SI200 Information Technology for the Junior Officer

Catalog: SI200 Information Systems for the Junior Officer (3-2-4). This is a hands-on lab course introducing computer programming and database management. Topics include: web programming using HTML and XHTML, structured and object oriented computer programming using a scripting language (such as JavaScript) or 4th Generation Language (such as Java or C++), and designing, implementing, and querying databases using a Database Management System (such as Access or SQL Server). The course includes a series of Internet computing and programming projects of increasing complexity. No prior knowledge of databases, web programming, or computer programming is assumed.

Textbook: The Complete Reference HTML, Third Edition, Thomas Powell, McGraw-Hill, Osborne Media, Fourth Edition, 2003.

Coordinator: TBD

Course Goals:

1. Familiarity with Internet and computing technology.
2. The ability to design and publish a web site using *any* available editor or tool.
3. Working knowledge of databases, including construction and query, with focus on using databases to solve problems, rather than on using a specific database management tool.
4. To introduce the fundamental concepts associated with problem solving using a structured programming language, including variables, functions, input/output, loops/statements, and algorithmic development.
5. To introduce object oriented programming as appropriate to advanced web development.
6. To introduce issues associated with Internet privacy and security.

Major Topics Covered:	**Hours**
1. Internet Privacy and Security	1
2. Internet, World Wide Web, and Browsers	2
3. Basic html/XHTML Programming	3
4. Web Graphics	2
5. Designing Lists and Tables	3
6. Web Style Sheets and Formatting	3
7. Frames and Forms	5
8. Introduction to Programming	2
9. Variables, Statements, and Arithmetic Operators	2
10. Input/Output	2
11. Looping Structures	2
12. Functions	3
13. Using Programming in Web Pages	3
14. Introduction to Databases	2

15. Database Design	3
16. Database Queries	2
17. Course Administration, Reviews	2
18. Exams	2
Total:	**44**
Labs:	**28**
Total:	**72**

Laboratory Projects: This course is lab based. Students will spend time each week in labs, programming and designing databases and web sites. There will be several projects required in the course, beginning with a simple web page and enhancing them to include interactive and dynamic elements. Programming projects and database design projects will also be included.

Oral and Written Communications: Students will be required to make one oral presentation typically following the web project.

Social and Ethical Issues: Internet privacy is a relevant and critical topic in the study of newer technologies. This course includes one hour on this topic.

SI204 Introduction to Computer Science

Catalog: SI204 Introduction to Computer Science (3-2-4). Introduction to algorithmic development, problem solving and software design. Principles and concepts to provide foundational knowledge and experience upon which later computer science courses will build. This is the first course for computer science majors. *Prereq: None.*

Textbook: Programming and Problem Solving with C++, Third Edition, Nell Dale, Chip Weems and Mark R. Headington, Jones and Bartlett Publishing, 2002.

References: C++ How to Program, Second Edition, Deitel & Deitel, Prentice Hall, 1998.

Coordinator: Chris Brown, Ph.D., Assistant Professor

Program Outcomes:
1. Be able to solve problems using the procedural, functional, and object-oriented programming paradigms.
2. Differentiate amongst the classic algorithms and data structures.
3. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To introduce the concepts of structured programming utilizing a high level programming language.
2. To introduce problem solving methods and algorithmic development.
3. To teach the design, coding, debugging, and documenting of programs using structured programming techniques.

Prerequisites by Topic: None

Major Topics Covered:	**Hours**
1. SEIofC: Property rights (copyright, patent, fair use), privacy (snooping), password security, abuse (harassing e-mail, viruses)	1
2. Basic Computer operation. Becoming familiar with the environment	1
3. An introduction to a programming language: program structure	1
4. Documentation, design, and programming style	2
5. Basic language constructs: basic data types, variables, assignment, arithmetic expressions, String class	5
6. Control flow, conditionals, logical expressions, iteration	6
7. File I/O	1
8. Functions, function arguments, and basic recursion	7
9. Arrays, arrays of characters, and simple pointers	6
10. Debugging	2
11. Introduction to classes, data members, structured types	3
12. The structure of multi-file programs	1
13. Pointers to single objects, linked lists	4

14. Exams 2
15. Review, evaluations, course admin 2
Total: 44

Laboratory Projects (specify number of weeks on each): Approximately 28-30 hours of closed laboratory time will be used to reinforce the concepts discussed in class. Labs are as follows (2 lab hours = 1 week).

1. Lab Familiarization, "Hello World!" (2 hours)
2. Types, Expressions and Representation (2 hours)
3. Control Structures (2 hours)
4. Loops (2 hours)
5. More Loops (2 hours)
6. Debugging (2 hours)
7. Simple Functions: Top Down Programming (2 hours)
8. Complex Functions (2 hours)
9. Recursion (2 hours)
10. Arrays (2 hours)
11. Pointers (2 hours)
12. Combining What We Know: Data Structures and Algorithms (2 hours)
13. Classes without member functions (2 hours)
14. Linked lists (2 hours)

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structures	1	
Algorithms	1	
Software Design	1	
Computer Organization and Architecture		
Concepts of Programming Languages	1	
Total	4	0

Oral and Written Communications:
Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth). Property rights (copyright, patent, fair use) with respect to software, software piracy including MP3 and Napster, password

security, harassing e-mail (1 hour). These topics will be evaluated via a hypothetical question on an exam.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. N/A.

Analysis:
Please describe the analysis experiences common to all course sections. Students are required on a continual basis to analyze English descriptions of computing problems. A certain degree of detail is purposely omitted as a method of forcing the students to gain a deeper appreciation for fully understanding a problem before developing a solution.

Design:
Please describe the design experiences common to all course sections. Students continually must design programs based on problem descriptions. Students are introduced to top-down, step-wise refinement, and stub programming.

SI221 Data Structures

Catalog: SI221 Data Structures (2-2-3). Data representation and information management. Dynamic memory, recursion, lists, stacks and queues. Storage structures, allocation and manipulation. *Prereq: SI204*

Textbook: Data Abstractions and Problem Solving, Third Edition, Carrano and Prichard, Addison Wesley, 2002

References: Programming and Problem Solving with C++, Third Edition, Dale, Weems and Headington, Jones and Bartlett, 2002.

Coordinator: Frederick Crabbe, Ph.D., Assistant Professor

Program Outcomes:
1. Be able to solve problems using the procedural, functional, and object-oriented programming paradigms.
2. Differentiate amongst the classic algorithms and data structures.
3. Evaluate contemporary legal, social, and ethical issues in computing professions.

Prerequisites by Topic:
1. An introduction to a programming language: program structure.
2. Documentation, design, and programming style.
3. Basic language constructs: basic data types, variables, assignment, arithmetic expressions and public classes.
4. Control flow, conditionals, and logical expressions.
5. Functions, function arguments, and basic recursion.
6. Linked lists.
7. Basic array operations; searching and sorting arrays.
8. File input and output.
9. Types and overloading.
10. Debugging and error handling.

Major Topics Covered:	**Hours**
1. SEIofC: Building safe and reliable software in critical systems	1
2. Analysis of algorithms: complexity & efficiency	3
3. Problem solving, abstraction and programming paradigms, ADTs	3
4. Programming language issues- parameter passing models, scope & lifetime	3
5. Object-oriented programming, Object oriented design, encapsulation, inheritance, polymorphism	7
6. Generic programming, Standard Template Library, Function Objects	4
7. List ADT and implementation, iterators	1
8. Stacks and queues	2
9. Programming in UNIX	1

10. Exams 2
11. Review, evaluations, course admin 2
Total: 29

Laboratory Projects (specify number of weeks on each): Individual lab programming exercises are conducted to emphasize and reinforce class lectures. Each line represents one week.

1. Using UNIX
2. Programming in UNIX
3. Modular Programming
4. OOP 1
5. OOP 2
6. Using Stacks
7. Implementing Stacks
8. Using Queues
9. Implementing a linked list ADT
10. Templates
11. STL
12. Inheritance 1
13. Inheritance & Polymorphism
14. Polymorphism
15. Experimentation & analysis with algorithms

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structures	1.5	
Algorithms	.5	
Software Design	.5	
Computer Organization and Architecture		
Concepts of Programming Languages	.5	
Total	3	0

Oral and Written Communications:
Every student is required to submit at least two written reports (not including exams, tests, quizzes, or commented programs) of typically three pages and to make 0 oral presentation of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? Items covered included discussions

on building safe and reliable software in critical systems (pacemaker software). Discussion also focused on aircraft software (i.e., ALQ-39 iterations) as well as command and control simulation software. (1 hour). These topics will be evaluated via test questions.

Lecture on property rights and inappropriate use of computers. A question on computer ethics is included on one of the exams.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. Analysis of algorithms: complexity & efficiency => 1 hour.

Analysis:
Please describe the analysis experiences common to all course sections. Students are required to analyze the requirements of problem statements specific to the data structures under study. This analysis serves to aid in understanding the requirements in order to structure their solutions to both laboratories and projects. Students analyze algorithms to determine space and time complexity.

Design:
Please describe the design experiences common to all course sections. Prior to implementing their solutions to laboratories and projects students are encouraged to formulate models that focus on the solution domain. Object-oriented design is emphasized.

SI232 Computer Architecture and Organization

Catalog: SI232 Computer Architecture and Organization (4-0-4). Performance metrics, instruction set architectures, assembly language, logic design, memory hierarchies. *Prereq: SI204. Coreq: SI262.*

Textbook: Computer Organization & Design, Second Edition, Patterson & Hennessey, Morgan Kaufmann, 1998.

References: None.

Coordinator: Margaret M. McMahon, Ph.D., Assistant Professor

Program Outcomes:
1. Describe the fundamentals of computer networks, architecture, and operating systems.
2. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To present concepts and theories related to general-purpose computer systems, and to build an understanding of the planning, design, construction, and use of such systems.
2. To introduce strategies needed to critically evaluate the designs of computer systems.
3. To instruct midshipmen in the security, privacy and ethical issues relevant to information systems.

Prerequisites by Topic:
1. Boolean algebra
2. Circuit minimization
3. Elementary programming

Major Topics Covered:	**Hours**
1. SEIofC: Reverse Engineering, Copyright Law, DMCA, Fair Use	1
2. Computer Anatomy	1
3. Performance Metrics/Benchmarks	3
4. Instruction Set Architectures/Assembly Language	10
5. Digital Logic (Decoders, Multiplexors, PLAs, ROMs)	3
6. Computer Arithmetic, ALU Design	7
7. Sequential Logic and Memory (Flip-Flops, SRAM, DRAM)	5
8. Control Unit Design (Finite State Machines, Multiprogramming)	12
9. Exceptions and Interrupts	1
10. Pipelining	1
11. Memory Hierarchy (Caches, Virtual Memory)	6
12. Input/Output, Busses	3
13. Multi-processors	1
14. Exams	2

15. Review, evaluations, course admin 2
Total: 58

Laboratory Projects:
Four major assignments will be required. The first two assignments will involve programming in MIPS assembly language. The last two assignments will utilize the Logic Works circuit prototyping application. Each assignment will be of approximately two weeks in length.

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		
Software Design		
Computer Organization and Architecture	3	
Concepts of Programming Languages	1	
Total	4	0

Oral and Written Communications:
Every student is required to submit at least one written report of typically five pages covering and advanced topic in computer architecture or a specific chip design.

Social and Ethical Issues:
Reverse Engineering, Copyright Law, Digital Millennium Copyright Act, Fair Use and computer hardware (1 hour): Students are evaluated on their understanding of the topics by a short essay homework assignment and a final exam question.

Theoretical Content:
Number representation (1 hour)

Analysis:
Four major assignments (two in assembly language, two covering logic design) requiring individual problem analysis.

Design:
Four major assignments (two in assembly language, two covering logic design) requiring individual solution design.

SI262 Discrete Structures

Catalog: SI262 Discrete Structures (2-0-2). An introduction to the mathematical foundations of computing through logic, set theory, functions, proofs, counting, relations, prepositional logic, and Boolean algebra. *Prereq: SI204.*

Textbook: Discrete Mathematics and Its Applications, Fourth Edition, Kenneth Rosen, McGraw-Hill, 1999.

References: Discrete Mathematics, Third Edition, Ross and Wright, Prentice Hall, 1992.

Coordinator: Ric Crabbe, Ph.D., Assistant Professor

Student Outcomes:
1. Discuss the theoretical and mathematical foundations of computer science.

Course Goals:
1. To provide an introduction to the many mathematical foundations of computing.
2. To provide an introduction to numerical linear algebra.

Prerequisites by Topic:
1. Recursion principle
2. Arrays
3. Dot product
4. Algebra II
5. Geometry (proof techniques)

Major Topics Covered:	**Hours**
1. Propositional logic and first-order predicate calculus, well formed formulas	5
2. Set theory, functions, relations: properties, closures, partial orders, equivalence relations	3
3. Mathematical induction, proof techniques, program correctness	7
4. Counting, permutations, combinations	5
5. Recursive definitions and recurrence relations; solving recurrence relations	2
6. Numerical linear algebra: vector and matrix theory and operations, Gaussian elimination, determinants	3
7. Exams	2
8. Review, evaluations, course admin	2
Total:	**29**

Note: Boolean algebra and minimization of circuits must be covered first so students have the foundation necessary for the concurrent SI232 course.

Laboratory Projects (specify number of weeks on each): None.

Oral and Written Communications:
Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes durations. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? N/A.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

1. Boolean algebra (3 hours)
2. Counting/Permutations/Combinations (4 hours)
3. Logic (3 hours)
4. Mathematical induction/proof techniques (3 hours)
5. Numerical linear algebra: Matrix operations/linear independence/Gaussian elimination and determinants (5 hours)
6. Recursive definition/recurrence relations/solutions (3 hours)
7. Set theory/functions/relations/partial orders/equivalence relations (4 hours)

Analysis:
Please describe the analysis experiences common to all course sections.
1. Recurrence relation solutions

Design:
Please describe the design experiences common to all course sections. N/A.

SI321 Advanced Data Structures

Catalog: SI321 Advanced Data Structures (3-2-4). Advanced OO design, including abstract classes, polymorphisms, templates and inheritance. Advanced data structures including trees, heaps, priority queues and graphs. Graph algorithm analysis. *Prereq: SI221, SI262.*

Textbook: Data Structures and Algorithms in Java, Third Edition, M. T. Goodrich and R. Tamassia, Wiley, and Sons, 2003

References: None

Coordinator: Ric Crabbe, Ph.D., Assistant Professor

Program Outcomes:
1. Be able to solve problems using the procedural, functional, and object-oriented programming paradigms.
2. Differentiate amongst the classic algorithms and data structures.
3. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To define, design, implement, and apply advanced data structures such as the tree, heaps, and hashing.
2. To understand advanced Object Oriented Programming, inheritance, polymorphism and component design and reuse.
3. To study the mathematical foundation of graphs.
4. To study advanced data structures and algorithms in terms of ordering in time and space, complexity and efficiency.
5. To study searching and an analysis of different search mechanisms.

Prerequisites by Topic:
1. Object Oriented Programming
2. Searching and Sorting
3. Vectors, Lists, Stacks and Queues
4. Program performance

Major Topics Covered:	**Hours**
1. SEIofC: Equity of computer access	1
2. Inheritance, Class interfaces	4
3. Abstract data types, abstract classes	2
4. Polymorphism, templates	5
5. Trees and performance analysis Mathematical concepts, binary trees traversals, expression trees, inserting nodes, analysis	6
6. Searching	4

Analysis of tree searching, AVL, RB

7. Advanced Data Structures 6
   Hashing, heaps, heap sort and heap sort analysis, priority queues
8. Graphs 12
   Mathematical concepts, breadth first search, searching for paths, Prim, shortest path algorithms, minimum weight spanning Trees, Euler paths and circuits, Hamilton paths and circuits, Analysis of algorithms
9. Exams 2
10. Review, evaluations, course admin 2

Total: 44

Laboratory Projects (specify number of weeks on each): Individual lab programming exercises are conducted to emphasize and reinforce class lectures. Each line represents one week.

1. Program performance (Big O) review
2. Inheritance
3. Abstraction
4. Exception Handling and software reuse
5. Binary Trees
6. Binary Tree insertion and deletion
7. BST vs. AVL/RB searches
8. Hashing
9. Heaps and heap sort
10. Priority queues
11. Breadth first vs. searching for path
12. Prim
13. Dijkstra
14. Hamilton paths

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure	2	
Algorithms	1	
Software Design		
Computer Organization and Architecture		
Concepts of Programming Languages	1	
Total	4	0

Oral and Written Communications:
Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral). Equity of computer access based on gender, class, disabled user access. (1 hour). A question on computer ethics is included on one of the exams.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage: Data structure performance analysis, 5 hours

Analysis:
Please describe the analysis experiences common to all course sections.
The course will examine the analysis of time and space complexity of various operations on advance data structures.

Design:
Please describe the design experiences common to all course sections.
Prior to implementing their solutions to laboratories and projects students are encouraged to formulate models that focus on the solution domain. Object-oriented design is emphasized.

SI333 Algorithms and Functional Languages

Catalog: SI333 Algorithms and Functional Languages (4-0-4). Presents techniques for designing and analyzing computer algorithms including divide and conquer, dynamic programming and greedy methods. Introduces classic algorithms for problems such as searching and sorting, graph analysis, file compression and cryptology. *Prereq: SI321.*

Textbook: Introduction to Algorithms, Cormen, Leiserson, Rivest and Stein, McGraw Hill, 2001.

References: None

Coordinator: Chris Brown, Ph.D., Assistant Professor

Program Outcomes:
1. Be able to solve problems using the procedural, functional, and object-oriented programming paradigms.
2. Differentiate amongst the classic algorithms and data structures.
3. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To introduce Midshipmen to the paradigm of Functional Programming, including familiarity with and significant programming experience in at least one functional language.
2. To familiarize Midshipmen with a variety of standard algorithms – the "classics".
3. To provide Midshipmen with the skills needed to compare and analyze the performance of algorithms.
4. To enable Midshipmen to employ a variety of standard techniques to devise efficient algorithms on their own.

Prerequisites by Topic:
1. Stacks, queues and trees
2. Searching and sorting
3. Induction
4. Sets
5. Graph theory

Major Topics Covered:	**Hours**
1. SEIofC: Legal and ethical use of encryption technology	1
2. Functional programming	12
3. Analysis of algorithms	6
- Asymptotic analysis & notation, analysis of simple iterative programs, analysis of recursive programs, recurrence relations, analysis in the presence of functional composition	
4. Divide and Conquer	6
- Application of paradigm and analysis with respect to algorithms such as	

binary search, divide and conquer exponentiation, mergesort, quicksort, quick-selection, Karatsuba's algorithm, Strassen's algorithm

5. Memoization and Dynamic Programming 4
   - Application of paradigm and analysis with respect to algorithms such as matrix-chain multiplication, longest common subsequence, optimal binary search trees, assembly-line scheduling
6. Greedy Algorithms 6
   - Application of paradigm and analysis with respect to algorithms such as activity selection, knapsack problems, Huffman encoding, various graph algorithms, knapsack-based public key cryptography
7. P, NP and NP-Completeness 6
   - Problem encodings and input size, time complexity and complexity classes P and NP, reduction, NP-completeness, a look at NP-complete problems such as SAT, 3-SAT, HAMCYCLE, 0/1 Knapsack, TSP, CIRCUITSAT
8. Number Theoretic Algorithms and Cryptography 4
   - Euclidean GCD, extended GCD, modular arithmetic, modular exponentiation, RSA cryptography, complexity and security
9. Amortized Analysis 4
   - Application of paradigm and analysis with respect to algorithms/data structures such as binary counter increment, multipop stacks, extensible arrays
10. Advanced Topics 5
    - Topics such as parallel algorithms, computational geometry, mathematical algorithms, approximation algorithms
11. Exams 2
12. Review, evaluations, course admin 2

Total: 58

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms	3	
Software Design		
Computer Organization and Architecture		
Concepts of Programming Languages	1	
Total	4	0

Oral and Written Communications:
Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations). Legal and ethical use of encryption technology. Coverage of the ethics requirement centered on encryption technology import/export laws. (1 hour) Students will have to read the commerce department's web-site info on the topic and write a 1-page paper on the topic.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage.
1. Asymptotic analysis, 6 hours
2. Complexity classes, P, NP and NP completeness – 6 hours

Analysis:
Please describe the analysis experiences common to all course sections. The primary focus of the course is on the analysis of time and space complexity of algorithms, and the use of such analysis to provide guidance in designing efficient algorithms and data structures.

Design:
Please describe the design experiences common to all course sections. The design of efficient algorithms and data structures is a unifying thread for most of this course. Students apply some common paradigms for the construction of efficient algorithms to a variety of problems.

SI334 Software Engineering

Catalog: SI334 Software Engineering (4-0-4). An introduction to the basic principles of software engineering. Structured, object-oriented, and formal approaches are studied, with an emphasis on life cycles, object-oriented techniques and team-oriented software development. *Prereq: SI321, SI411.*

Textbook: Classical and Object-Oriented Software Engineering with UML and Java, Fifth Edition, Schach, McGraw-Hill, 2001. Java, How to Program, Fourth Edition, Deitel & Deitel, Prentice Hall, 2001.

References: None.

Coordinator: Donald M. Needham, Ph.D., Associate Professor

Program Outcomes:
1. Apply the basic principles or requirements gathering, specification, design, implementation, and testing to solve problems and make effective use of the foundational core.
2. Can communicate effectively orally, in writing, and via multimedia in both a technical and non-technical manner.
3. Can collaborate effectively in a team environment.
4. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. Introductory software engineering course (with Java) that presents the software development life-cycle and methodologies for dealing with each life-cycle phase.
2. Introduce the latest trends in large-scale software development.
3. Apply software development principles to a large-scale group design/development project.

Prerequisites by Topic:
1. Basic software development techniques to include design, coding, debugging, and documentation
2. Problem solving methods and algorithmic development
3. Data structures
4. File input/output
5. Dynamic memory allocation.
6. Exception Handling
7. Concurrent programming

Major Topics Covered:	**Hours**
1. SEIofC: ACM/IEEE's Software Engineering Code of Ethics	3
2. Fundamental Problem-Solving Concepts: Environments, components, interfaces, classes, inheritance, streams, GUI development, event control and processing, communication and collaboration between objects and	8

software development related topics are covered by introducing Java.

3.	Software Development Process: Software life-cycle models, specification and design tools, software design objectives, documentation, configuration management, reliability, safety, and risk assessment. Software development metrics.	6
4.	Requirements and Analysis: Determining software requirements proto-typing user interfaces. Structured vs. object oriented analysis, use-case modeling, scenarios, class modeling, noun extraction, CRC cards, dynamic modeling, state diagrams.	10
5.	Specification. Structured and object-oriented software specification techniques, informal, semi-formal, and formal specification tools, natural language, data flow diagrams, entity relationship diagrams, and finite state machines.	6
6.	Design: Object-oriented design techniques, interaction diagrams, collaboration diagrams, detailed class diagrams.	8
7.	Verification and Validation. Verification and validation techniques (pre/post-conditions, invariants, proofs of correctness), code and design reviews. Verification as part of the incremental design process.	7
8.	Testing, and Maintenance, structured code walk-throughs, test plan development, glass/blackbox testing, unit and integration testing, regression testing, test case design, and acceptance testing), and maintenance activities.	6
10.	Exams	2
11.	Review, evaluations, course admin	2
Total:		**58**

Laboratory Projects (specify number of weeks on each): The following are covered during the course of each student's involvement in their group-oriented software development project.

1. GUI development, event control and processing (1 week)
2. Multiple document interfaces (1 week)
3. Requirements Analysis/Rapid Prototyping (1 week)
4. Specification (1 week)
5. Object-Oriented Analysis (1 week)
6. Object-Oriented Design (1 week)
7. Integration (1 week)
8. Testing (1 week)

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		
Software Design		4
Computer Organization and Architecture		
Concepts of Programming Languages		
Total	0	4

Oral and Written Communications:
Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy. Each team is required to submit at least four written reports (not including exams, tests, quizzes, or commented programs) of typically eight pages and to make at least four oral presentations of typically 15 minutes duration. Each member of the team must participate in the presentations and contribute to the reports.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? ACM/IEEE's Software Engineering Code of Ethics - Class coverage of the material includes each team orally presenting two of the eight ethics topics, as well as exam coverage.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. N/A.

Analysis:
Please describe the analysis experiences common to all course sections.
Students cover software requirements elicitation and analysis during the course of their group project development.

Design:
Please describe the design experiences common to all course sections.
Students develop both a high-level design and a detailed design during the course of their group software development project.

SI411 Operating Systems

Catalog: SI411 Operating Systems (3-0-3). The study of the operating system as a resource manager. Topics include process management, interrupt processing, memory management, deadlock handling, file systems, multiprogramming, multiprocessing, data security and protection. *Prereq: SI221, SI232*

Textbook: Operating Systems Concepts, Fifth Edition, Silberschatz & Galvin, Addison-Wesley, 1998.

References: None.

Coordinator: Margaret McMahon, Ph.D., Assistant Professor

Program Outcomes:
1. Describe the fundamentals of computer networks, architecture, and operating systems.
2. Analyze and assess the computing environment foundational concepts with respect to security, performance, and evolving technologies.
3. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To develop an understanding of the organization and architecture of computer systems at the programming level of system description.
2. To introduce operating system principles.
3. To introduce the principles of distributed and concurrent systems.

Prerequisites by Topic:
1. Stacks and queues
2. Computer organization

Major Topics Covered:	**Hours**
1. SEIofC: privacy (surveillance & monitoring) and security	1
2. Systems overview	2
3. Operating system structures	3
4. Concurrent processing/threads	6
5. Process management and CPU scheduling	7
6. Deadlocks	4
7. Memory management and virtual memory	8
8. File systems	3
9. Secondary storage/disk management	2
10. Protection	2
11. Networking & distributed systems	2
12. Exams	2
13. Review, evaluations, course admin	2
Total:	**44**

Laboratory Projects (specify number of weeks on each): Individual programming exercises are conducted to emphasize and reinforce class lectures. Projects will involve modifying and writing various segments of an operating system. There are four laboratory assignments in this course and the students ere given two weeks for each one.

1. Lab1: Unix system calls
2. Lab2: I/O buffering
3. Lab3: Threads
4. Lab4: Semaphores

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		2
Software Design		
Computer Organization and Architecture		
Concepts of Programming Languages		1
Total	0	3

Oral and Written Communications:
Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? Computer ethics is discussed in terms of operating systems security and the necessity of maintaining privileged instructions and maintaining critical data. Privacy is discussed in terms of monitoring users, etc. Traditionally, homework and exams contain questions on these topics.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. N/A.

Analysis:
Please describe the analysis experiences common to all course sections. The students analyze the efficiency of the major CPU and disk scheduling algorithms.

Design:

Please describe the design experiences common to all course sections. The major design component occurs in the first laboratory in which the student must develop a timer that tracks CPU and user usage.

SI412 Compiler Construction

Catalog: SI412 Compiler Construction (3-0-3). Using formal programming language theory as a foundation, the issues of automata theory, program optimization and language translation are examined. Sections of actual compilers are constructed to study the practical application of theory presented. *Prereq: SI232, SI321, SI472.*

Textbook: Modern Compiler Implementation in C, A. W. Appel, Cambridge University Press, 1998.

References: A Retargetable C Compiler: Design and Implementation, Fraser, et al., Addison-Wesley, 1995. Lex and Yacc. J. R. Levine, T. Mason, and D. Brown, O'Reilly, 1995.

Coordinator: TBD

Program Outcomes:

Course Goals:
1. To understand the concepts of interpretation and compilation and be able to describe the general structure of a compiler and an interpreter.
2. To develop scanners that can translate, lexically analyze, and tokenize expressions using finite automata.
3. To understand and be able to develop standard classes of BNF grammars and parsers for those grammars.
4. To understand and be able to develop routines for semantic processing.
5. To understand and develop concepts of symbol table management and storage management including heaps and run time stacks.
6. To understand and develop code generators and local and global optimizers.
7. To develop a functional compiler that includes management of a run time stack and a heap.
7. To know and be able to use compiler design tools such as scan generators, parse generators, and optimizers.

Prerequisites by Topic:
1. Theory of computing: Deterministic finite automata.
2. Theory of computing & Programming languages: Formal languages and regular expressions.
3. Programming languages: runtime stack organization, activation records, stack frames, scope and extent, static and dynamic allocation policies, etc.
4. Data structures: stacks, queues, hash tables, memory management and heaps.
5. Assembly language and instruction sets.
6. Basic computer architecture: registers, CPU, memory, etc.

Major Topics Covered:	**Hours**
1. SEIofC: Implications of compiler errors	1

2. Interpretation & compilation	2
3. Lexical analysis	6
4. Parsing	6
5. Abstract Syntax	4
6. Type Checking	5
7. Activation Records	3
8. Translations to Intermediate Code	3
9. Basic Blocks and Traces	3
10. Instruction Selection	3
11. Liveness Analysis	3
12. Register allocation	3
13. Putting it all together – full compiler	1
14. Evaluations, course admin	1
Total:	**44**

Laboratory Projects (specify number of weeks on each): Labs involve modifying and writing various segments of a compiler following the same sequence as the topical list.

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		1
Algorithms		0.5
Software Design		0.5
Computer Organization and Architecture		0.5
Concepts of Programming Languages		0.5
Total	0	3

Oral and Written Communications:
Every student is required to submit at least one written report (not including exams, tests, quizzes, or commented programs) of typically 50-80 pages (every student is required to submit a compiler with extensive annotations and documentation. This final submission is typically 50-80 pages in length) and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? One hour is spent discussing the implications of compiler design on the safety and precision of the resulting programs.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. These topics are woven into the entire fabric of the course.

1. Theory of computing – finite automata (3 hours)
2. Formal grammars and parsers (6 hours)
3. Language specification (6 hours)
4. Assemblers, interpreters and virtual machines (6 hours)

Analysis:
Please describe the analysis experiences common to all course sections. Students are required to analyze an extensive specification for a compiler, a target machine (instruction set) and a complete language definition.

Design:
Please describe the design experiences common to all course sections. Based on this extensive analysis, the student designs and then implements a compiler including both the front and back ends.

SI420 Artificial Intelligence

Catalog: SI420 Artificial Intelligence (3-0-3). A study of the fundamental concepts and techniques in the design and implementation of functionally intelligent machines. Topics include problem-solving using state-space search, problem-reduction techniques, game trees, general problem solver, knowledge representation using production systems and first-order predicate calculus. *Prereq: SI333, SM230.*

Textbook: Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Prentice Hall, 1995.

References: None

Coordinator: Ric Crabbe, Ph.D., Assistant Professor

Program Outcomes:
1. Apply the basic principles of requirements gathering, specification, design, implementation, and testing to solve problems and make effective use of the foundational core.
2. Can communicate effectively orally, in writing, and via multimedia in both a technical and non-technical manner.
3. Can collaborate effectively in a team environment.
4. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To introduce the basic concepts and techniques of artificial intelligence, Lisp programming, knowledge-based systems design and neural architectures.

Prerequisites by Topic:
1. Programming
2. Basics of discrete math
3. Design and analysis of basic data structures and algorithms
4. Understanding of probability

Major Topics Covered:	**Hours**
1. SEIofC: reliability of expert systems	1
2. State space search	4
3. Heuristic search	4
4. The predicate calculus	5
5. Review of Common Lisp	2
6. Advanced topics such as:	
a. Game playing	3
b. Knowledge representation	4
c. Knowledge-based systems	9
d. Machine learning and Neural Networks	5
e. Uncertainty	3

7. Exams 2
8. Review, evaluations, course admin 2

Total : 44

Laboratory Projects (specify number of weeks on each): Individual programming exercises are conducted to emphasize and reinforce class lectures.

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		2
Software Design		
Computer Organization and Architecture		
Concepts of Programming Languages		1
Total	0	3

Oral and Written Communications:
Every student is required to submit at least one written report (not including exams, tests, quizzes, or commented programs) of typically two pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? One hour is spent on ethical issues involved with expert systems. Students are required to write an essay to demonstrate their understanding of these issues.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. The course is about half theoretical in content, including such topics as: state spaces and the properties of various informed and blind search strategies, formal logic and inference procedures, information theory in the context of learning, neural networks, complexity analysis, and computability. See Major Topics Covered above for breakdown of time devoted to each area.

Analysis:
Please describe the analysis experiences common to all course sections. Algorithms and problem domains are analyzed both in theory and in practice.

Design:

Please describe the design experiences common to all course sections. The design process is fundamental to several course activities from data representation to the creation of algorithms and software production.

SI430 Fundamentals of Microcomputer Systems

Catalog: SI430 Fundamentals of Microcomputer Systems (2-2-3). Analysis and design of software systems for microprocessors. Includes characteristics and organization of microprocessors, peripheral interface software and applications of software design. *Prereq: SI321, SI232.*

Textbook: PC Intern (System Programming), Tischer, Abacus, 1992

References: None

Coordinator: None

Course Goals:
1. To study a typical microcomputer system at the hardware and system software levels.

Prerequisites by Topic: N/A

Major Topics Covered:	**Hours**
1. SEIofC: 1	
2. DOS, interrupts, interrupt handlers: BIOS and DOS	2
3. Disk structure: boot sector, BPB, FAT, root directory	5
4. Hard disk structure: partition table	2
5. Keyboard basics: scan codes	2
6. Video basics: text and graphics modes	4
7. The parallel and serial ports: RS-232 communications	5
8. Sound generation and the tick timer	1
9. Memory structure: conventional, HMA, EMS, XMS	3
10. Exams	2
11. Review, evaluations, course admin	2
Total :	**29**

Laboratory Projects (specify number of weeks on each): Approximately 29 hours of "closed" microcomputer laboratory time will be used to reinforce class lectures.

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		
Software Design		
Computer Organization and Architecture		1
Concepts of Programming Languages		2
Total	0	3

Oral and Written Communications:
Every student is required to submit at least N/A written reports (not including exams, tests, quizzes, or commented programs) of typically N/A pages and to make N/A oral presentations of typically N/A minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? N/A.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. N/A.

Analysis:
Please describe the analysis experiences common to all course sections. N/A.

Design:
Please describe the design experiences common to all course sections. N/A.

SI435 Advanced Software Engineering

Catalog: SI435 Advanced Software Engineering (2-2-3). This course will present the latest trends in modern techniques and methods for large-scale software development activities, such as object-oriented programming. The use of CASE tools and group design projects is stressed. *Prereq: SI334*

Textbook: The Unified Software Development Process, I.J. Jacobson, G. Booch, and J. Rumbaugh, Addison-Wesley, 1999. UML Toolkit, H. Eriksson, and M. Penker, Wiley, 1998. Rapid Development, S. McConnell, Microsoft Press, 1996.

References: Patterns in Java Volume I, M. Grand, Wiley, 1998. A Discipline For Software Engineering, W. Humphrey, Addison-Wesley, 1995. The Java (TM) Tutorial Continued, Campione, etal, Addison-Wesley, 1998.

Coordinator: Donald Needham, Ph.D., Associate Professor

Program Outcomes:
1. Apply the basic principles of requirements gathering, specification, design, implementation, and testing to solve problems and make effective use of the foundational core.
2. Can communicate effectively orally, in writing, and via multimedia in both a technical and non-technical manner.
3. Can collaborate effectively in a team environment.
4. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. Develop a substantial software application using the major aspects of software development: definition/requirements, analysis, planning, design, implementation, testing and deployment.
2. Understand advanced concepts in O-O analysis, planning, design, implementation, testing and deployment of software.
3. Learn to use tools for analysis, design and implementation.

Prerequisites by Topic:
1. Fundamental problem solving using a high level O-O language
2. Data structures and algorithms
3. Software development process
4. Concepts for writing requirements and specifications and doing analysis
5. Concepts and approaches for software design and implementation
6. Verification and validation approaches
7. Testing techniques
8. Maintenance activities

Major Topics Covered:	**Hours**
1. SEIofC: professionalism	1
2. Problem and requirements definition	2
3. Defining software architectures	2
4. Analysis using the UML and use cases	5
5. Planning and risk analysis	2
6. Design using patterns	6
7. Implementation	2
8. Testing	4
9. Deployment	1
10. Exams	2
11. Review, evaluations, course admin	2
Total:	**29**

Laboratory Projects (specify number of weeks on each): Team oriented lab assignments focusing on specification, design, integration and testing are conducted to emphasize and reinforce class lectures.

1. Risk Analysis (1 week)
2. Requirements Engineering and Analysis (1 week)
3. GUI-oriented CASE-Tools such as Net-Beans (1 week)
4. Rapid Prototyping (1 week)
5. Acceptance Testing/Test Plan Development (1 week)
6. Design-oriented CASE-Tools such as TogetherJ and UMLet (1 week)
7. Software Architecture (1 week)
8. Connections to databases such as Java Data Base Connectivity (JDBC) (1 week)
9. Application Programming Interfaces such as the Java Media Framework (JMF) (1 week)
10. System Integration/Testing (1 week)
11. Current Trends in Software Engineering such as branded interface toolkits, software reuse, frameworks, advanced exception handling mechanisms, large-scale software development, security aware software, reverse engineering. (4 weeks)

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		0.5
Algorithms		0.5
Software Design		2
Computer Organization and Architecture		
Concepts of Programming Languages		
Total	0	3

Oral and Written Communications:
Every student team is required to submit at least one written report (an analysis document, design document, documentation and a users manual) with their product (not

including exams, tests, quizzes, or commented programs) of typically 20-40 pages and to make one oral presentation of typically 30 minutes duration. Each member of the team must participate in the presentations and contribute to the reports. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? Ethical issues are integrated throughout the course. Ethical and legal issues pertinent to the development and deployment of software are an integral part of the development effort. Topics discussed include understanding licensing agreements, liability for the software engineer and contracts.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. N/A.

Analysis:
Please describe the analysis experiences common to all course sections. Students do problem analysis using use cases and the UML to develop requirements and specifications for a project. During the planning phase, they do risk analysis.

Design:
Please describe the design experiences common to all course sections. Students do software design using the UML and integrated development environments.

SI440 Database Organization

Catalog: SI440 Database Organization (3-0-3). Topics include database systems architecture, the various approaches to database organization including relational, hierarchical and network models; normalization and implementation issues.
Coreq: SI321

Textbook: Fundamentals of Database Systems, Fourth Edition, Elmasri and Navathe, Addison Wesley.

Coordinator: Thomas R. Hendricks, NSA Visiting Professor

Program Outcomes:
1. Apply the basic principles of requirements gathering, specification, design, implementation, and testing to solve problems and make effective use of the foundational core.
2. Can communicate effectively orally, in writing, and via multimedia in both a technical and non-technical manner.
3. Can collaborate effectively in a team environment.
4. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To develop an understanding of various physical level data file organizations. To introduce the concept of data modeling.
2. To introduce the concepts and structures necessary to design and implement database systems.
3. To discuss integrity, security, optimization, concurrency, and distributivity in database systems.

Prerequisites by Topic:
1. Analysis of Algorithms: complexity & efficiency
2. Generalized list structures
3. Binary trees and tree transversals
4. Searching & internal sorting
5. Advanced tree structures: AVL, heaps
6. B-Trees
7. External sorting and file management
8. Recursive definitions and recurrence relations
9. Countability, sets, functions, relations: properties, closures, partial orders, equivalence relations

Major Topics Covered:	**Hours**
1. SEIofC: privacy, sensitive data	1
2. Introduction to database concepts	3
3. Physical level data organization	5

4. Conceptual data modeling, ER diagrams	5
5. Hierarchical, network database models	1
6. Relational database model, relational algebra	6
7. Query languages	3
8. Database design, functional dependencies normalization	6
9. Query optimization	2
10. Integrity, security	2
11. Concurrency	2
12. Distributed database systems	2
13. Current developments in database systems	1
14. Exams	3
15. Review, examinations, course administration	2
Total:	**44**

Laboratory Projects (specify number of weeks on each): Students will design a project database to emphasize and reinforce class lectures.

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		1
Algorithms		1
Software Design		1
Computer Organization and Architecture		
Concepts of Programming Languages		
Total	0	3

Oral and Written Communications:
Every student is required to submit at least one written report (not including exams, tests, quizzes, or commented programs) of typically ten pages and to make one oral presentation of typically ten minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
One hour is spent in class discussing relevant issues and current cases involving dissemination of data privacy and sensitivity.

Theoretical Content: N/A.

Analysis: N/A.
Design: N/A.

SI452 Advanced Computer Architecture

Catalog: SI452 Advanced Computer Architecture (3-0-3). This course provides an advanced study of the design and evaluation of high performance computer systems. *Prereq: SI232.*

Textbook: Computer Architecture: A Quantitative Approach, Second Edition, John L. Hennessy and David A. Patterson, Morgan Kaufmann Publishers, 1995.

References: None

Coordinator: Margaret M. McMahon, Ph.D., Assistant Professor

Program Outcomes:
1. Describe the fundamentals of computer networks, architecture, and operating systems.
2. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To present concepts and theories related to high-performance computer systems, and to build an understanding of the planning, design, construction, and use of such systems.
2. To introduce strategies needed to critically evaluate the designs of high-performance Computer systems.

Prerequisites by Topic:
1. Digital logic
2. Gates, combinational circuits, sequential circuits
3. Registers, register transfer operations, shifters, counters, multipliers
4. Memories
5. Bus structures
6. Timing signals
7. Data control/transfer
8. Instruction types
9. Addressing
10. Control unit
11. Fetch/execute cycle
12. Hardwired control and microprogrammed control
13. Data representations
14. Basic machine organization
15. Operating systems
16. I/O programming
17. Interrupts
18. Macros.

Major Topics Covered: **Hours**

1. SEIofC: Responsibilities of computer system designers, manufacturers and consumers 1
2. Instruction set architecture, RISC, performance measures 6
3. Input/output devices and interfaces: bus structures, program controlled and memory mapped I/O, interrupt-drive I/O and DMA 6
4. Cache memory: structure, design, analysis, replacement policies, modeling performance and performance metrics 9
5. Memory and virtual memory: structure, mapping replacement algorithms 3
6. Pipeline principles: design and control, forwarding and scheduling, performance analysis 11
7. Exploiting pipeline techniques: loop unrolling, superscalar and VLIW architecture 4
8. Exams 2
9. Review, evaluations, course admin 2

Total: 44

Laboratory Projects (specify number of weeks on each): None

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		1
Software Design		
Computer Organization and Architecture		2
Concepts of Programming Languages		
Total	0	3

Oral and Written Communications:
Every student is required to submit at least two written reports (not including exams, tests, quizzes, or commented programs) of typically five pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? One hour is spent in a class discussion about ethical issues in computer architecture. A short paper is assigned in which the student addresses current ethical issues including responsibilities of computer system designers, manufacturers and consumers.

Theoretical Content:

Please list the types of theoretical material covered, and estimate the time devoted to such coverage. Approximately one-third of the course is an introduction to the algorithms relevant to computer architecture.

Analysis:
Please describe the analysis experiences common to all course sections. Algorithms considering the operating, updating and communication between computer components are analyzed. Techniques are developed to quantitatively describe computer performance. Students assess the influence of design decisions on the performance of a computer system.

Design:
Please describe the design experiences common to all course sections. Students participate in design exercises in class and in written homework assignments.

SI454 Computer Networks

Catalog: SI454 Computer Networks (3-0-3). The course presents major topics in the area of computer networks. It views a computer network as a group of related layers or abstract machines as exemplified by the International Standards Organization (ISO) network reference model and Internet Layer Model. The course presents the basic concepts necessary to understand the design and operation of computer communication networks. *Prereq: SI232, SI321, SM230C.*

Textbook: Computer Networks and Internets, Third Edition, Comer, Prentice Hall, 2001.

References: Internet Engineering Task Force (IETF) Requests for Comment (RFCs)

Coordinator: Margaret McMahon, Ph.D., Assistant Professor

Program Outcomes:
1. Describe the fundamentals of computer networks, architecture, and operating systems.
2. Analyze and assess the computing environment foundational concepts with respect to security, performance, and evolving technologies.
3. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. Present concepts and major topics of data and computer communications and computer networks.
2. Introduce communication networking principles, concepts and techniques.
3. Introduce network architectures and protocols.
4. To develop an awareness of the characteristics of local area networks (LAN).

Prerequisites by Topic:
1. Digital logic
2. Memories
3. Instruction types
4. Addressing
5. Queues, Trees and Graphs

Major Topics Covered:	**Hours**
1. SEIofC: Internet, censorship, flaming	1
2. Intro to Networking	3
3. Packets, Frames and Error Detection	3
4. Network topology, media access and LANs	9
5. Layered models of internetworking	2
6. Internetworking packets and protocols	8
7. Reliable transport	4
8. Routing algorithms	3

9. Application layer protocols: virtual terminal, file transfer and electronic mail 3
10. Introduction to Client-Server programming 4
11. Exams 2
12. Review, evaluations, course admin 2

Total: 44

Laboratory Projects (specify number of weeks on each): None.

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		2
Software Design		
Computer Organization and Architecture		1
Concepts of Programming Languages		
Total	0	3

Oral and Written Communications:
Every student is required to submit at least 1 written report (not including exams, tests, quizzes, or commented programs) of typically 2 pages and to make no oral presentations.

Social and Ethical Issues:
One hour is spent in class discussing relevant issues and current cases involving privacy on the Web, censorship or spamming. A short paper may be assigned addressing current issues in the field.

Theoretical Content:
Approximately one-third of the course is an introduction to the algorithms relevant to network operation and communication paradigms. Their implementation in networks is discussed through the use of examples.

Analysis:
Students participate in group discussions analyzing advantages and disadvantages of network design choices. Techniques are introduced to analyze network behavior. These techniques are reinforced in homework assignments.

Design:
Design issues are highlighted in the course as we progress from network components to common implementations of networks. Consequences of design decisions are explored as network components are pieced together. Exercises in class and in homework assignments reinforce students' development of skills to critique designs.

SI455 Advanced Computer Networks

Catalog: SI455 Advanced Computer Networks (3-0-3). This course provides an in-depth technical study of high-speed networking, client-server programming and applications, network firewall architectures and security procedures, and the ATM network. *Prereq: SI454.*

Textbook: Computer Networks and Internets, Second Edition, Comer, Prentice Hall, 1999. TCP/IP Vol III: Client-Server Programming and Applications, Comer and Stevens, Prentice Hall, 1997.

References: None

Coordinator: Margaret McMahon, Ph.D., Assistant Professor

Program Outcomes:
1. Apply the basic principles of requirements gathering, specification, design, implementation, and testing to solve problems and make effective use of the foundational core.
2. Can communicate effectively orally, in writing, and via multimedia in both a technical and non-technical manner.
3. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. Present concepts involved with high-speed computer networks.
2. Obtain practical experience with client-server programming.
3. Introduce network firewall architectures and security procedures.
4. Discuss ATM Wide Area Network.

Prerequisites by Topic:
1. Data transmission: analog and digital data transmission, transmission media.
2. Digital data communication techniques: asynchronous and synchronous transmission, error detection techniques, interfacing.
3. Data link control: line configurations, flow control, error control, bit-oriented link control.
4. Communication networking techniques: circuit switching, message switching, packet switching broadcast networks.
5. Packet switching: virtual circuits and datagrams, routing, traffic control, error control.
6. Local area networks: bus/tree/ring topology, medium access control protocols, performance.
7. Computer communications architecture: the OSI model.

8. Internetworking: bridge/router/gateway, connection-oriented and connectionless Internetworking.

Major Topics Covered: **Hours**

1. SEIofC: privacy, censorship, unauthorized use 1
2. TCP /IP Review: protocol architecture, addressing, routing, ports, services 3
3. Client-Server: software organization, design issues, concurrent, client-server execution tunneling, socket API, Remote procedure calls 19
4. Network Security Architectures, Issues, and Approaches 6
5. ATM: cells, adaptation layers, virtual paths, and channels, addressing quality of service, LAN Emulation, deployment strategies 3
6. Student Presentations of Advanced Network Topics 8
7. Exams 2
8. Review, evaluations, course admin 2

Total: 44

Laboratory Projects (specify number of weeks on each): None

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		0.5
Algorithms		1
Software Design		1
Computer Organization and Architecture		
Concepts of Programming Languages		0.5
Total	0	3

Oral and Written Communications:
Every student is required to submit at least one written report (not including exams, tests, quizzes, or commented programs) of typically 2-5 pages and to make one oral presentations of typically 5-10 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? Each student is required to hand in a written summary of an article addressing some aspect of network ethics such as Internet, censorship, flaming.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. None.

Analysis:
Please describe the analysis experiences common to all course sections. Performance analysis involving timing of communications.

Design:
Please describe the design experiences common to all course sections. Design is covered by using COMNET III tutorials. Software design is also done through programming projects.

SI457 Information Assurance (2-2-3)

Catalog: SI457 Information Assurance (2-2-3). The course provides an introduction to the theoretical and practical facets of Information Assurance including: secure operating systems, public key cryptography, system and network security, and offensive and defensive information warfare operations. Laboratory work will include student exercises demonstrating information assurance concepts. *Prereq: SI411, SI454.*

Textbook: Introduction to Computer Security, Matt Bishop, ISBN 0-321-24744-2; Publisher Addison Wesley Professional; Copyright 2005

Coordinator: Thomas R. Hendricks, NSA Visiting Professor

Program Outcomes:
1. Analyze and assess the computing environment foundational concepts with respect to security, performance, and evolving technologies.
2. Can collaborate effectively in a team environment.
3. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. Describe the fundamental encryption principles, including: symmetric, asymmetric, and hybrid algorithms, cryptography; and message authentication.
2. Apply system security techniques, detection and protection strategies.
3. Assemble a secure network using Intrusion Detection Systems and Firewalls.
4. Manage the protection mechanisms necessary to provide network and web security.

Prerequisites by Topic:
1. Network Protocols TCP/IP (SI454).
2. Operating Systems Modes of Operations (SI411).
3. Client/Servers (SI454).

Major Topics Covered:	**Hours**
1. Ethics: Viruses	1
2. Intro to Information Assurance theory and Information Protection Models	2
3. Encryption Principles, Algorithms, Modes of Operation.	2
4. Principles of Cryptography and Cryptographic Algorithms	3
5. Authentication Applications	2
6. Network and Internet Security	3
7. Applications Security: Electronic Mail and Web	2
8. Malicious Software	2
9. Firewalls – Principles, Characteristics, Configurations, Trusted Systems	3
10. Intruders and Intrusion Detection Systems	2
11. Security Evaluations and Vulnerability Assessments	3
12. Exams	2
13. Review and Course Administration	2
Total:	**29**

Laboratory Projects (specify number of weeks on each):	**Weeks**
1. Mandatory Access Control (MAC)/Discretionary Access Control (DAC) and System Administration	1
2. Spyware: Registry Changes	1
3. Public Key Infrastructure/Certifications	2
4. Virtual Private Networks (VPN)/Secure Sockets Layer (SSL)	1
5. Network Setup: troubleshooting using Ethereal	1
6. Network Mapping: Network Mapper (NMAP)	1
7. Domain Name Server (DNS) installation and table maintenance	1
8. Email proxy services: Sendmail	1
9. Firewalls and Internet Protocol (IP) Tables	1
10. Intrusion Detection Systems: SNORT	1
11. Malicious Software: Sub-Seven and Back Orifice	1
12. Cyber Defense Exercise	3

Oral and Written Communications:
Every student is required to submit at least 1 written report (not including exams, tests, quizzes, or commented programs) of typically 2 pages and makes no oral presentations.

Social and Ethical Issues:
One hour is spent in class discussing relevant issues and current cases involving dissemination of computer malicious software.

Theoretical Content:
Approximately one-third of the course covers algorithms relevant to encryption and secure networking protocols. Their implementation in networks is further discussed through the use of examples.

Analysis:
Students participate in laboratory exercises requiring the implementation of computer services, which are frequently the target of hostile exploits. Students analyze the advantages and disadvantages of network design and security implementation choices.

Design:
Design issues are highlighted in the course as we progress from security components to common implementation of networked systems. The concept of a “layered” security defense is developed throughout the course. Consequences of design decisions are explored as network components are pieced together.

SI460 Computer Graphics

Catalog: SI460 Computer Graphics (2-2-3). Presents the concepts, theories and algorithms related to computer graphics. Builds understanding in evaluating and implementing graphical support in various application settings. *Prereq: SI321*

Textbook: Interactive Computer Graphics, A Top-Down Approach Using OpenGL™, Third Edition, Angel, Addison Wesley 2003.

Coordinator: David J. Stahl. Jr., Ph.D., Assistant Professor

Program Outcomes:
1. Apply the basic principles of requirements gathering, specification, design, implementation, and testing to solve problems and make effective use of the foundational core.
2. Can communicate effectively orally, in writing, and via multimedia in both a technical and non-technical manner.
3. Can collaborate effectively in a team environment.
4. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. Understand how images are displayed on a raster graphics device.
2. Be able to use a graphics API to create simple 2D and 3D interactive graphics applications.

Prerequisites by Topic:
1. Structured programming (SI321)
2. Vectors and matrix theory (SI262)

Major Topics Covered:	**Hours:**
1. Graphics primitives: points, lines, polygons, curves, text, clipping	5.5
2. Coordinate systems, modeling, viewing, surfaces, hidden surfaces, orientation	8
3. Illumination & shading, texture, blending	5.5
4. Graphics API, event-driven programming, picking	6
5. Social and Ethical Issues of Computing: digital imagery	1
6. Exams, review, evaluations	3
Total:	**29**

Laboratory Projects:	**Weeks:**
1. Drawing points, lines, polygons, curves, text	4
2. Modeling and viewing 2D & 3D objects	5
3. Illumination & shading	4
4. User-interaction and event-driven programming	3

Estimate CAC/ABET Category Content:

Area	Core	Advanced

Data Structure		
Algorithms		1
Software Design		1
Computer Organization and Architecture		
Concepts of Programming Languages		1
Total	0	3

Oral and Written Communications:
Each student is required to submit a written report typically of 2-3 pages in length and to make an oral presentation typically of 5-15 minutes duration, describing the approaches taken and lessons learned in designing and implementing the final programming project.

Each student us required to submit a written report typically of 2-3 pages in length discussing the social and ethical issues associated with the use of digital imagery.

Social and Ethical Issues:
The use of digital imagery in areas of public influence such as advertising and election campaigning is investigated and discussed in a paper.

Theoretical Content:
1. Color and human perception 1 hr
2. Representing orientation 2 hr
3. Illumination and shading models 1 hr
4. Planar projections 1 hr
5. Curve and surface representations 3 hr

Problem Analysis: N/A

Solution Design:
Students design and implement graphical applications individually and as a member of a small team.

SI462 Advanced Computer Graphics

Catalog: SI462 Advanced Computer Graphics (2-2-3). A project-based course involving advanced graphics techniques such as ray-tracing, radiosity, volume rendering, virtual and augmented reality, haptics, and pixel shaders. *Prereq: SI460*

Textbook: Advanced Animation and Rendering Techniques, Alan Watt, M. Watt, Addison-Wesley, 1992.

Coordinator: David J. Stahl. Jr., Ph.D., Assistant Professor

Program Outcomes:
1. Apply the basic principles of requirements gathering, specification, design, implementation, and testing to solve problems and make effective use of the foundational core.
2. Can communicate effectively orally, in writing, and via multimedia in both a technical and non-technical manner.
3. Can collaborate effectively in a team environment.
4. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To understand advanced computer graphics concepts, algorithms and techniques.

Prerequisites by Topic:
1. Familiarity with a graphics programming API (SI460)
2. Basic lighting models (SI460)
3. 2D textures (SI460)

Major Topics Covered:	**Hours:**
1. Illumination models (e.g., raytracing, radiosity, shadows)	6
2. Volume rendering (e.g., raycasting, splatting, 3D texture)	5
3. Immersion (e.g., augmented and virtual reality, stereo)	6
4. Physically-based modeling (e.g., spring-mass models, haptics)	3
5. Vertex and pixel shaders (e.g., Cg, OpenGL Shading Language)	1
6. Current, historical, and seminal papers in computer graphics	4
7. Exams, review, evaluations	4
Total:	**29**

Laboratory Projects:	**Weeks:**
1. Ray-tracing	6
2. Volume rendering	4
3. AR, VR, stereo, haptics	4

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		1
Software Design		2
Computer Organization and Architecture		
Concepts of Programming Languages		
Total	0	3

Oral and Written Communications:
Each student is required to submit a written outline of typically 2 pages in length and to make an oral presentation of typically 15 minutes duration, leading an instructor-assisted discussion reviewing a current, historically significant, or seminal paper in the field of computer graphics.

Social and Ethical Issues: None

Theoretical Content:
1. Illumination and shading models 6 hr
2. Physically-based models 3 hr

Problem Analysis: N/A

Solution Design:
Students design and implement graphics applications as a member of a small team.

SI472 Theory of Computing

Catalog: SI472 Theory of Computing (3-0-3). Presents the theoretical foundations for computing, including the study of finite state machines, pushdown automata, and Turing machines. *Prereq: SI321.*

Textbook: An Introduction to Formal Languages and Automata, Third Edition, Linz, Jones and Bartlett, 2001.

References: JFLAP software package from Duke University

Coordinator: Kay G. Schulze, Ph.D., Professor

Program Outcomes:
1. Discuss the theoretical and mathematical foundations of computer science.

Course Goals:
1. To study the various formal models of computation, from both the formal language and the corresponding machine model perspective.
2. To present a general overview of the theory of computability and decidability.

Prerequisites by Topic:
1. Mathematical Proofs
2. Stacks
3. Strings

Major Topics Covered:	**Hours**
1. Mathematical Preliminaries (proofs, languages, sets)	3
2. Regular languages, regular expressions	2
3. DFAs, NFAs, NFA's, minimal DFAs	4
4. Equivalence of DFAs, NFAs, NFA's	2
5. Kleene's theorem	2
6. Myhill-Nerode theorem, the Pumping lemma for regular languages	2
7. Applications of regular languages	1
8. CFLs, CFGs, regular CFGs, Chomsky Normal Form	6
9. PDAs, DPDAs	3
10. Equivalence of CFGs and PDAs	2
11. Ambiguity, parsing, applications of CFGs	3
12. The Pumping Lemma for CFLs	1
13. Turing Machines, variation's of TMs, Universal Turing Machine	5
14. Recursive and partial recursive functions, recursive and recursively enumerable languages	2
15. Church's Thesis, decidable and undecidable problems (the halting problem)	2
16. Exams	2
17. Review, evaluations, course admin	2
Total:	**44**

Laboratory Projects (specify number of weeks on each): Individual programming exercises with a software tool are conducted to emphasize and reinforce class lectures.

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		1
Software Design		
Computer Organization and Architecture		
Concepts of Programming Languages		
Total	0	1

Oral and Written Communications:
Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? N/A.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. This entire course is a theoretical course. See Major Topics Covered section for topics and amount of time devoted to each topic.

Analysis:
Please describe the analysis experiences common to all course sections. The students have to develop the best algorithm to use to develop a theoretical machine for each problem they solve.

Design:
Please describe the design experiences common to all course sections. The students design and then encode, using the JFLAP software package, FSM, PDA, and TM.

SI475 Intelligent Robotics

Catalog: SI475 Intelligent Robotics (2-2-3). Presents the concepts and theories related to computer-driven robotic systems and computer-based vision systems.
Students apply acquired knowledge in a laboratory setting by designing, coding and testing robotic control and vision systems. *Prereq: SI221, Si262, SM230.*

Textbook: None. Course material is provided by instructor.

References: Introduction to Robotics, McKerrow, Addison Wesley, 1991. Fundamentals of Robotics, Schilling, Prentice Hall, 1990. Mobile Robots: Inspiration to Implementation, Jones/Flynn, AK Peters, 1993. Sensors for Mobile Robots, Everett, AK Peters, 1995. Machine Vision, Jain/Kasturi/Schunck, McGraw Hill, 1995. Navigating Mobile Robots (CD-ROM), Borenstein, et al, AK Peters, 1996. Robix RCS-6 – UserGuide and Project Book, Advanced Design Inc., 1995. Rug Warrior Pro Assembly Guide, Jones, AK Peters, 1998.

Coordinator: Ric Crabbe, Ph.D., Assistant Professor

Program Outcomes:
1. Apply the basic principles of requirements gathering, specification, design, implementation, and testing to solve problems and make effective use of the foundational core.
2. Can communicate effectively orally, in writing, and via multimedia in both a technical and non-technical manner.
3. Can collaborate effectively in a team environment.
4. Evaluate contemporary legal, social, and ethical issues in computing professions.

Course Goals:
1. To present concepts and theories related to computer- driven robotics systems and computer-based vision systems.
2. To present methods by which robotics devices use sensor information to learn about their environment, and how they apply AI techniques to this information to plan and accomplish tasks.
3. To present concepts and theories related to processing and analyzing information provided by vision systems.
4. To provide students with an opportunity to apply acquired knowledge in a laboratory environment by designing, coding and debugging robotics control programs using a variety of robotics platforms and sensors.

Prerequisites by Topic:
1. Ability to program in a high-level language
2. Implementation of stack data structure
3. Matrix multiplication
4. Algebraic and geometric problem solving
5. Understanding of probability

Major Topics Covered:	**Hours**
1. SEIofC: Social impact of robotics, software reliability	1
2. Robot Types	1
3. Coordinate Systems	2
4. Direct Kinematics, Homogeneous Transformation	4
5. Sensors: Touch, Light, Infrared, Ultrasonic	1
6. Task Planning	1
7. Behavior Based Robotics	4
8. Planning/Behavior-Based hybrids	2
9. Navigation and Path Planning	3
10. Map making, localization	1
11. Advanced Topics, Vision, Learning, Social Robotics, etc	5
12. Exams	2
13. Review, evaluations, course admin	2
Total:	**29**

Laboratory Projects (specify number of weeks on each): Approximately 29 hours of hands-on experience is provided through use of laboratories equipped with robotics and vision systems.

1. Familiarization and script program development for robot manipulator (Robix RCS-6) (1 week).
2. Integration of robot controller/files with HLL IDE environment (1 week).
3. Mathematical development of manipulator and workspace geometry to solve single-servo fetch placement problem (1 week).
4. Implement transformation-matrix solution to the direct kinematics problem (3 weeks).
5. Implement trigonometric solution to the inverse kinematics problem (3 weeks).
6. Gray-scale image analysis to isolate, locate, and orient multiple objects (3 weeks).
7. Build and interface a mobile robot sensor, then implement behavior-fusion (subsumption architecture) by programming robot to navigate and accomplish a mission (4 weeks).

Estimate CAC/ABET Category Content:

Area	Core	Advanced
Data Structure		
Algorithms		2
Software Design		0.5
Computer Organization and Architecture		0.5

Concepts of Programming Languages		
Total	0	3

Oral and Written Communications:
Every student is required to submit at least one written report (not including exams, tests, quizzes, or commented programs) of typically three pages and to make one oral presentation of typically three minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues:
Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)? Changing attitudes toward warfare as more fighting is done by robots (e.g., cruise missiles, future automatons). Degree and impact of job losses to robots.

Theoretical Content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage. N/A.

Analysis:
Please describe the analysis experiences common to all course sections. Students benefit from seeing a physical manifestation of their program design; algorithm refinement thus proceeds from analysis of resultant physical motion. Students can experiment with an image-thresholding technique of their own design. Visual analysis of the resultant binary image provides the feedback necessary for further algorithm refinement.

Design:
Please describe the design experiences common to all course sections. Students are taught the basic metrics by which a manipulator can be modeled in software. Specifically: link and joint parameters are translated into matrix representations for computer processing. Course provides unique computer science experience in which students are tasked to apply previously learned mathematical and programming principles to solve three-dimensional motion problems.

SI481/2/3/4/5/6 Advanced Topics in Computer Science

Catalog: SI481/2/3/4/5/6 Advanced Topics in Computer Science (X-X-X) (not in catalog; credits vary). Because Computer Science is a dynamic field, an advanced topics course may be offered from time to time covering selected material. In such a course, the selected topic is explored in detail for the more advanced students. *Prereq: Permission of the instructor or the published prerequisite.* SI481/3/5 are Fall offerings. SI482/4/6 are Spring offerings.

Course Goals:
This course is used to run new major electives for the first time. It is also used to run a course on a onetime basis if we have a visiting professor. Traditionally, it is run as (3-0-3) course. This course is also used to provide advanced students with the opportunity for direct study course.

Course History:
1. Spring 02 SI486A: Information Assurance – *Prereq SI454*
2. SI486B: Beowulf Computing – *Prereq SI332*
3. Spring 00 SI486A: Computer and Information Security for the Junior Officer – *Prereq: SI332*
4. Fall 99 SI485A: Firewall Theory and Design – *Prereq: SI332*
5. Spring 99 SI486A: High Speed Networks – *Prereq: SI454*
6. SI486C: Enterprise Computing Using Java – *Prereq: SI434*
7. SI486D: Expert Systems – *Coreq: SI304*

SI495/6 Computer Science Research Project (3-0-3)

Catalog: SI495/6 Computer Science Research Project (3-0-3) (not in catalog). Research projects are reserved for those students who, through experience with advanced courses, have gained an ability to pursue a course of study without a great deal of guidance. This can be shown by independently attempting problems not pursued in existing courses or by pursuing interests outside the realm of material covered in these courses. *Prereq: See Computer Science Instruction 1531.4 series for approval procedure.*

Course Goals:
The basic object of research project activity is to give the student an opportunity to tackle a reasonably large problem from the start. Guidance is given only to the extent that the student is kept from lengthy pursuit of worthless study. Faculty guidance also involves reviewing of periodic reports on student progress.

Course Relation:
SI495/6 are generally taken in the first class year by students who have shown the ability to do outstanding work and work independently.

Service Course Descriptions
SI250 Information Systems for the Junior Officer

Catalog: SI250 Information Systems for the Junior Officer (2-2-3). The primary emphasis of the course is practical applications of personal computers and the Internet in the Fleet/Fleet Marine Force (FMF), with coverage of some special tactical computers as well. Application software is addressed from a junior officer's viewpoint, as an operational unit Branch/Division/Company Officer or as a support staff member. *Prereq: None.*

Textbook: Inline/Online; Fundamentals of the Internet and the World Wide Web, Raymond Greenlaw, Ellen Hepp, WCB McGraw Hill, 1998. Excellent HTML, Timothy Gottleber, Irwin McGraw Hill, 1998.

References: None

Coordinator: TBD

Course Goals:
1. To instruct midshipmen in the effective use of information systems utilizing applications that are relevant to the junior officer in today's Navy and Marine Corps.
2. To instruct midshipmen in the security, privacy, and ethics issues relevant to information systems.
3. To introduce midshipmen to the tactical and administrative support systems currently in use in the Fleet/Fleet Marine Force.

Prerequisites by Topic: None

Major Topics Covered:	**Hours**
1. System Components	4
2. Operating Systems and Utilities	8
3. Application Software	4
4. Web Content Creation (Application Software 2)	8
5. Databases	4
6. Internet Search Strategies (Application Software 2)	8
7. Internet and Networks	4
8. Security and Ethics (Application Software 2)	8
9. Hardware and Upgrades	4
10. Fleet/FMF Systems	8

Laboratory projects (specify number of weeks on each): Weekly labs are held covering the following areas:
1. Getting Started – UNIX, FTP, Account basics
2. E-mail and Web Browser Basics
3. Evaluating Internet Information
4. Introduction to Search Engines

5. Presentation Software
6. Introduction to HTML and Web Publishing
7. Basic HTML
8. Project Presentations
9. Advanced HTML
10. Java and JavaScript
11. Databases
12. Word Processing
13. Navy/Marine Corps Online Resources
14. Spreadsheets

SI283 Programming for Engineers (1-2-2)

Catalog: SI283 Programming for Engineers (1-2-2). An introduction to a structured programming language and its use in implementing algorithms to solve engineering problems. *Prereq: None.*

Textbook: C Program Design for Engineers, J.R. Hanly, E.B. Koffman, and J.C. Horvath, Addison-Wesley, 1995**.**

References: C++ How to Program, Second Edition, Deitel & Deitel, Prentice Hall, 1998.

Coordinator: TBD

Course Goals:
1. To introduce the concepts of structured programming utilizing a high level programming language.
2. To introduce problem solving methods and algorithmic development using engineering applications.

Prerequisites by Topic: None

Major Topics Covered:	**Hours**
1. Basic language constructs: basic data types, variables, assignment, arithmetic expressions, input/output	3
2. Control flow, conditionals, and logical expressions	3
3. Functions, function arguments	2
4. Character and string types	1
5. Arrays, matrices and pointers	3
6. File input and output	2
7. Data visualization using Matlab	1

Laboratory Projects (specify number of weeks on each): Approximately 30 hours of laboratory time will be used to reinforce the concepts discussed in class. These labs consist of engineering problems that use the C constructs taught that week.
1. Program structure, compilation, linking, debugging (1 week).
2. Basic I/O, Simple types and expressions (1 week).
3. Data types, arithmetic operators, math library, precedence/assoc (1 week).
4. Selection structures: if and case (1 week).
5. Repetition and loops: for, while do-while (1 week).
6. User defined functions (1 week).
7. Functions and scope (1 week).
8. Recursion functions and composition (1 week).
9. Declaring, initializing and referencing one-dimensional arrays (1 week).
10. Passing arrays to functions (1 week).
11. Strings and character arrays, file IO (1 week).
12. Multi-dimensional arrays (1 week).

13. Modeling linear equations using Multi-dimensional arrays (2 weeks).
14. MATLAB visualizing data (1 week).

SI305 Computer Programming

Catalog: SI305 Computer Programming (3-0-3). Use of top-down design and software engineering methodology to develop structured algorithms for problem-solving. A high-level language is used to implement the algorithmic designs. (Not for CS credit) *Prereq: None.*

Textbook: Problem Solving and Computation, Steve Lerman, Prentice Hall, 19XX.

References: None.

Coordinator: TBD

Course Goals:

1. This course is designed for those students who are not majoring in computer science but would like to develop programming skills and learn a high level language such as Java, C, or Ada. This course is especially appropriate for those midshipmen dealing with the advanced and necessarily more complex applications required when majoring in engineering and the sciences.
2. Although intended primarily as an elective for scientific majors, it might very well serve any major where knowledge of a high level programming language is considered useful.

Prerequisites by Topic: None

Major Topics Covered:
This course has not run in several years and will probably be eliminated during the curriculum review process.

Laboratory Projects (specify number of weeks on each): Individual programming exercises are conducted to emphasize and reinforce class lectures.