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
- Networking and System Administration

**Undergraduate Major (B.A.)**
- Computer Science

**Minor**
- Computer Science

**Program Certificates**
- Computer Applications
- Computer Science

**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 1140, CS 1150, and CS 1160 are regressive to CS 1520 and any course having it as prerequisite.

### Bachelor of Science Degree Programs

#### Computer Science Major

The B.S. Computer Science major requires a minimum of 126 total hours to graduate. This total includes Liberal Arts Core requirements and the following specified major requirements, plus electives to complete the minimum of 126 hours.

#### Required

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

**Research:**
- CS 4800 (810:180) Undergraduate Research in Computer Science (topic pre-approved by department) 1

#### Electives

**Mathematics:**

Select four from the following:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 1420</td>
<td>Calculus I ^,#</td>
<td>4</td>
</tr>
<tr>
<td>MATH 1421</td>
<td>Calculus II #</td>
<td>4</td>
</tr>
<tr>
<td>MATH 2500</td>
<td>Linear Algebra for Applications</td>
<td>3</td>
</tr>
<tr>
<td>MATH 3440</td>
<td>Numerical Analysis</td>
<td>3</td>
</tr>
</tbody>
</table>

**Computer Science:**

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:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 3530</td>
<td>Design and Analysis of Algorithms</td>
<td>3</td>
</tr>
<tr>
<td>CS 3540</td>
<td>Programming Languages and Paradigms</td>
<td>3</td>
</tr>
<tr>
<td>CS 3810</td>
<td>Theory of Computation</td>
<td>3</td>
</tr>
<tr>
<td>CS 4550</td>
<td>Translation of Programming Languages ^</td>
<td>3</td>
</tr>
<tr>
<td>CS 4880</td>
<td>Topics in Computer Science †</td>
<td>3</td>
</tr>
</tbody>
</table>

**Data and Applications:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 3140</td>
<td>Database Systems</td>
<td>3</td>
</tr>
<tr>
<td>CS 3150</td>
<td>Information Storage and Retrieval</td>
<td>3</td>
</tr>
<tr>
<td>CS 3610</td>
<td>Artificial Intelligence #</td>
<td>3</td>
</tr>
<tr>
<td>CS 3650</td>
<td>Computational Biology</td>
<td>3</td>
</tr>
<tr>
<td>CS 4620</td>
<td>Intelligent Systems *</td>
<td>3</td>
</tr>
</tbody>
</table>
Department of Computer Science

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Department</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 4880/5880 (810:188g)</td>
<td>Topics in Computer Science †</td>
<td>CS 4880/5880 (810:188g)</td>
<td>3</td>
</tr>
<tr>
<td>CS 2720 (810:172)</td>
<td>Software Engineering</td>
<td>CS 2720</td>
<td>3</td>
</tr>
<tr>
<td>CS 3120/5120 (810:112g)</td>
<td>User Interface Design</td>
<td>CS 3120/5120 (810:112g)</td>
<td>4</td>
</tr>
<tr>
<td>CS 3750/5750 (810:175g)</td>
<td>Software Verification and Validation</td>
<td>CS 3750/5750 (810:175g)</td>
<td>3</td>
</tr>
<tr>
<td>CS 4740/5740 (810:174g)</td>
<td>Real-Time Embedded Systems †</td>
<td>CS 4740/5740 (810:174g)</td>
<td>3</td>
</tr>
<tr>
<td>CS 4880/5880 (810:188g)</td>
<td>Topics in Computer Science †</td>
<td>CS 4880/5880 (810:188g)</td>
<td>3</td>
</tr>
</tbody>
</table>

**Software Engineering:**
- CS 2720 (810:172) Software Engineering
- CS 3120/5120 (810:112g) User Interface Design
- CS 3750/5750 (810:175g) Software Verification and Validation
- CS 4740/5740 (810:174g) Real-Time Embedded Systems †
- CS 4880/5880 (810:188g) Topics in Computer Science †

**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 (800:060) has prerequisite of MATH 1140 (800:046), or MATH 1110 (800:043) and MATH 1130 (800:044), or equivalent.
- A project course must be taken as one of the three in the specialty area.
- MATH 1420 (800:060), MATH 1421 (800:061), and CS 4740/5740 (810:174g) are 4-hour courses. CS 3610/5610 (810:161g) is a 4-hour course if taken with lab.
- CS 4880/5880 (810:188g) 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 126 total hours to graduate. This total includes Liberal Arts Core requirements and the following specified major requirements, plus electives to complete the minimum of 126 hours.

**Required**

**Mathematics:**
- MATH 1420 (800:060) Calculus I ^
- MATH 1421 (800:061) Calculus II

**Computer Science:**
- CS 1410 (810:041) Computer Organization
- CS 1510 (810:051) Introduction to Computing
- CS 1520 (810:052) Data Structures
- CS 1800 (810:080) Discrete Structures
- CS 3430/5430 (810:143g) Operating Systems
- CS 3470/5470 (810:147g) Networking
- CS 3730/5730 (810:173g) Project Management

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

Total Hours 57

- Has prerequisite of satisfactory score on ALEKS exam or subsequent remediation.

**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 Liberal Arts Core requirements and the following specified major requirements, plus electives to complete the minimum of 120 hours.

**Required**

**Computer Science:**
- CS 1410 (810:041) Computer Organization
- CS 1510 (810:051) Introduction to Computing
- CS 1520 (810:052) Data Structures
- CS 1800 (810:080) Discrete Structures
- CS 2530 (810:053) Intermediate Computing
- CS 3730/5730 (810:173g) Project Management

**Electives**

**Mathematics:**
- Select two of the following:
- MATH 1420 (800:060) Calculus I ^, #
MATH 1421 (800:061)  Calculus II #
MATH 2500 (800:076)  Linear Algebra for Applications
MATH 3440/5440 (800:176g)  Numerical Analysis
MATH 3530/5530 (800:143g)  Combinatorics
MATH 3752/5752 (800:152g)  Introduction to Probability
STAT 1772 (800:072)  Introduction to Statistical Methods

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

Foundations:
CS 3530 (810:153)  Design and Analysis of Algorithms
CS 3540 (810:154)  Programming Languages and Paradigms
CS 3810/5810 (810:181g)  Theory of Computation
CS 4550/5550 (810:155g)  Translation of Programming Languages *
CS 4880/5880 (810:188g)  Topics in Computer Science †

Data and Applications:
CS 3140/5140 (810:114g)  Database Systems
CS 3150/5150 (810:115g)  Information Storage and Retrieval
CS 3610/5610 (810:161g)  Artificial Intelligence #
CS 3650/5650 (810:166g)  Computational Biology
CS 4620/5620 (810:162g)  Intelligent Systems *
CS 4880/5880 (810:188g)  Topics in Computer Science †

Software Engineering:
CS 2720 (810:172)  Software Engineering
CS 3120/5120 (810:112g)  User Interface Design
CS 3750/5750 (810:175g)  Software Verification and Validation
CS 4740/5740 (810:174g)  Real-Time Embedded Systems *,#
CS 4880/5880 (810:188g)  Topics in Computer Science †

Systems:
CS 2420 (810:142)  Computer Architecture and Parallel Programming
CS 3430/5430 (810:143g)  Operating Systems
CS 3470/5470 (810:147g)  Networking
CS 4400/5400 (810:140g)  System Administration
CS 4410/5410 (810:141g)  System Security *
CS 4420  Applied Systems Forensics *
CS 4880/5880 (810:188g)  Topics in Computer Science †

Electives  3

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 (800:060) has prerequisite of MATH 1140 (800:046), or MATH 1110 (800:043) and MATH 1130 (800:044), or equivalent.
* A project course must be taken as one of the three in the specialty area.
#  MATH 1420 (800:060), MATH 1421 (800:061), and CS 4740/5740 (810:174g) are 4-hour courses. CS 3610/5610 (810:161g) 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 Computer Science major cannot also receive a Computer Science minor.

Required
Computer Science:
CS 1410 (810:041)  Computer Organization  3
CS 1510 (810:051)  Introduction to Computing  4
CS 1520 (810:052)  Data Structures  4
CS 1800 (810:080)  Discrete Structures  3
CS 2530 (810:053)  Intermediate Computing  3

Electives
any Computer Science course that counts toward the Computer Science B.A. major  9

Total Hours  26

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 Applications

Required
Computer Science:
CS 1000 (810:021)  Computing Skills and Concepts  3
CS 1010 (810:022)  Microcomputer Applications and Systems Integration  3
CS 1020 (810:023)  Microcomputer Systems  3

Electives
Computer Science:
CS 1130 (810:030)  Visual BASIC Programming  3
CS 2880 (810:088)  Topics in Computing  3
Certificate in Computer Science

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

Required

Computer Science:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1120 (810:056)</td>
<td>Media Computation</td>
</tr>
<tr>
<td>CS 1130 (810:030)</td>
<td>Visual BASIC Programming</td>
</tr>
<tr>
<td>CS 1140</td>
<td>Programming Environments for Secondary Education</td>
</tr>
<tr>
<td>CS 1150</td>
<td>Programming Environments for Elementary Education</td>
</tr>
<tr>
<td>CS 1160 (810:036)</td>
<td>C/C++ Programming</td>
</tr>
<tr>
<td>CS 1510 (810:051)</td>
<td>Introduction to Computing</td>
</tr>
<tr>
<td>CS 1520 (810:052)</td>
<td>Data Structures</td>
</tr>
</tbody>
</table>

Two courses, from ONE of the following groups:

Group 1:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1800 (810:080)</td>
<td>Discrete Structures</td>
</tr>
<tr>
<td>CS 2530 (810:053)</td>
<td>Intermediate Computing</td>
</tr>
</tbody>
</table>

or Group 2:

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1410 (810:041)</td>
<td>Computer Organization</td>
</tr>
<tr>
<td>CS 2420 (810:142)</td>
<td>Computer Architecture and Parallel Programming</td>
</tr>
</tbody>
</table>

Total Hours: 12

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)

CS 1010 (810:022). Microcomputer Applications and Systems Integration — 3 hrs.
Emphasis on developing macros and programs, importing and exporting files between applications, solving and implementing applications and problems, and other more advanced topics. Examination of software such as word processing, spreadsheets, and database languages. Prerequisite(s): CS 1000 (810:021) or equivalent. (Fall)

Functional description of microcomputer operating systems; commands and utilities, system and file organization, memory and file management, troubleshooting strategies, and networks. Prerequisite(s): CS 1010 (810:022) or any one-semester programming course. (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 (810:017). 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)

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 1140. Programming Environments for Secondary Education — 3 hrs.
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 1150. 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. Lecture/discussion, 1 period; lab, 2 periods. Prerequisite(s): TEACHING 2017. (Fall and Spring)

CS 1160 (810:036), 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)

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 (810:051). (Fall and Spring)

CS 1510 (810:051). 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 (810:052). Data Structures — 4 hrs.
Introduction to use and implementation of data and file structures such as sets, hash tables, stacks, trees, queues, heaps, and graphs. Basic algorithm analysis. Searching and sorting. Basic object-oriented analysis, design, and modeling tools. Prerequisite(s): CS 1510 (810:051) or department approval. Prerequisite(s) or corequisite(s): CS 1800 (810:080) or department approval. (Fall and Spring)

CS 1800 (810:080). 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 (810:030),
CS 1160 (810:036), or CS 1510 (810:051). (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 (810:017) or consent of instructor. (Fall and Spring)

CS 2420 (810:142). Computer Architecture and Parallel
Programming — 3 hrs.
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 (810:041). (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 (810:051); CS 1520 (810:052); CS 1800
(810:080); or department approval. (Fall and Spring)

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 (810:052); CS 1800 (810:080). (Fall and Spring)

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
(810:052); CS 1100 (810:017) or COMM 2555 (48C:025). (Variable)

CS 3120/5120 (810:112g). 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 (810:052);
CS 1800 (810:080); junior standing. Prerequisite(s) for non-Computer
Science majors: junior standing; department approval. (Variable)

CS 3140/5140 (810:114g). Database Systems — 3 hrs.
Storage 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 (810:052); CS 1800 (810:080); junior
standing. (Fall)

CS 3150/5150 (810:115g). 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
(810:052); CS 1800 (810:080); junior standing. (Spring)

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 taken for up to six credits total. (Fall and Spring)

CS 3430/5430 (810:143g). Operating Systems — 3 hrs.
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 (810:041);
CS 1520 (810:052); CS 1800 (810:080); junior standing. (Spring)

CS 3470/5470 (810:147g). Networking — 3 hrs.
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 (810:041); CS 1520 (810:052); CS 1800 (810:080); junior
standing. Prerequisites for Department of Technology majors:
TECH 1037 (330:037); TECH 2041 (330:041); TECH 2042 (330:042);
CS 1160 (810:036). (Fall)

CS 3510 (810:151). 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
(810:052); CS 1800 (810:080). (Spring)

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 (810:052); CS 1800 (810:080). (Variable)

CS 3540 (810:154). 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
(810:052); CS 1800 (810:080); junior standing. (Spring)

CS 3610/5610 (810:161g). Artificial Intelligence — 3-4 hrs.
Models of intelligent behavior and problem solving; knowledge
representation and search methods; learning; topics such as
knowledge-based systems, language understanding, and vision;
optional 1-hour lab to include exploration of one or more additional
AI programming techniques such as robotics, symbolic programming,
heuristic programming, etc. Prerequisite(s): CS 1520 (810:052);
CS 1800 (810:080); junior standing. (Variable)

CS 3650/5650 (810:166g). 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 (810:052) and CS 1800 (810:080); junior standing. (Spring)

CS 3730/5730 (810:173g). Project Management — 1 hr.
Examination of problems of organizing, controlling, managing, and evaluating a software project; software metrics and human input. Prerequisite(s): CS 2530 (810:053); junior standing. (Fall and Spring)

CS 3750/5750 (810:175g). 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 2720 (810:172); junior standing. (Variable)

CS 3810/5810 (810:181g). Theory of Computation — 3 hrs.
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 (810:080); junior standing. (Variable)

CS 4400/5400 (810:140g). 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 (810:147g); junior standing. (Variable)

CS 4410/5410 (810:141g). System Security — 3 hrs.
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 (810:147g); junior standing. (Variable)

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 incidence response to compromised systems. Prerequisite(s): CS 1410 (810:041); CS 1520 (810:052); CS 1800 (810:080); CS 3430/5430 (810:143g).

CS 4550/5550 (810:155g). Translation of Programming Languages — 3 hrs.
Introduction to analysis of programming languages and construction of translators. Prerequisite(s): CS 2530 (810:053) and one of the following: CS 3530 (810:153), CS 3540 (810:154), CS 3810/5810 (810:181g); junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730 (810:173g). (Variable)

CS 4620/5620 (810:162g). 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 (810:053); CS 3610/5610 (810:161g) or equivalent; junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730 (810:173g). (Variable)

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 (810:053); CS 2720 (810:172); junior standing. Prerequisite(s) or corequisite(s): CS 3730/5730 (810:173g). (Fall)

CS 4800 (810:180). Undergraduate Research in Computer Science — 1-3 hrs.
May be repeated for maximum of 6 hours. (Fall and Spring)

CS 4880/5880 (810:188g). 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 (810:052); CS 1800 (810:080); junior standing. (Variable)

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 (810:142) or CS 3430/5430 (810:143g). Prerequisite(s) for non-Computer Science majors: proficiency in a high-level programming language; consent of instructor. (Variable)

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 (810:153). (Variable)

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 (810:181g). (Variable)