior standing. (Spring)

CS 3650/5650. Computational Biology — 3 hrs.
Applications of computer science techniques to biological problems; introduction to computational DNA and protein sequence analysis; dynamic programming; optimal alignment algorithms; DNA sequencing and fragment assembly; gene expression arrays; clustering algorithms; evolutionary trees; multiple alignments; hidden Markov models. Prerequisite(s): CS 1520 and CS 1800; junior standing. (Spring)

CS 3730/5730. Project Management — 1 hr.
Examination of problems of organizing, controlling, managing, and evaluating a software project; software metrics and human input. Prerequisite(s): For Networking and System Administration majors: junior standing. For all other majors: CS 2530; junior standing. (Fall and Spring)

CS 3750/5750. Software Verification and Validation — 3 hrs.
Taxonomy of software defects and approaches to identifying them, including black box testing, functional testing, control flow testing, data flow testing, software inspection. Introduction to use of static analysis in development of high integrity software. Prerequisite(s): CS 1520; CS 1800; junior standing. (Spring)

Topics include regular languages and grammars; finite state automata; context-free languages and grammars; language recognition and parsing; and Turing computability and undecidability. Prerequisite(s): CS 1800; junior standing. (Spring)

CS 4400/5400. System Administration — 3 hrs.
Major concepts and mechanisms associated with computer system administration. Focus on issues surrounding user management, the configuration of services, and the coordination of distributed resources. Prerequisite(s): CS 3470/5470; junior standing. (Spring)

Topics include the need for security services, data integrity, network intrusion and monitoring, configuration of secure services, root kits, and buffer overflow techniques and remedies. Additional topics include enterprise-wide monitoring, honeypots, and recognizing trends in a networked environment. Prerequisite(s): CS 3470/5470; junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730. (Spring)

Essentials of applied computer system forensics; evidence preserving techniques; reconstruction of storage volumes and deleted volumes and files; identification of evidence in computer memory; encryption; log analysis; generation of forensic reports. Topics are covered in the context of incident response to compromised systems. Prerequisite(s): CS 1410; CS 1520; CS 1800; CS 3430/5430. (Fall)

CS 4550/5550. Translation of Programming Languages — 3 hrs.
Introduction to analysis of programming languages and construction of translators. Prerequisite(s): CS 2530 and one of the following: CS 3530, CS 3540, CS 3810/5810; junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730. (Fall)

CS 4620/5620. Intelligent Systems — 3 hrs.
Design and implementation of programs that apply artificial intelligence techniques to problems such as design, diagnosis, and distributed problem solving. Emphasis on team design and development of large systems. Prerequisite(s): CS 2530; CS 3610/5610 or equivalent; junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730. (Spring)

Specification, design, and implementation principles and techniques for real-time embedded systems. Topics include programming languages and paradigms, reliability and fault tolerance, concurrent programming, scheduling, and the interaction between hardware and software. Student teams will complete a significant real-time embedded software project. Prerequisite(s): CS 2530; CS 2720; junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730. (Fall)

CS 4800. Undergraduate Research in Computer Science — 1-3 hrs.
May be repeated for maximum of 6 hours. (Fall and Spring)

CS 4880/5880. Topics in Computer Science — 3 hrs.
Topics of general interest from any area of computer science, including systems, software, or theory. Can be counted in any specialization area, with department approval for individual topics. Prerequisite(s): CS 1520; CS 1800; junior standing. (Variable)

CS 6140. Database Management Systems — 3 hrs.
Database system concepts, physical data organization, the network model and the DBTG Proposal, the hierarchical model, the relational model, relational query languages, design theory of relational databases, query optimization, and normalization. Prerequisite(s): consent of instructor. (Variable)

Survey of issues in computer system analysis and design. Emphasis on relationship between system hardware and software including tools and environments for software development on parallel and distributed computer systems. Prerequisite(s): for Computer Science majors: CS 2420 or CS 3430/5430. Prerequisite(s) for non-Computer Science majors: proficiency in a high-level programming language; consent of instructor. (Variable)

CS 6500. Algorithms — 3 hrs.
Survey of analysis and design of algorithms. Topics include algorithm design techniques, efficient algorithms for classical problems, and intractable problems and techniques for solving them. Prerequisite(s): CS 3530. (Variable)
Department of Computer Science

Survey of theoretical models of computation and basic theory of computability. Topics include Turing machines, undecidability, NP-completeness, and computational complexity. Consideration of applications of theory to contemporary problems in computing. Prerequisite(s): CS 3810/5810. (Variable)