**BACHELOR OF SCIENCE IN COMPUTER SCIENCE (BSCS)**

**SUGGESTED COURSE SEQUENCE**

<table>
<thead>
<tr>
<th>FRESHMAN YEAR</th>
<th>SOPHOMORE YEAR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fall Semester</strong></td>
<td><strong>Spring Semester</strong></td>
</tr>
<tr>
<td>ENGL 1150 3</td>
<td>ENGL 1160 3</td>
</tr>
<tr>
<td>CMST 1110 3</td>
<td>Nat/Phys Sci 4</td>
</tr>
<tr>
<td>CIST 1400 3</td>
<td>CSCI 1620 3</td>
</tr>
<tr>
<td>MATH 1950 5</td>
<td>MATH 1960 5</td>
</tr>
<tr>
<td>Elective 1</td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL 15</strong></td>
<td><strong>TOTAL 15</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JUNIOR YEAR</th>
<th>SENIOR YEAR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fall Semester</strong></td>
<td><strong>Spring Semester</strong></td>
</tr>
<tr>
<td>MATH 2050 3</td>
<td>CSCI 3550 3</td>
</tr>
<tr>
<td>CSCI 3710 3</td>
<td>CSCI 3660 3</td>
</tr>
<tr>
<td>Humanities 3</td>
<td>CSCI 4350 3</td>
</tr>
<tr>
<td>Div/Soc Science 3</td>
<td>Div/Humanities 3</td>
</tr>
<tr>
<td>Core Extension 3</td>
<td>Core Extension 3</td>
</tr>
<tr>
<td><strong>TOTAL 15</strong></td>
<td><strong>TOTAL 15</strong></td>
</tr>
</tbody>
</table>
Overview of Courses Required for the Major

1. Course number and name
   CIST1400 – Introduction to Computer Programming

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Robert Fulkerson

4. Text book, title, author, and year

5. Specific course information
   Catalog description: An introduction to programming within the context of a high
   level modern programming language. Coverage of fundamental programming
   concepts and program design; including arrays, user defined types, and objects.
   This course has an optional laboratory component in CIST 1404.
   Prerequisites of the course: MATH 1310
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course,
   • The student should be able to write programs in the Java language that
     accomplish moderately difficult tasks.
   • The student is prepared for the continuation of their Java studies in CSCI 1620.
   • The student should be comfortable working in a Unix-based console environment.

Student outcomes addressed:
(a) An ability to apply knowledge of computing and mathematics appropriate to the
    discipline;
(b) An ability to analyze a problem, and identify and define the computing
    requirements appropriate to its solution;
(c) An ability to design, implement and evaluate a computer-based system, process,
    component, or program to meet desired needs;
(i) An ability to use current techniques, skills, and tools necessary for computing
    practices
7. Brief list of topics to be covered
   - Overview of programming languages
   - Introduction to console-based Java Applications
   - Algorithms and Control Structures
   - Arithmetic and Primitive Data Types
   - Methods
   - Arrays, Random Numbers and Strings
   - Introduction to Classes and Objects
1. Course number and name
   CIST 2100– Organizations, Applications and Technology

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Paul van Vliet

4. Text book, title, author, and year

5. Specific course information
   Catalog description: This survey course provides an introduction to organizations and the role information and information systems play in supporting operations, decision-making, processes, quality management, and strategic activities of an organization. In addition, the course covers management of the IS function, strategic and regulatory issues of telecommunications, and ethical and legal issues.

   Prerequisites of the course: ENGL 1150 prior to or concurrent with enrollment

   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will
   • Understand the role of information systems, their exciting potential, and the associated challenges in today’s competitive and global business environment
   • Understand the diversity of information systems and networks in the enterprise
   • Understand information systems, organizations and management models, and their impact on the decision making process
   • Understand how Internet technology, electronic commerce, and electronic Web-based systems have transformed organizations, business models, supply chains and quality
   • Understand ethical and social issues related to information systems
   • Understand the technical foundations of information systems, including infrastructure, databases, telecommunications, and security and control
   • Have a basic understanding of networks, including the Internet
   • Understand the business value of systems and technology
   • Understand how organizations can use a variety of systems for managing knowledge, enhancing decision making, and collaborating
• Understand the overall process of systems development, including phases and techniques for systems analysis, design, implementation, and project management
• Appreciate the global dimensions of information systems and organizations

Student outcomes addressed:

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(d) An ability to function effectively on teams to accomplish a common goal;
(e) An understanding of professional, ethical, legal, security, and social issues and responsibilities
(g) An ability to analyze the local and global impact of computing on individuals, organizations and society
(i) An ability to use current techniques, skills, and tools necessary for computing practices
(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

7. Brief list of topics to be covered
   a. Information Systems in Global Business Today
   b. Global E-Business and Collaboration
   c. Information Systems, Organizations, and Strategy
   d. Ethical and Social Issues in Information Systems
   e. IT Infrastructure and Emerging Technologies
   f. Foundations of Business Intelligence: Databases and Information Management
   g. Telecommunications, the Internet, and Wireless Technology
   h. Securing Information Systems
   i. Achieving Operation Excellence and Customer Intimacy: Enterprise Applications
   j. E-Commerce: Digital Markets, Digital Goods
   k. Managing Knowledge
   l. Enhancing Decision Making
   m. Building Information Systems
   n. Managing Projects
   o. Managing Global Systems
1. Course number and name
   CIST 2500– Introduction to Applied Statistics for IS&T

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Lotfollah Najjar

4. Text book, title, author, and year

5. Specific course information
   Catalog description: The course emphasizes the function of statistics in
   information science and technology including topics such as descriptive statistical
   measures, probability discrete probability, sampling, estimation analysis,
   hypothesis testing, regression, and analysis of variance. A well-known computer
   package will be used to support the problem-solving process.
   Prerequisites of the course: Math 2040 or Math 2030 or CSCI 2030.
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will
   - Gain an overview of the functions of statistics in modern business.
   - Understand the concepts and the development of the skills needed to apply
     statistics to the business decision-making process.
   - Understand the use of computer-based statistical tools and provide opportunities
     to use them to analyze business problems.

   Student outcomes addressed:
   (a) An ability to apply knowledge of computing and mathematics appropriate to the
       discipline;
   (b) An ability to analyze a problem, and identify and define the computing
       requirements appropriate to its solution;
   (i) An ability to use current techniques, skills, and tools necessary for computing
       practices;

7. Brief list of topics to be covered
a. Data and Statistics
b. Descriptive Statistics I: Tabular and Graphical Methods
c. Descriptive Statistics II: Numerical Methods
d. Continuous Probability Distributions
e. Sampling and Sampling Distributions
f. Interval Estimation
g. Hypothesis Testing
h. Statistical Inference about Means with Two Populations
i. Analysis of Variance
j. Regression Analysis
k. Interpretation of Statistics for Technology Professionals
1. Course number and name  
   CIST 3000– Information Technology Ethics

2. Credits and contact hours  
   3 credit hours

3. Instructor’s or course coordinator’s name  
   Ilze Zigurs

4. Text book, title, author, and year  

5. Specific course information  
   Catalog description: Advanced Composition for IS & T provides students with instruction and practice in academic writing for the technical sciences. The course focuses on principles of rhetoric and composition, advanced library-based research techniques, academic modes of writing suited to the technical sciences, style, grammar, and punctuation, all with attention to adapting writing to suit the needs of various academic and professional audiences.
   Prerequisites of the course: ENGL 1160.
   Required or Elective: Required

6. Specific goals for the course  
   On successful completion of the course, a student will develop  
   - an understanding of various writing strategies and genres  
   - an ability to apply field-related, advanced research skills  
   - an understanding of grammar and language issues  
   - a developed proficiency in appropriate language for audiences  
   - an understanding of the writing process  
   - an understanding of writing functions  
   - an awareness of audience

   Student outcomes addressed:  
   (f) An ability to communicate effectively with a range of audiences
7. Brief list of topics to be covered
   • Introduction to technical communications
   • Overview of the technical writing process
   • Memos
   • Thinking critically about the research process
   • Meeting the needs of specific audiences
   • Evaluating and interpreting information primary research
   • Weighing the ethical issues
   • Summarizing research findings
   • Formal analytical reports
   • Social media
   • Organizing for readers
   • Proposals
   • Workplace letters
   • Executive summaries
   • Technical definitions, specifications
   • Informal reports
   • Editing for a professional tone and style
   • Designing pages and documents
   • Designing and testing documents for usability
   • Designing visual information
   • Transmittal letters
   • Email and instant messaging
1. Course number and name
   CIST 3110– Information Technology Ethics

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Leah Pietron

4. Text book, title, author, and year

5. Specific course information
   Catalog description: The course will cover the development and need for issues regarding privacy and the application of computer ethics to information technology.

   Prerequisites of the course: Math 2040 or Math 2030 or CSCI 2030.

   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will
   - Be able to determine the impact of the privacy laws on information security policies.
   - Understand the issues related to intellectual freedom, intellectual property, and copyright law as they relate to electronic publishing.
   - Be able to determine and identify ethical procedures and behaviors in the organization related to information security.
   - Be able to identify issues of professional conduct in information technology case studies.
   - Be able to apply University standards of ethical conduct in preparing assignments for all coursework.
   - Learn the areas most impacted by ethical decisions by professionals in the computing field and will gain skills in making such decisions.
   - Identify key ethical concerns of information technology specialists.
   - Apply theories of ethics to case situations in the context of organizational use of information technology.
   - Appreciate how rapid changes in technology might affect ethical issues and changing norms of behavior.
   - Understand the issues related to privacy and confidentiality as they relate to information technology.
• Understand the ethical issues associated with gathering, storing and accessing genetic information in databases.
• Recognize the differences in ethical codes of conduct in different cultures and countries.
• Understand the ethical issues that arise from findings in genomic and bioinformatics analyses.

Student outcomes addressed:
(e) An understanding of professional, ethical, legal, security, and social issues and responsibilities
(g) An ability to analyze the local and global impact of computing on individuals, organizations and society
(h) Recognition of the need for, and an ability to engage in, continuing professional development

7. Brief list of topics to be covered
• Frameworks for Ethical and Policy Analysis
• Unwrapping the Gift
• Privacy
• Freedom of Speech
• Intellectual Property
• Computer Crime
• Evaluating and Controlling Technology
• Errors, Failures, and Risk
• Professional Ethics and Responsibilities
• The Software Engineering Code and the ACM Code
• Ethical Cases in Social, Legal, Privacy and Ethics
1. Course number and name
   CSCI 1620 – Introduction to Computer Science II

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Patrick Cavanaugh

4. Text book, title, author, and year

5. Specific course information
   Catalog description: Advanced topics in programming; topics in data representation and manipulation, data structures, problem solving and algorithm design. This is the follow up course to CIST 1400 Intro to Computer Programming. This course will focus more on the object oriented aspect of programming using the Java programming language. The basics of algorithms, including basic sorting techniques and Big O notation will be discussed, as well as introduction to data structures and file I/O.
   Prerequisites of the course: CSCI 1610 or CIST 1400
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will be expected
   • To fluently create and use Java classes
   • Understand and utilize basic sorting techniques
   • Develop algorithms and determine their efficiency
   • Create and use basic data structures
   • Perform file I/O in Java

   Student outcomes addressed:
   (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;

(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;

(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;

(k) An ability to apply design and development principles in the construction of software systems of varying complexity;

7. Brief list of topics to be covered
   - Overview of programming languages
   - Introduction to console-based Java Applications
   - Algorithms and Control Structures
   - Arithmetic and Primitive Data Types
   - Methods
   - Arrays, Random Numbers and Strings
   - Introduction to Classes and Objects
1. Course number and name
   CSCI 2030 – Mathematical Foundations of Computer Science

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Mahadevan Subramaniam

4. Text book, title, author, and year

5. Specific course information
   Catalog description: This course introduces discrete mathematics concepts that are foundational for the study of computer science. The covered topics include functions, relations, and sets, basic logic, methods of proof, mathematical induction, computational complexity, recursion, counting, recurrences, and relations.
   Prerequisites of the course: MATH 1950 or MATH 1930
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will be:
   - Proficient in the logical concepts of propositional and aspects of first-order logic including quantification and proof methods based on these logical foundations.
   - Able to comprehend and construct proofs used to reason about the working and correctness of several computing artifacts including data types, algorithms, and programs.
   - Proficient in the use of recursive methods for constructing finite and infinite sets, functions, and data types and understand the close relationship between inductive reasoning and recursive specifications.
   - Able to comprehend and formulate recursive sets and functions, and data types and reason about the behavior of these recursive structures using mathematical and structural induction methods.
   - Proficient in basic counting principles including inclusion-exclusion, pigeon-hole principle, binomial theorem, and recurrences.
   - Able to use counting techniques to quantitatively estimate the space and the time requirements of data, and algorithms designed to solve problems in several computer applications.
   - Students will also learn the theory of relations and be able to reason about relations specified using computer databases.
Student outcomes addressed:
(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
(g) An ability to analyze the local and global impact of computing on individuals, organizations and society
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

7. Brief list of topics to be covered
   • The Foundations: Logic and Proofs
   • Basic Structures: Sets, Functions, Sequences, and Sums
   • The Fundamentals: Algorithms, the Integers, and Matrices
   • Induction and Recursion
   • Counting
   • Advanced Counting Techniques
   • Relations
1. Course number and name
   CSCI 2240 – Introduction to C Programming

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Patrick Cavanaugh

4. Text book, title, author, and year

5. Specific course information
   **Catalog description:** Programming in C in a UNIX® operating system environment; algorithm and program development and file manipulation using C; UNIX-like utility development.

   This course is designed primarily for individuals already possessing programming facility in another high-level language who want to learn how to program in C in the UNIX environment.

   **Prerequisites of the course:** CSCI 1620
   **Required or Elective:** Required

6. Specific goals for the course
   On successful completion of the course, a student will
   - Be able to construct syntactically correct C programs using all language features.
   - Be able to recognize and correct syntax errors in C programs. This includes correct declarations and prototypes in both “K&R” C and ANSI C.
   - Be able to use primitive and complex data types and utilize/implement type conversion (This includes structures, typedefs, unions, enumerations and bit fields).
   - Be able to effectively use the operators in C.
   - Be able to explain the scope, lifetime, and initialization possibilities for all variable types in C. This includes the use of static to alter scope.
• Be able to describe and the basic low-level UNIX input/output facilities, including at least the open, close, read, write, unlink, and lseek system calls.
• Be able to explain and use the malloc and free functions to create and destroy simple dynamic data structures like variable-sized arrays and linked lists.
• Be comfortable with standard Unix command line tools, make, man, nm, etc., as well as concepts of pipes and redirection in the shell and use of the preprocessor and linker.

Student outcomes addressed:
(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
(k) An ability to apply design and development principles in the construction of software systems of varying complexity;

7. Brief list of topics to be covered
   a. Basics of the C language
   b. Conditional execution
   c. Looping and iteration
   d. Arrays and strings
   e. Functions
   f. Storage classes and scope
   g. Additional data types
   h. Pointers
   i. Command line processing
   j. The C preprocessor
   k. Explicit dynamic storage allocation
   l. Standard C library input/output functions
   m. UNIX system calls and library functions
   n. String handling functions (string.h)
   o. Character classification functions (ctype.h)
   p. Simple mathematical functions (math.h)
   q. Other standard library functions
Course number and name
CSCI 3320 – Data Structures

1. Credits and contact hours
   3 credit hours

2. Instructor’s or course coordinator’s name
   Jong-Hoon Youn

3. Text book, title, author, and year
   a. Data Structures and Algorithms Analysis in Java, Mark A. Weiss, Addison
   b. A Practical Introduction to Data Structures and Algorithm Analysis, Clifford
   c. Algorithms in Java, Parts 1-4 (3rd Edition), Robert Sedgewick, 3rd edition,
      Addison Wesley, 2002.
   d. Introduction to Algorithms, Cormen, Leiserson and Rivest, McGraw Hill, 3rd

4. Specific course information
   Catalog description: The purpose of this course is to familiarize students with
   algorithms and programming techniques for different data structures used in
   software systems. Students will also learn time analysis of computer programs,
   various searching and sorting techniques, program solving strategies and storage
   management algorithms.

   Prerequisites of the course: CSCI 1620 and CSCI 2030 (or equivalent)
   Required or Elective: Required

5. Specific goals for the course
   On successful completion of the course, a student will
   • Be Familiarize with different data structures such as lists, stacks, queues, trees,
     heaps, hash-tables and file structures which are used to store runtime data of a
     computer program
   • Understand different data manipulation strategies including different algorithms
     for sorting and searching items within various data structures
   • Be exposed to different program solving problems including greedy
     programming, dynamic programming, and backtracking
   • Be able to develop algorithms and solutions to different data structures related
     problems
   • Understand the importance of efficiency in terms of time and storage space for the
     computer programs developed by the students

   Student outcomes addressed:
(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;

(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;

(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

6. Brief list of topics to be covered
   • Algorithm Analysis and Time complexity
   • Lists, Stacks, Queues
   • Trees
   • Hashing
   • Priority Queues (Heaps)
   • Sorting
   • Graph Algorithms
   • Algorithm Design Techniques
Course number and name
CSCI 3550 – Communication Networks

1. Credits and contact hours
3 credit hours

2. Instructor’s or course coordinator’s name
Azad Azadmaneh

3. Text book, title, author, and year

4. Specific course information
   Catalog description: The purpose of this course is to familiarize students with algorithms and programming techniques for different data structures used in software systems. Students will also learn time analysis of computer programs, various searching and sorting techniques, program solving strategies and storage management algorithms.
   Prerequisites of the course: CSCI 1620 and CSCI 2030 (or equivalent)
   Required or Elective: Required

5. Specific goals for the course
On successful completion of the course, a student will
   • Learn about traditional and modern network technologies.
   • Learn about details of common network protocols.
   • Have a better understanding Internet implementation.
   • Have the basic ability to write application programs.
   • Learn how to compare different network protocols.
   • Have the ability to configure and debug their computers in case of network difficulties.
   • Be able to work-out and explain the important networking performance parameters.
   • Have a good understanding of various addressing types and the reasons behind them.
   • Have the ability to clearly distinguish between TCP/IP vs. Ethernet; and when they are applied.
   • Have the basic understanding of how the impaired communicated messages get corrected.
• Have a good understanding on how the major components of the entire protocol suite fit together.
• Have the basic understanding of various protocols in each layer.
• Have a better appreciation for the unique characteristics of wireless communication in comparison to wired technologies.

Student outcomes addressed:
(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(i) An ability to use current techniques, skills, and tools necessary for computing practices;
(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

6. Brief list of topics to be covered
   • Application Layer
   • Transport Layer
   • Network Layer
   • Data-link Layer
   • Physical layer
1. Course number and name
   CSCI 3660 – Theory of Computation

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Hai-Feng Guo

4. Text book, title, author, and year

5. Specific course information
   Catalog description: The course is intended for an introductory course on formal languages, automata, and computability. The topics covered in the course include finite automata, non-determinism, regular expressions and languages, context-free grammars, pushdown automata, context-free languages, Backus normal form, ambiguity, Turing machines, decidability, and the Chomsky Hierarchy.
   Prerequisites of the course: CSCI 1620 and CSCI 2030 (or equivalent)
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will understand topics: Finite Automata, Regular Languages, Regular Grammars, Context-Free Languages, Pushdown Automata, Turing Machines, Decidability, and the Chomsky Hierarchy.

   Student outcomes addressed:
   (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
   (c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

7. Brief list of topics to be covered
   - Finite Automata
   - Regular Languages and Regular Grammars
   - Properties of Regular Languages
   - Context-Free Languages
   - Simplification of Context-Free Grammars and Normal Forms
   - Pushdown Automata
   - Properties of Context-Free Languages
   - Turing Machines
   - Other Models of Turing Machines
   - A Hierarchy of Formal Languages and Automata
   - Limits of Algorithmic Computation
1. Course number and name
   CSCI 3710 – Introduction to Digital Design and Computer Organization

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Hassan Farhat

4. Text book, title, author, and year

5. Specific course information
   Catalog description: This course starts with the basics of the digital design and progresses to arithmetic units design, register files and datapath design, and computer organization. Topics covered from digital design include: Boolean algebra and minimization; primitive gates and the RC model for gate delay; decoders, encoders, multiplexers and demultiplexers; memory primitives; sequential circuits design and analysis procedures; counters and registers. Topics covered from computer organization include number representations, arithmetic, logic and shift unit design for unsigned and signed numbers; register-files single-output and multiple-output ports organization and design; memory organization and design DRAM, SRAM, ROM and flash; the three computer organizations stack, AC and general-purpose; and assembly programming.

   Prerequisites of the course: CSCI 3320
   Required or Elective: Required

6. Specific goals for the course

   1.1 On successful completion of the course, a student will understand topics:
      o gate designs at switch level,
o Boolean algebra, minimization, and combinational circuit design
o primitive memory elements, analysis and design of sequential circuits
o number representations
o computer organization at the register level
o the design of arithmetic, logic and shift units: signed and unsigned
o register-files and bus design
o memory design
o assembly instruction formats for different architectures
o converting high-level constructs to assembly constructs

Student outcomes addressed:
(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

7. Brief list of topics to be covered
   - Gate designs at the switch level
   - RC circuits and gate delays
   - Boolean algebra, minimization, and design
   - Decoders, multiplexers, encoders and demultiplexers design
   - Computer data representations
   - Computer arithmetic logic units efficient designs
   - Sequential circuits
   - Sequential circuit design procedures and analysis
   - Register-Files
   - Datapath organization
   - Memory organization and design at gate and switch level
   - The three architectures
   - Register Transfer Language and realization
   - Sample computer organization
Course number and name
CSCI 4000 – Assessment

1. Credits and contact hours
0 credit hours

2. Instructor’s or course coordinator’s name
Harvey Siy

3. Text book, title, author, and year
n/a

4. Specific course information
Catalog description: This course provides various resources to students about to graduate, and provides a mechanism that guarantees these students complete the final assessments required to maintain the currency and quality of the program. It is intended for undergraduate computer science majors in their last semester prior to graduation. It is required for all students entering after the spring 2004 semester.

Prerequisites of the course: All degree requirements should be completed by the end of the semester during which this course is taken. Students taking this course will be expected to file an application for graduation during the semester.

Required or Elective: Required

5. Specific goals for the course
n/a

6. Brief list of topics to be covered
n/a
1. Course number and name
   CSCI 4220 – Principles of Programming Languages

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Victor Winter

4. Text book, title, author, and year

5. Specific course information
   Catalog description: This course covers the foundational concepts and principles underlying the design and implementation of programming languages. Language constructs including assignment, equality, references, aggregations, scope, encapsulation, and parameter passing are discussed. A central theme is how a particular language construct relates to the concept of equational reasoning (referential transparency). Formal notations for describing syntax and semantics are presented.
   Prerequisites of the course: CSCI 3320 & CSCI 3660
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, will be expected to understand:
   - **Language Design Principles.** To appreciate the intimate relationship between language and thought and why therefore a well-designed language must conform to certain principles
   - **Syntax.** To be able to read and write formal notations describing the syntax of a programming language
   - **Semantics.** To develop a more precise understanding of language semantics as well as the issues and trade-offs underlying various language design decisions (e.g., mutable versus immutable values).
   - **Types.** To develop a rudimentary appreciation of the contribution of type systems to language design.
   - **Language Constructs.** To understand how various language constructs relate and differ from one another both within a single language as well as across languages (e.g., parameter passing and assignment).
   - **Language Paradigms.** To develop a basic understanding of the similarities and differences among a variety of common language paradigms (e.g., imperative, object-oriented, functional, declarative). Students will be expected to write programs in non-imperative language paradigms.
• **Experience and Ethics.** To become better prepared culturally to succeed in the workplace.

Student outcomes addressed:
(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(k) An ability to apply design and development principles in the construction of software systems of varying complexity

7. Brief list of topics to be covered
   • Background and Motivation
     o Language Design Principles
     o Brief history of programming languages
   • Syntax
   • Semantics
   • Language Paradigms
   • Experience and Ethics
1. Credits and contact hours
   3 credit hours

2. Instructor’s or course coordinator’s name
   Azad Azadmanesh

3. Text book, title, author, and year

4. Specific course information
   Catalog description: Students are introduced to the technology and architecture of computers. The objectives are to introduce students to a large body of concepts and current trends in computer architecture, so that the design issues and their tradeoffs can be appreciated. Topics covered include top level view of computers, performance metrics; cache memory and cache coherence; internal and external memory (RAM, disk, RAID, magnetic and optical); pipelined computer architecture; RISC versus CISC; and parallel processing.
   Prerequisites of the course: CSCI 3320 & CSCI 3710
   Required or Elective: Required

5. Specific goals for the course
   On successful completion of the course, a student will
   - learn a large body of concepts and current trends in computer architecture.
   - learn about the design issues and their tradeoffs
   - learn about performance parameters and related issues such as: speed up, utilization, throughput, bandwidth, response time, CPU & system times
   - learn parallel processing, multiprocessor system design & tradeoffs

   Student outcomes addressed:
   (c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
   (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems
in a way that demonstrates comprehension of the tradeoffs involved in design choices

6. Brief list of topics to be covered
   • Top Level View of Computers
   • Cache Memory
   • Internal Memory
   • External Memory
   • Instruction Sets:
   • Pipelined Processor Structure and Function
   • RISC vs. CISC
   • Parallel Processing
1. Course number and name
   CSCI 4500 – Operating Systems

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Stanley Wileman

4. Text book, title, author, and year

5. Specific course information
   Catalog description: Overview of operating systems and operating system principles. Concurrency, process scheduling and dispatch, memory management, security and protection, virtual machines, device management, and file systems.
   Prerequisites of the course: CSCI 3320 & CSCI 4350 (recommended)
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will
   - understand the purposes, major components and key mechanisms of operating systems;
   - understand the type of decisions involved in operating system design;
   - understand the context within which the operating system functions;
   - understand the relation between computer system architecture and operating system features;
   - understand the historical development of architecture and operating systems;
   - understand the major algorithms used in various operating system components and the factors used to evaluate different designs,
   - utilize the application program interface (API) for at least one contemporary operating system to construct programs that illustrate that API; and
   - be familiar with methods for providing concurrency, communication and synchronization among concurrent tasks.

   Student outcomes addressed:
   (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;

(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

7. Brief list of topics to be covered
   • Introduction to Operating Systems
   • Concurrency
   • Input/Output
   • Memory management
   • File systems
   • System security
1. Course number and name
   CSCI 4830 – Introduction to Software Engineering

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Harvey Siy

4. Text book, title, author, and year
   c. A. Fox and D. Patterson. Engineering Software as a Service: An Agile Approach
      http://beta.saasbook.info/
   d. R. Pressman. Software Engineering – A Practitioner’s Approach, 7th Ed.,
   f. S. Schach. Object-Oriented and Classical Software Engineering, 8th Ed.,

5. Specific course information
   Catalog description: Basic concepts and major issues of software engineering,
   current tools and techniques providing a basis for analyzing, designing,
   developing, maintaining and evaluating software systems. Technical,
   administrative and operating issues. Privacy, security and legal issues.

   Prerequisites of the course: CSCI 3320
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will
   • Perform analysis and design of small and medium-sized software project using
     structured methods.
   • Be able to participate in design of small and medium-sized software project using
     object-oriented software development methodologies.
   • Prepare software project management documents.
   • Be able to participate in a project team.
   • Develop parts/whole prototype as well as implementation of small or medium-sized
     software projects.
   • Understand socio-technical and ethical issues in the development of real-world
     software systems.
Student outcomes addressed:
(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(d) An ability to function effectively on teams to accomplish a common goal;
(e) An understanding of professional, ethical, legal, security, and social issues and responsibilities
(i) An ability to use current techniques, skills, and tools necessary for computing practices
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

7. Brief list of topics to be covered
   • Computer-based System Engineering
   • Software Processes
   • Project Management
   • Managing People
   • Software Cost Estimation
   • Quality Management
   • Process Improvement
   • Software Requirements
   • Requirements Engineering Processes
   • System Models
   • Software Prototyping
   • Architectural Design
   • Object-oriented Design
   • Design with Reuse
   • User Interface Design
1. Course number and name
   CSCI 4970 – Computer Science Capstone

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Harvey Siy

4. Text book, title, author, and year

5. Specific course information
   Catalog description: The Capstone Project completes a Computer Science student’s undergraduate experience. Students will work on a team-based real-world project, practicing software engineering skills and applying fundamental computer science principles acquired throughout their undergraduate study.

   Prerequisites of the course: CSCI 4830
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will
   • Apply fundamental computer science principles in solving real-world problems.
   • Develop communication skills through interaction with client.
   • Gain proficiency in modeling, implementing and testing large software applications.
   • Learn to work in teams.

   Student outcomes addressed:
   (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
   (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
   (c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
   (d) An ability to function effectively on teams to accomplish a common goal;
(e) An understanding of professional, ethical, legal, security, and social issues and responsibilities

(f) An ability to communicate effectively with a range of audiences

(i) An ability to use current techniques, skills, and tools necessary for computing practices

(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

7. Brief list of topics to be covered
   a. Team collaboration
      i. Group dynamics
      ii. Geographically distributed software development
   b. Modeling requirements and solutions
      i. Requirements elicitation and analysis
      ii. Architecture and design
   c. Source code analysis
      i. Reasoning about programs
      ii. Version control
   d. Testing
      i. Test case development
      ii. Regression testing
1. Course number and name
   MATH 1950– Calculus I

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Jack Heidel

4. Text book, title, author, and year

5. Specific course information
   Catalog description: This is a course in plane analytic geometry emphasizing the study of functions, limits, derivatives and applications, and an introduction to integration.

   Prerequisites of the course: MATH 1320 and MATH 1330, or MATH 1340.

   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will be
   - Proficient regarding limits and continuity.
   - Proficient regarding the derivative of a function, including logarithmic, exponential, and inverse trigonometric functions
   - Proficient regarding graphing the derivative
   - Proficient regarding the applications of the derivative
   - Proficient regarding the basics of integration

   Student outcomes addressed:
   (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;

7. Brief list of topics to be covered
   - Review of functions, including inverse functions, exponential and logarithmic functions
   - Limits and continuity
   - Derivatives
   - Applications of the derivative
   - The integral
1. Course number and name
   MATH 1960 – Calculus I

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Judith Carol Downey

4. Text book, title, author, and year

5. Specific course information
   Catalog description: This course introduces applications of integration, techniques of integration, infinite sequences and series, vectors in the plane, and polar functions. A mathematical software package is introduced, with required assignments.

   Prerequisites of the course: MATH 1950
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will be
   • Proficient regarding applications of the definite integral
   • Proficient regarding techniques of integration
   • Proficient regarding infinite sequences and infinite series
   • Proficient regarding polar coordinates and conics

   Student outcomes addressed:
   (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;

7. Brief list of topics to be covered
   • Applications of the Integral
   • Techniques of Integration
   • Sequences and Series
   • Vectors in the plane
1. Course number and name
   MATH 2050—Applied Linear Algebra

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Zhenyuan Wang

4. Text book, title, author, and year

5. Specific course information
   Catalog description: The purpose of the course is to present students with the
   basic concepts, terminology, and techniques for the solution of linear algebra
   problems, including the related computational considerations. Linear algebra is
   valuable for explaining fundamental principles and simplifying calculations in
   Mathematics, Statistics, Computer Science, Engineering, Physics, Biology, and
   Economics.
   Prerequisites of the course: MATH 1950
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will be
   - Proficient regarding linear equations in linear algebra.
   - Proficient regarding matrix algebra.
   - Proficient regarding determinants.
   - Proficient regarding vector spaces.
   - Proficient regarding eigenvalues and eigenvectors.
   - Proficient regarding orthogonality.

   Student outcomes addressed:
   (b) An ability to apply knowledge of computing and mathematics appropriate to the
   discipline;

7. Brief list of topics to be covered
   - Linear Equations in Linear Algebra
   - Matrix Algebra
   - Determinants
   - Vector Spaces
   - Eigenvalues and Eigenvectors
   - Orthogonality
1. Course number and name
   CMST 1100 Public Speaking Fundamentals

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Karen Dwyer

4. Text book, title, author, and year

5. Specific course information
   Catalog description: Public Speaking Fundamentals helps students become effective public speakers, as well as critical listeners and evaluators of public communication. Students will learn the principles of audience adaptation, topic selection, organization, development of ideas and presentation of speeches. Each student will design and present a minimum of four public speeches. (Special 'Speaking Confidently' sections are available for the students with excessive levels of fear about public communication. Contact the School of Communication for applications.)

   Prerequisites of the course: None
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will be able to:
   - Demonstrate knowledge of the basic principles of public speaking;
   - Describe and apply public speaking as a two-way communication process instead of a one-way performance;
   - Explain and demonstrate the steps in the speechmaking process;
   - Manage communication anxiety;
   - Select, narrow, and design communication objectives to fit the topic, situation, and audience;
   - Adapt messages and language to the needs and expectations of various audiences through the use of audience analysis;
   - Develop Information Literacy Skills;
   - Collect, analyze, select, and use supporting materials (from the library and other sources) to form informative and persuasive messages;
   - Select and effectively use visual aids and presentational software to enhance informative and persuasive messages;
   - Organize ideas, supporting material, and evidence into coherent, logical, and interesting messages using a structured outline format;
• Speak in an extemporaneous and conversational delivery style using effective eye contact, gestures, body movement, voice projection, and vocal variety;
• Speak ethically, confidently, and competently in public settings;
• Listen critically and evaluate public communication encountered in daily life.

Student outcomes addressed:
(f) An ability to communicate effectively with a range of audiences

7. Brief list of topics to be covered
• Speaking with Confidence and Decreased Anxiety
• The Speech Communication Model
• Speaking Freely and Ethically
• Listening to Speeches with Critical Listening Skills
• Analyzing the Audience
• Developing a Speech
• Selecting a Topic and Purpose
• Organizing and Outlining a Speech
• Introducing and Concluding a Speech
• Using Words Well: Speaker Language and Style
• Using Visual and/or Presentational Aids Effectively
• Researching the Speech Topic
• Developing Information Literacy Skills
• Using Supporting Material and Critical Thinking
• Delivering the Speech
• Organizing and Presenting Informative Speeches
• Organizing and Presenting Persuasive Speeches
• Using Effective Persuasive Strategies
• Organizing and Presenting Ceremonial Speaking
1. Course number and name
   ENGL 1150 English Composition I

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Nora Bacon

4. Text book, title, author, and year
   - Bash, Rachel, Maggie Christensen, and Tammie M. Kennedy, eds. From the Heartland: Critical Reading and Writing at UNO. Plymouth, MI: Hayden-McNeil, 2011.

5. Specific course information
   Catalog description: Instruction and practice in academic literacy practices, especially writing summaries, analyses, and critical essays in response to assigned texts. The purpose of Composition I is to introduce students to the writing practices of the academy. The course is designed around a sequence of reading and writing assignments designed to provide practice in critically reading, evaluating, and responding to other writers' texts; developing papers with a clear thesis, logical structure, and cohesive, well-developed paragraphs; writing clear sentences with usage and mechanics conforming to Standard Edited English. The course includes discussion of the purposes and processes of academic writing.
   Prerequisites of the course: Placement by English Placement and Proficiency Exam (EPPE) or ENGL 1050
   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will be
   - Improved proficiency in these skills –
     o close reading
     o active listening
     o summarizing a text
     o critically interpreting and evaluating texts
     o integrating (paraphrasing, quoting, and acknowledging) materials from other texts
     o evaluating other writers' drafts, giving appropriate feedback
     o sentence-level editing and proofreading
   - The ability to write a paper with these characteristics -
     o a clear thesis
     o a clear, reader-friendly structure
     o thorough, honest exploration of ideas
o clear, varied, well-constructed sentences
o inclusion of graphics as appropriate
o usage and mechanics conforming with Standard Edited English

- A generative conception of writing -
  o understanding of writing as a complex, recursive process involving prewriting, drafting, substantive revision, and editing
  o understanding of writing as a process whereby ideas are developed, explored, and evaluated
  o understanding of writing as communication addressed to a particular audience and governed by a particular set of purposes.

Student outcomes addressed:
(f) An ability to communicate effectively with a range of audiences

7. Brief list of topics to be covered
   - The writing process
   - Reading challenging texts
   - Analyzing audience in academic contexts
   - Providing constructive feedback
   - Writing a summary
   - Writing a personal response
   - Writing an analysis
   - Integrating material from other texts: paraphrase, quotation, citation
   - Developing a thesis
   - Organizing an essay: introductions, middle paragraphs, conclusions, transitions
   - Paragraph development and cohesion
   - Sentence structure and style
   - Editing common sentence-level errors
   - Editing for usage and mechanics
1. Course number and name
   ENGL 1160 English Composition II

2. Credits and contact hours
   3 credit hours

3. Instructor’s or course coordinator’s name
   Nora Bacon

4. Text book, title, author, and year

5. Specific course information
   Catalog description: Instruction and practice in academic inquiry, especially researching, analyzing, and writing arguments. The purpose of Composition II is to further develop the writing skills taught in Composition I (reading, evaluating, and responding to other writers’ texts; developing papers with a clear thesis, logical structure, and cohesive, well-developed paragraphs; writing clear sentences with usage and mechanics conforming to Standard Edited English). Additionally, Composition II covers information literacy – the effective use of print and digital resources in a university library – and analysis of arguments.

   Prerequisites of the course: Placement by English Placement and Proficiency Exam (EPPE) or ENGL 1150

   Required or Elective: Required

6. Specific goals for the course
   On successful completion of the course, a student will be
   - The course objectives are to improve students’ proficiency as writers,
   - highlighting the writing skills valued in the academy (research and argumentation). Specific objectives are listed below.
   - Improved proficiency in these skills –
     - Close reading, summary, and analysis of other writers’ texts
     - Navigating the college library
     - Locating and evaluating print and online information sources
     - Evaluating arguments
     - Crafting well-informed, carefully-reasoned arguments
     - Evaluating other writers’ drafts, giving feedback in appropriate ways
     - Sentence-level editing and proofreading
   - The ability to write papers with these characteristics –
     - A clear thesis defended by a well-reasoned argument
Thorough, honest exploration of ideas
Clear, varied, well-constructed sentences
Usage and mechanics conforming with standard edited English
Effective introduction and integration of sources
Documentation that conforms to MLA or APA guidelines

- A generative conception of writing as –
  - a complex, recursive process involving planning, drafting, substantive revision, and editing
  - a means to explore, evaluate, and communicate ideas, using one’s own writing to challenge and/or extend the thinking of others
  - communication addressed to a particular audience governed by a particular set of purposes and shaped by the conventions of the genre (e.g., position paper, proposal, evaluation)

Student outcomes addressed:
(f) An ability to communicate effectively with a range of audiences

7. Brief list of topics to be covered
   - Definitions of “argument”
   - The logical structure of arguments
   - Rhetorical strategies: appeals to ethos, logos, pathos
   - Kinds of evidence: personal experience, observation, interview data, survey data, testimony, etc.
   - Locating information in the university library
   - Evaluating print and online sources
   - Integrating material from other texts: summary, paraphrase, quotation, response to counterarguments
   - Documentation styles (MLA and/or APA)
   - Organizing an extended argument: strategies for introductions, transitions, conclusions
   - Paragraph development and cohesion
   - Editing common errors in syntax, usage, and mechanics
   - Editing for clarity and style