

Department of Computer Science

Web Site: <http://www.cs.odu.edu>

Engineering & Computational Sciences Bldg.
4700 Elkhorn Ave, Suite 3300
Norfolk, VA 23529-0162

Ravi Mulkamala, Department Chair
Yaohang Li, Graduate Program Director
Danella Zhao, Graduate Program Director

Programs

The Department of Computer Science offers programs leading to the Master of Science with a major in computer science, a linked five-year combined BSCS and MS with a major in computer science, and the Doctor of Philosophy in computer science. The Department of Computer Science also offers a Master of Science in computer science with an information and communications technology concentration (jointly with the Information Technology and Decision Sciences Department in the Strome College of Business).

Computer science traces its foundation to mathematics, logic, and engineering. Studies in computer science encompass theory, experimental techniques, and engineering methodology. The computer science curriculum exposes students to aspects of each of these disciplines and fosters an appreciation and understanding of them. Students are exposed to the broad theoretical basis of computer science through lecture and laboratory experience. The Computer Science Department has a unique curricular model that applies computer science education to the real world. In addition, the Computer Science Department offers a set of courses to professionals who need supplementary experience. A graduate of the computer science program will have a broad fundamental knowledge of the field and in-depth knowledge in a particular subject area. To acquire breadth, graduate students in the department are required to take core courses which together with the undergraduate core courses cover major aspects of computers and computation. At the master's level, the department supports in-depth study in the following areas:

- bioinformatics,
- data mining,
- web science and digital libraries,
- high performance computing,
- networking,
- cybersecurity,
- software engineering, and
- computational foundations.

At the PhD level, areas of specialization are limited only by the interests of the available faculty. The department has an excellent state of the art computing facility. Please visit the department's home page for details: <http://www.cs.odu.edu>.

Master of Science - Computer Science

Entrance Requirements

Students entering the Master of Science program in computer science should meet the minimum university graduate admission requirements (<http://www.odu.edu/admission/graduate> (<http://www.odu.edu/admission/graduate/>)). In addition, an applicant must have a strong background in computer science. Students who do not have a sufficient background in computer science may enter the graduate program as provisional students and make up for their deficiencies by taking appropriate courses. Applicants are required to take the GRE general test. For the Information & Communications Technology concentration (described below), the GMAT aptitude test may be used. Two letters of recommendation from faculty members of academic institutions are required in addition to all transcripts at

the postsecondary level. For students whose native language is not English, either a TOEFL score of 550 (paper-based) and 79 (internet-based) or IELTS score of 6.5 is also required.

Requirements

The departmental requirements for the Master's degree are described below. All these requirements must be satisfied in addition to the University requirements outlined under the Academic Information section of this Catalog.

Core Courses

The following core courses are required:

CS 665	Computer Architecture	3
and one of the following:		3
CS 500	Foundations of Computing	
CS 600	Algorithms and Data Structures	
Total Hours		6

MS students in the thesis option are required to take CS 665 and CS 600 to satisfy the core requirement.

Colloquium

Each student is required to take a one-credit CS 690 (Computer Science Colloquium) and attend at least 10 departmental colloquiums during their MS study.

Course Options

Three options are available for candidates for master's degrees:

- thesis option,
- project option, and
- course-only option.

Thesis Option

Course work	24
Thesis research	6
Colloquium	1
Total Hours	31

A minimum of 31 credit hours is required. The candidate is required to write a thesis and make an oral presentation of the results.

Project Option

Course work	30
Project work	3
Colloquium	1
Total Hours	34

A minimum of 34 credit hours is required. The candidate is required to prepare a written report on the project and to present it orally.

Course-Only Option

Course work	33
Colloquium	1
Total Hours	34

A minimum of 34 credit hours is required. In addition, the candidate is required to appear for an exit examination that requires a comprehensive written report and an oral examination.

Course Restrictions

No more than nine credits of the following courses may be counted towards the degree:

CS 697	Independent Study in Computer Science
CS 791	Graduate Seminar
CS 796	Topics in Computer Science

A maximum of four 500-level courses can be applied to the program.

At least three credits counted toward the computer science degree must be taken at the 700-level from courses other than CS 791 and CS 796.

Time Limit

All requirements for the master's degree must be completed within eight years.

Master of Science - Computer Science - Information & Communications Technology Concentration

Requirements

This concentration, offered jointly with the Department of Information Technology and Decision Sciences in the Strome College of Business, is appropriate for students with either a bachelor's degree in business administration with a major in information systems and a computer science minor or with a bachelor's degree in computer science with a business administration minor.

Colloquium

Each student is required to take a one-credit CS 690 (Computer Science Colloquium) and attend at least 10 departmental colloquiums during their MS study.

Course options

Same as described above in the M.S. in Computer Science, with the following conditions:

- Project option: 15 credits (5 courses) must be CS graduate courses, and 15 credits (5 courses) must be IT graduate courses.
- Course-only option: 18 credits (6 courses) must be CS graduate courses, and 15 credits (5 courses) must be IT graduate courses.

There are no core courses required for the ICT concentration.

Course restrictions

Same as described above in the M.S. in Computer Science.

In addition, at least two of the CS graduate courses (6 credits) must be taken at the 600-level or 700-level from courses other than CS 697, CS 791, and CS 796.

Time Limit

All requirements for the master's degree must be completed within eight years.

Linked B.S. and M.S. in Computer Science

This program allows for exceptionally successful students to earn both a B.S. and M.S. in Computer Science by allowing them, under certain conditions, to count up to 12 credits of graduate coursework toward both their bachelor's and master's degree in Computer Science. All options available under the MS program are available under this program. See the Graduate Catalog section on Linked Programs for more information.

Graduate Certificate Program in Cybersecurity (Online)

This online certificate program, which may be completed in one year, is designed to train working professionals (in computer science and related fields) in cyber security. The courses offered under this program may be taken by current students or new students. A student must first apply to the certificate program and be admitted to the program to be eligible to receive the certificate. A student admitted to this program may subsequently apply for the MS in Computer Science. If admitted, the courses taken during the certificate program may be considered towards the MS program.

Candidates for the program must at least have a 4-year bachelor's degree from an accredited academic institution. The program requires a student to take four (or 12 credits) of the following courses. Each course has a normal

letter grade and a student is expected to obtain a C or better in each course and maintain a 3.0 GPA. The following are the current offerings. Additional electives may be added in future.

CS 562	Cybersecurity Fundamentals	3
CS 563	Cryptography for Cybersecurity	3
CS 564	Networked Systems Security	3
CS 565	Information Assurance for Cybersecurity	3
Total Hours		12

Graduate Certificate in Modeling & Simulation - Computing & Informatics

This certificate program has a focus on computational science and informatics disciplines. The required coursework includes introductory topics in fundamental theories and approaches in computer modeling and simulation. The elective courses enable students to develop deeper understanding in computer modeling and simulation and/or to make use of their knowledge to disciplines where modeling and simulation play a critical role.

Candidates for the program must have a 4-year Bachelor's degree from an accredited academic institution. The program requires a student to take four of the following courses. The student is required to take the core course, *at most two* foundation elective courses, and *at least one* advanced elective course. Each course has a normal letter grade. The student is expected to obtain a C or better in each course and maintain a 3.0 GPA.

Core Course

MSIM 601	Introduction to Modeling and Simulation	3
----------	---	---

Foundation Elective Courses (select up to 2 from the following): 6

CS 517	Computational Methods and Software	
CS 578	Computational Geometry, Methods and Applications	
CS 586	Introduction to Parallel Computing	
CS 600	Algorithms and Data Structures	

Advanced Elective Courses (select at least 1 of the following at 700/800 level): 3

CS 712	Stochastic Modeling	
CS 714	Monte Carlo Simulation	
CS 722	Machine Learning	
CS 723	Introduction to Bioinformatics	
CS 724	High Performance Computing and Big Data	
CS 725	Information Visualization	
CS 726	Application of Graphs in Bioinformatics	

Total Hours		12
-------------	--	----

In addition to the listed courses, certain CS 795/CS 895 topics courses may be allowed to count as advanced elective courses, subject to recommendation by the course's instructor and approval by the GPD.

Doctor of Philosophy - Computer Science

Admission

Students entering the PhD program in the Computer Science Department should meet the minimum university graduate admission requirements (<http://www.odu.edu/admission/graduate> (<http://www.odu.edu/admission/graduate/>)). Students may be admitted directly to the PhD program with either a bachelor's or a master's degree. Prior to applying for admission, students are encouraged to visit the department's website and ensure that their research interests match with that of a faculty member. Students are expected to show proficiency in Problem Solving and Programming, Introduction to Computer Architecture, Advanced Data Structures and Algorithms, Introduction to Theoretical Computer Science, and Operating Systems at an undergraduate level. Those entering the PhD program with deficiencies in these areas will have to make them up.

Requirements

A candidate for the doctoral degree in computer science must meet all of the following requirements in addition to the university requirements outlined under the Academic Information section in this Catalog:

- Complete a minimum of 78 credit hours beyond the bachelor's degree or 48 credit hours beyond the master's degree (credits include those for formal coursework, breadth requirements, and dissertation/research work).
- Develop a dissertation topic that is approved by the student's Ph.D. advisor.
- Pass the candidacy examination. *After this step, and after formal courses are complete and the topic for the dissertation is approved by the advisor, the student is considered to be a Ph.D. candidate.*
- Write and defend the dissertation prospectus (proposal).
- Attend at least 10 departmental colloquium events and take the 1 credit hour CS 690 Colloquium.
- Complete 24 credit hours of CS 899 (Doctoral Dissertation).
- Successfully defend the dissertation.

The above must be completed within eight years after admission to the Ph.D. program. Note that students with a degree in a discipline outside of computer science will be required to take prerequisite undergraduate courses that will not be counted towards the 78 credit hours requirement.

Advisor

Upon admission to the PhD program, a faculty advisor will be assigned to the student for general guidance. The student, however, is expected to find a dissertation advisor by the time he or she completes formal coursework. The guidelines for advising are outlined under Student Advising in the University Requirements for Graduate Degrees (<http://catalog.odu.edu/graduate/universityrequirementsforgraduatedegrees/>) section of the Graduate Catalog.

Course Requirements

Students *with* a master's degree in computer science or related field must complete *at least 48 credit hours* of coursework as specified below:

1. A minimum of 24 credit hours of post-master's coursework at 800-level that meets the breadth course requirements. 3 credit hours of coursework at the 600-level may be included here.
2. 24 credit hours or more of dissertation work (CS 899).
3. A maximum of six hours may be transferred into the Ph.D. program from post-master's coursework done elsewhere.

Students *without* a master's degree in computer science or related field must complete *at least 78 credit hours* of coursework as specified below:

1. 12 credit hours of core coursework approved by the GPD from a list of courses including: CS 500, CS 517, CS 550, CS 555, CS 600, and CS 665.
2. A minimum of 42 credit hours of coursework at 600-level or above, of which at least 30 credit hours must be at 800-level that meets the breadth course requirements.
3. 24 credit hours or more of dissertation work (CS 899).
4. Demonstrate proficiency in Problem Solving & Programming, Introduction to Computer Architecture, Advanced Data Structures and Algorithms, Introduction to Theoretical Computer Science, and Operating Systems at an undergraduate level

All students must take and pass CS 800 Research Methods (P/F graded course) during their first year in the PhD program (or within one year of completing required undergraduate prerequisites).

Breadth Course Requirements

Excellence in breadth is assessed through graduate coursework at ODU and evaluated by multiple faculty members. To demonstrate this, a student must take *four* (4) CS graduate courses at the 600 level or above that meet the following conditions:

1. For breadth: At least *one course* in *three (3) different research areas* ("Foundations" does not count as a research area)
2. For depth: At least *three (3) courses* at the 800 level (these courses may also be used to satisfy condition 1)
3. Earn a GPA of at least 3.5 across these four courses.
4. The four courses must be regular courses that also earn a letter grade (thus CS 697, CS 800, CS 891, CS 896, and CS 899 *do not* count towards this requirement). Most CS 895 Topics courses will count towards this requirement, and those eligible will be assigned to an appropriate research area.
5. Courses transferred from another university *do not* count towards this requirement.

See the Department of Computer Science webpage or contact the GPD for information on the course to research area mapping.

Candidacy Examination Process

Upon completing formal coursework (not counting CS 899 credits), before becoming heavily involved in dissertation work, and no later than three years after acceptance into the PhD program (preferably during the first 24 months after admission into the program), the student must pass the Candidacy Examination process. Students may satisfy this requirement by passing the Candidacy Examination or by completing an acceptable Research Event. Both options require a written and an oral component.

Option 1: Candidacy Examination

The topics and papers covered in the Candidacy Examination are set by the student's PhD advisor and other members of the appropriate research area committee. The student will be provided a set of 5-10 papers to review in their research area.

Written Component: The student will write a summary of the papers, highlighting their contributions to the research area, noting how they are related and how they are different. The student must also propose areas for potential future work. Ideally, this document could form the basis for the related work section of the student's dissertation proposal. The written summary should be 15-20 pages in the standard ODU dissertation format.

Oral Component: After the written component has been approved by the area committee, the student will give an oral presentation (about 20-30 minutes long) on the subject to the area committee. At the end of the presentation, the area committee may ask questions to further assess the student's understanding of the papers. At the end of the examination process, the committee will make a pass/fail decision and report this to the GPD.

The goal of this exam is to assess the student's understanding of the content of the papers and the student's ability to critically analyze and articulate the important issues raised in the papers. The student should also be able to identify potential areas for future research that build off of these papers.

A pass decision means that the area committee believes that the student is prepared and qualified to undertake PhD-level research in computer science.

A fail decision means that the area committee believes the student is not ready to undertake PhD-level research in computer science. The area committee should provide recommendations to the student on how to improve their performance. A student who fails the Candidacy Examination may retake the exam once or may attempt the Research Event option once. Two failures will result in the student being dismissed from the PhD program. The student may request a different set of papers to be assigned for a retake.

Option 2: Research Event

Students who enter the PhD program with an acceptable MS thesis or are first author on a high-quality academic submission may be able to satisfy the Candidacy Examination process without taking the Candidacy Exam. Each research area committee determines its own set of conferences and journals in their research area that would be acceptable.

Written Component: The written component may be satisfied either through either an accepted publication or a high-quality, but rejected,

submission to an approved conference or journal. For either option, the student must be the first author.

- Accepted: The research area committee may approve an accepted publication to count as the written component of the Candidacy Examination process.
- Rejected: The research area committee may approve a rejected, but high-quality, submission as the *baseline* for the written component of the Candidacy Examination process. In addition to the submitted paper, the student must construct a document that outlines how the student would address the reviewers' comments.

Oral Component: After the written component has been approved by the area committee, the student will give an oral presentation (about 20-30 minutes long) on the subject to the area committee. At the end of the presentation, the area committee may ask questions to further assess the student's understanding of the research and related work. At the end of the examination process, the committee will make a pass/fail decision and report this to the GPD.

A *pass decision* means that the area committee believes that the student is prepared and qualified to undertake PhD-level research in computer science.

A *fail decision* means that the area committee believes the student is not ready to undertake PhD-level research in computer science. The area committee should provide recommendations to the student on how to improve their performance. A student who fails the Research Event option may attempt the Research Event option once more or may take the Candidacy Exam once. Two failures will result in the student being dismissed from the PhD program.

To use the Research Event option, the student must provide the GPD and the chair of the appropriate research area committee a copy of the document proposed to satisfy the requirement. The document will be evaluated by the area committee to determine if it meets the stated criteria. The chair of the area committee will notify the student and the GPD of the outcome of the evaluation (acceptable or unacceptable). If the publication is deemed unacceptable, the student may submit another publication or take the Candidacy Examination. This does not count as an attempt.

If a student is not first author on a submitted paper to an approved conference/journal before they have completed formal coursework, they must take the Candidacy Examination within the next 6 months.

After passing the Candidacy Exam, completing formal coursework, and having an approved dissertation topic, the student must register for at least 1 credit hour each semester (including summer) until graduation.

Dissertation Committee

After the candidacy exam has been passed and dissertation topic approved, the Dissertation Committee is formed to supervise the dissertation research.

A Dissertation Committee is formed according to the following procedure:

1. The faculty advisor selects the members of the Dissertation Committee in cooperation with the student and the Graduate Studies Committee.
2. The Dissertation Committee consists of at least three ODU faculty members. At least two of these must be from the Computer Science Department and one must be from outside of the Computer Science Department. All committee members must be certified for graduate instruction. The current research interests of the computer science members of the committee should be related to the research goals of the student.
3. Additional members may be appointed to the committee. Adjuncts (approved for graduate instruction) and non-university members may be added with approval of the GPD.
4. The committee must complete a PhD Dissertation Committee Form. This form is submitted to the GPD by the Graduate Studies Committee and to the Dean of the College of Sciences for approval.

Dissertation Prospectus

The oral examination of the written dissertation prospectus, or proposal, is designed to test the student's knowledge of background material related

to the dissertation topic and to determine if the student has identified a significant problem, has a plan of attack, and is ready to proceed with the dissertation research.

At least one week before the scheduled oral examination time (and preferably two weeks before), the student must provide the examination committee with a written dissertation research proposal. The proposal should contain the following items (not necessarily in this order):

- a comprehensive literature review on the dissertation topic that should in particular discuss limitations of current approaches and open problems in the topic area
- a description of the research problem
- a discussion of how the problem relates to other work in the field
- a detailed research plan, including proposed tasks and a timeline for completion
- a list of expected contributions.

During the examination, the student will give a 45-minute presentation of the dissertation proposal to be followed by questions from the committee. The examination committee will be made up of at least three faculty members, all of whom must be graduate certified. These are typically the members of the student's dissertation committee. The exam is expected to last no more than 2 hours.

The presentation of the dissertation proposal is open to the public and will be publicized by the GPD at least one week in advance of the exam. Once the presentation has concluded and the audience has asked general questions, the audience will be excused. The examination by the committee will be held in private, but graduate faculty members are welcome to observe the exam.

Dissertation

A minimum of 24 credit hours of dissertation work (CS 899) is required. The work must represent an achievement in research and must be a significant contribution in the field. Students are required to publish (or have in the revision process) at least one paper in a refereed journal or refereed conference proceedings based on their dissertation work.

Dissertation Defense

The examination will be oral and the examination committee must have the completed dissertation at least two weeks before the examination date. In addition to the examination, students are required to give a public oral presentation on their dissertation results.

Time Requirement

PhD students should normally be full-time. A full-time student can be expected to satisfy all the PhD requirements in three to five years when entering with an MS degree or four to six years with a bachelor's degree. No student (full-time or part-time) will be allowed to study for the PhD degree beyond eight years from the date of admission into the program.

COMPUTER SCIENCE Courses

CS 500. Foundations of Computing. 3 Credits.

The course aims to provide students foundational training in computing. This includes topics in discrete mathematics, counting and combinatorics, probability, proofs methods, basic automata theory and algorithm design and analysis. Prerequisites: MATH 211 or equivalent, CS 250 or equivalent experience with C++ programming and basic data structures.

CS 502. Formal Software Foundations. 3 Credits.

Laboratory work required. Foundational principles and techniques for building correct-by-construction software systems with provable guarantees. Includes functional programming, algebraic and polymorphic data types, pattern matching, computer-assisted theorem proving, proof automation, extraction of certified executable code, examples of verified algorithms. Prerequisite: CS 381 or equivalent experience.

CS 510. Professional Workforce Development I. 3 Credits.

Laboratory work required. Provides students with challenges of business environments in developing a technology based project. Students identify a societal problem, identify solutions, define project solutions, develop project objectives, conduct feasibility analysis, establish organizational group structure to meet project objectives and develop formal specifications. Students make formal technical project presentations and develop web documentation. Students prepare a draft grant proposal.

CS 511. Professional Workforce Development II. 3 Credits.

Laboratory work required. Students write professional and non-technical documents and continue the development of the project defined in CS 410. Written work is reviewed and returned for corrective rewriting. Students will design and develop a project prototype, and demonstrate the prototype to a formal panel along with delivering the formal product specifications and a draft formal grant proposal. This is a writing intensive course.

CS 517. Computational Methods and Software. 3 Credits.

Laboratory work required. Algorithms and software for fundamental problems in scientific computing. Topics: properties of floating point arithmetic, linear systems of equations, matrix factorizations, stability of algorithms, conditioning of problems, least-squares problems, eigenvalue computations, numerical integration and differentiation, nonlinear equations, iterative solution of linear systems.

CS 518. Web Programming. 3 Credits.

Laboratory work required. Overview of Internet and World Wide Web; web servers and security, HTTP protocol; web application and design; server side scripts and database integration, and programming for the Web. Prerequisites: A grade of C or better in CS 312 and CS 330, or equivalent experience.

CS 531. Web Server Design. 3 Credits.

Laboratory work required. Extensive coverage of the hypertext transfer protocol (HTTP), specifications and commentary (IETF RFCs), and implications for servers and clients. Students will develop a web server providing common HTTP functionality and implementing all HTTP (including unsafe and conditional) methods, content negotiation, transfer and content encoding, basic & digest authentication, and server-side execution of programs (i.e., dynamic resources). Frequent in-class demonstrations of progress and protocol conformance will be required. Prerequisites: Familiarity with Internet and network (including socket) programming.

CS 532. Web Science. 3 Credits.

An overview of the World Wide Web and associated decentralized information structures, focusing mainly on the computing aspects of the Web: how it works, how it is used, and how it can be analyzed. Students will examine a number of topics including: web architecture, web characterization and analysis, web archiving, Web 2.0, social networks, collective intelligence, search engines, web mining, information diffusion on the web, and the Semantic Web.

CS 541. App Development for Smart Devices. 3 Credits.

Laboratory work required. Project-oriented coverage of the principles of application design and development for Android platform smart devices. Topics include user interface; input methods; data handling; network techniques; localization and sensing. Students are required to produce a professional-quality mobile application. Prerequisites: Knowledge of Java.

CS 550. Database Concepts. 3 Credits.

Laboratory work required. Three level database architecture. The relational database model and relational algebra. SQL and its use in database procedures and with conventional programming languages. Entity relationship modeling. Functional dependencies and normalization. Transactions, concurrency and recovery. Prerequisites: Familiarity with elementary set theory, propositional logic, and any two programming languages are expected; a course in finite mathematics or discrete structures is recommended.

CS 554. Network Management. 3 Credits.

Laboratory work required. The administration of computer networks and their interaction with wide area networks: network topologies for local and wide area networks, common protocols and services, management of distributed file services, routing and configuration, security, monitoring and trouble-shooting.

CS 555. Introduction to Networks and Communications. 3 Credits.

Internet and the 5-layered protocol architecture for the Internet, applications built on top of data networks, specifically the Internet, the web, the transport layer, TCP and UDP protocols, the network layer, the data link layer, also some of the technologies for the physical layer. Prerequisites: Familiarity with C++ or Java programming for Unix systems.

CS 558. Unix System Administration. 3 Credits.

Laboratory work required. Aspects of administering a SOLARIS/UNIX operating system in a networked environment are covered. Topics covered include installation, file system management, backup procedures, process control, user administration, device management, Network File Systems (NFS), Network Information Systems (NIS), UNIX security, Domain Name Services (DNS), and integration with other operating systems. Prerequisites: experience with UNIX.

CS 560. Computer Graphics. 3 Credits.

Laboratory work required. An introduction to graphical systems and methods. Topics include basic primitives, windowing, transformations, hardware, interaction devices, 3-D graphics, curved surfaces, solids, and realism techniques such as visible surface, lighting, shadows, and surface detail. Requires project involving OpenGL programming.

CS 562. Cybersecurity Fundamentals. 3 Credits.

Introduction to networking and the Internet protocol stack; Vulnerable protocols such as HTTP, DNS, and BGP; Overview of wireless communications, vulnerabilities, and security protocols; Introduction to cryptography; Discussion of cyber threats and defenses; Firewalls and IDS/IPS; Kerberos; Transport Layer Security, including certificates; Network Layer Security.

CS 563. Cryptography for Cybersecurity. 3 Credits.

This course covers mathematical foundations, including information theory, number theory, factoring, and prime number generation; cryptographic protocols, including basic building blocks and protocols; cryptographic techniques, including key generation and key management, and applications; and cryptographic algorithms--DES, AES, stream ciphers, hash functions, digital signatures, etc. Prerequisites: MATH 162M.

CS 564. Networked Systems Security. 3 Credits.

Authentication in cyber systems including password-based, address-based, biometrics-based, and SSO systems; Authorization and accounting in cyber systems; Securing wired and wireless networks; Secured applications including secure e-mail services, secure web services, and secure e-commerce applications; Security and privacy in cloud environments.

CS 565. Information Assurance for Cybersecurity. 3 Credits.

Introduction to information assurance. Topics to be covered include metrics, planning and deployment; identity and trust technologies; verification and evaluation, and incident response; human factors; regulation, policy languages, and enforcement; legal, ethical, and social implications; privacy and security trade-offs; system survivability; intrusion detection; and fault and security management. Prerequisites: MATH 162M or familiarity with computer security area.

CS 566. Principles and Practice of Cyber Defense. 3 Credits.

This course is to help students gain a thorough understanding of vulnerabilities and attacks in systems and networks and learn cyber defense best practices. It covers fundamental security design principles and defense strategies and security tools used to mitigate various cyber attacks. The topics may include identification of Recon Ops, intrusion detection, identification of C2 Ops, data exfiltration detection, identifying malicious codes, network security techniques, cryptography, malicious activity detection, system security architectures, defense in depth, distributed/cloud and virtualization. Laboratory work required. Prerequisites: Basic knowledge of programming, computer architecture, computer networks and operating systems as covered in such courses as CS 250, CS 270 and CS 455; no prior knowledge of computer security is necessary.

CS 567. Introduction to Reverse Software Engineering. 3 Credits.

Laboratory work required. Covers all the major components such as static analysis, dynamic analysis, Windows x86/64 Assembly, APIs, DLL/process injection, covert launching methods, behaviors, anti-disassembly, anti-VM, packing/unpacking, shell code, C++, buffer overflow attacks and various kinds of networking attacks; includes a final project that analyzes a piece of real malware. Prerequisites: CS 250 and CS 270 or equivalent experience.

CS 569. Data Analytics for Cybersecurity. 3 Credits.

The course introduces classical and advanced models and techniques in machine learning and deep learning. It applies these techniques in the cybersecurity domain including anomaly detection, network security, and malware detection and classification. Advanced applications such as self-driving cars and IoT systems are also discussed. In addition, cyber-attacks on machine learning techniques and AI systems and the possible consequences are also discussed. Prerequisites: CS 462/CS 562 or CS 465/CS 565 or experience in cybersecurity.

CS 575. Introduction to Computer Simulation. 3 Credits.

Efficient implementation methods. Time management. Planning and design of simulation experiments. Statistical issues in simulation. Generation of random numbers and stochastic variates. Programming with graphically- and text-based simulation languages. Verification and validation of simulation models. Distributed simulation. Special topics such as HLA will be discussed.

CS 576. Systems Programming. 3 Credits.

Laboratory work required. This course is to help students fully understand and utilize the internal workings and capabilities provided by modern computing, networking and programming environments. Topics include: Shell Script Programming, X Windows (Xlib and Motif), UNIX internals (I/O, Processes, Threads, IPC and Signals), Network Programming (UDP/TCP Sockets and Multicasting) and Java Systems Programming (SWING, Multithreading and Networking).

CS 578. Computational Geometry, Methods and Applications. 3 Credits.

The discipline of Computational Geometry is devoted to the study of algorithms which are formulated in terms of spatially embedded arrangements of objects, such as points, lines, surfaces, and solids. This course covers fundamental algorithms including convex hulls, polygon triangulations, point location, Voronoi diagrams, Delaunaytriangulations, binary space partitions, quadrees, and other topics.

CS 580. Introduction to Artificial Intelligence. 3 Credits.

Laboratory work required. Introduction to concepts, principles, challenges, and research in major areas of AI. Areas of discussion include: natural language and vision processing, machine learning, machine logic and reasoning, robotics, expert and mundane systems.

CS 586. Introduction to Parallel Computing. 3 Credits.

Laboratory work required. The motivation for and successes of parallel computing. A taxonomy of commercially available parallel computers. Strategies for parallel decompositions. Parallel performance metrics. Parallel algorithms and their relation to corresponding serial algorithms. Numerous examples from scientific computing, mainly in linear algebra and differential equations. Implementations using public-domain network libraries on workstation clusters and computers.

CS 588. Principles of Compiler Construction. 3 Credits.

Laboratory work required. Theoretical and practical aspects of compiler design and implementation. Topics will include lexical analysis, parsing, translation, code generation, optimization, and error handling.

CS 595. Topics in Computer Science. 1-3 Credits.

Special topics.

CS 597. Independent Study in Computer Science. 1-3 Credits.

Independent study under the direction of an instructor. Prerequisites: permission of the instructor.

CS 600. Algorithms and Data Structures. 3 Credits.

This course covers the following topics: (i) Basic introduction to algorithms, their design and analysis (ii) Asymptotic notation (iii) Recurrence Relations and their solutions (iv) Sorting and Order Statistics: various algorithms for sorting and their analysis, lower bounds for sorting, computing medians, modes and various order statistics (v) Paradigms for algorithm design and analysis: Dynamic Programming, Greedy Method, Amortized Analysis, and (vi) Graphs and Elementary Graph Algorithms: Breadth-first and Depth-first Search, Topological Sort, Minimum Spanning Trees and Shortest Paths Algorithms. Prerequisites: CS 361 or equivalent and CS 381 or equivalent.

CS 610. Foundations of Cybersecurity. 3 Credits.

This course provides a comprehensive coverage of cybersecurity foundations including basics of networking, malware and cyber attacks, cryptography, firewalls, intrusion detection and prevention systems, Kerberos and SSL/TLS; basics of mobile computing, Internet of Things (IoT), Cyber Physical Systems and threats; information assurance (IA), incident management and response, vulnerabilities and risks; and legal, ethical, and social implications. This course will not be counted towards Computer Science graduate degrees. Prerequisite: CS 270 or familiarity with computer architecture.

CS 620. Introduction to Data Science and Analytics. 3 Credits.

This course will explore data science as a burgeoning field. Students will learn fundamental principles and techniques that data scientists employ to mine data. They will investigate real life examples where data is used to guide assessments and draw conclusions. This course will introduce software and computing resources available to a data scientist to process, visualize, and model different types of data including big data. Cross-listed with DASC 620.

CS 624. Data Analytics and Big Data. 3 Credits.

This course introduces the essential data science tools to work with different types of data including streaming data and big data, including static and streaming data using Python software packages; modeling and predictive analysis using basic machine learning techniques; work with real sample data sets from different disciplines, e.g., the health sciences and finance industry; and how to work with big data using emerging technology such as Apache Spark.

CS 625. Data Visualization. 3 Credits.

This course covers the theory and application of data visualization. This includes issues in data cleaning to prepare data for visualization, theory behind mapping data to appropriate visual representations, introduction to visual analytics, and tools used for data analysis and visualization. Modern visualization software and tools will be used to analyze and visualize real-world datasets to reinforce the concepts covered in the course.

CS 635. Parallel Computer Architecture. 3 Credits.

This is a first course in parallel architecture, with an emphasis on the description and evaluation of commercially available machines. Topics to be covered include: parallelization and performance metrics, scalability and the "laws" of Amdahl and Gustafson, computational similarity, models of computation, parallelization paradigms, network characteristics and topology, communication calculus and templates, pipelining and parallelism, processor types, memory hierarchy, cache coherence protocols, latency-hiding mechanisms, routing algorithms, and languages and libraries to support parallel architecture. Prerequisite: CS 665.

CS 656. Database Methodology. 3 Credits.

Laboratory work required. Analysis, design and implementation of databases and database applications using modern software engineering methods. Database CASE tools. Analysis using process, function, and dataflow analysis in conjunction with entity relationship modeling. Database diagrams and database design. Application suite design and high level design of applications. Refining implementations. Prerequisite: CS 550.

CS 660. 3D Computer Graphics. 3 Credits.

Laboratory work required. The mathematical tools needed for the geometrical aspects of 3D computer graphics. Fundamentals: homogeneous coordinates, transformations and perspective. Theory of parametric and implicit curve and surface models: polar forms, Bezier arcs and de Casteljau subdivision, continuity constraints, B-splines, tensor product, and triangular patch surfaces. Representations of solids and conversions among them. Beometric algorithms for graphics problems, with applications to ray tracing, hidden surface elimination, etc. Prerequisite: CS 560.

CS 665. Computer Architecture. 3 Credits.

A detailed and quantitative study of the architecture of modern uni-processor computers. The major components are: the technology drivers, performance measures, instruction sets (including 80X86, VAX, and a generic RISC which is very similar to the MIPS series), processor implementation, advanced pipelining and superscalar features, cache and memory design, and I/O. The emphasis is on obtaining quantitative measures of performance, describing interactions of the various components, studying trade-offs between the components in commercial processors, and integration into a complete computer system including interaction of the software and hardware. (offered spring).

CS 667. Cooperative Education. 1-3 Credits.**CS 669. Practicum. 1-3 Credits.****CS 690. Colloquium. 1 Credit.**

A one-hour weekly lecture given by faculty from Old Dominion and other institutions.

CS 695. Topics. 1-3 Credits.**CS 697. Independent Study in Computer Science. 1-3 Credits.**

Independent study under the direction of an instructor. Prerequisites: permission of the instructor.

CS 698. Master's Project. 3 Credits.

Departmental permission required.

CS 699. Thesis Research. 3 Credits.

Departmental permission required.

CS 710. Applied Algorithms. 3 Credits.

Laboratory work required. The course will involve solving two or three comprehensive projects anchored in computer science and engineering. Possible topics for projects include: computational issues in network design and analysis; scheduling problems and applications; digital geometry and pattern recognition; image processing and computer vision applications; robotics. The basic thrust is to demonstrate the usefulness and power of algorithm design and analysis in solving real-life problems. Prerequisite: CS 600.

CS 711. Software Validation. 3 Credits.

Laboratory work required. The most common path to improved confidence in a program is via testing. This course explores divergent and sometimes conflicting approaches to conducting testing and to measuring the resulting confidence. Topics include the theoretical basis for testing, common testing methods, statistical measures of program reliability, and the relationship between correctness and reliability.

CS 712. Stochastic Modeling. 3 Credits.

Stochastic processes are ways of quantifying the dynamic relationship of sequences of random events. This course will expose the participants to standard concepts and methods of stochastic modeling, as well as the rich diversity of applications. Topics include, but not limited to, Markov chains in discrete and continuous time, Poisson processes, renewal theory and branching processes.

CS 714. Monte Carlo Simulation. 3 Credits.

This course serves to illustrate important principles in Monte Carlo simulation methods and to demonstrate their power in applications. The course covers Metropolis-Hastings algorithm, Gibbs sampler, Markov Chain Monte Carlo, acceptance-rejection method, Monte Carlo integration, quasi-Monte Carlo, random walk, and random number generation.

CS 722. Machine Learning. 3 Credits.

This course presents both the foundational and the practical aspects of modeling, analyzing, and mining of computerized data sets, including classification, regression, clustering, semi-supervised learning, structured sparsity learning, etc. The course assignments are designed to contain both theoretical and programming components in order to train students to gain hands-on-experience.

CS 723. Introduction to Bioinformatics. 3 Credits.

This course introduces the fundamental knowledge in bioinformatics and the current advances in selected directions. The topics include: fundamental concepts and experimental techniques in molecular biology, computational methods in genomic sequence comparison and analysis, and computational methods in molecular structural modeling.

CS 724. High Performance Computing and Big Data. 3 Credits.

This course introduces parallel and distributed programming principles and has emphasis on hands-on programming and deploying high-performance computing applications with big data for different science and engineering disciplines. Topics includes programming on emerging technologies such as NVIDIA GPU, Hadoop Framework, and Apache Spark for large scale data analytics and mining applications.

CS 725. Information Visualization. 3 Credits.

This course covers the theory and application of information visualization. Research on graph design, visual perception, cognition, and interaction will be covered. Research and practical techniques for the display of graphs, networks, hierarchies, text, and complex multivariate data will be addressed. Course projects will require the development of interactive web-based visualizations.

CS 726. Application of Graphs in Bioinformatics. 3 Credits.

This course links the fundamental concepts and algorithms of graphs with the actual biological problems. Various biological problems will be selected to discuss the formulation of the graph, the graph algorithms, and the performance analysis.

CS 734. Introduction to Information Retrieval. 3 Credits.

Laboratory work required. Theory and engineering of information retrieval in the context of developing web-based search engines. Topics include issues related to crawling, ranking, query processing, retrieval models, evaluation, clustering, machine learning, and other aspects related to building web search engines. Students will perform a mix of hands-on development and coding, as well as theoretical exploration of the research literature.

CS 744. Performance Evaluation of Computer Systems and Networks. 3 Credits.

The course will introduce some of the commonly used techniques in the performance evaluation of computing systems. Students will be exposed to a variety of analytical and simulation tools used in this field. The applicability of the techniques will be illustrated through case studies.

CS 751. Introduction to Digital Libraries. 3 Credits.

Digital Libraries (DLs) are an increasingly popular research area that encompass more than traditional information retrieval or database methods and techniques. The course will cover a brief history of DL development, with emphasis on World Wide Web implementations. Case studies will be performed on various DLs. The class will focus heavily on project work. At the end of the course, students will be prepared to develop, evaluate, or apply digital library technologies in their work environment. Topics include: Repositories; Distributed Searching; Metadata Harvesting; Preservation, Reference Linking and Citation Analysis.

CS 752. Wireless Communications and Mobile Computing. 3 Credits.

This course looks at fundamental issues in the area of wireless networks and mobile computing. The course material is organized around the following broad themes: Basics of mobile and wireless communications; Cellular communications: Bandwidth allocation and reservation, Location management, Call admission strategies and QoS issues; Mobile IP and Mobile TCP; Mobile Ad-Hoc NETWORKS (MANET): Routing, Multimedia and QoS support; Sensor networks.

CS 762. Memory Analysis and Forensics. 3 Credits.

This course is based on the Intel processor architecture employed in Windows, Linux and MacOS operating systems. Students will learn how memory is assigned to processes and how it is addressed, how memory data structures can be exploited by malware, and what is available for forensic analysis of memory. The course involves several hands-on lab work on recognizing process data structures in memory, memory acquisition, and use of a set of tools to catch the malware while preserving evidence from live memory analysis. Course requires a set of assigned reading and lab work. Prerequisite: CS 270, CS 471, or instructor's permission.

CS 764. Blockchains and Cryptocurrencies: Fundamentals, Technologies, and Economics. 3 Credits.

This course covers different aspects of cryptocurrencies, including P2P networks, distributed consensus, Bitcoin and Ethereum, blockchain technologies, cryptographic techniques (secure hashing, encryption, decryption, digital signatures), privacy and anonymity, mining and mining puzzles, wallets, smart contracts, case studies, cryptocurrency ecosystem, legal aspects, implications and impact on economy and finance, and future of cryptocurrencies. Prerequisites: CS 471, CS 455/CS 555 or equivalent experience.

CS 771. Advanced Operating Systems. 3 Credits.

This course covers principles, design decisions, design techniques, policies, and mechanisms in the design and implementation of general-purpose multiprogramming and distributed operating systems. Topics to be covered include: concurrency, interprocess communication, threads, access control, protection and authentication, multiprocessor operating systems.

CS 772. Network Security: Concepts, Protocols and Programming. 3 Credits.

This course deals with the basic protocols, techniques and programming issues to secure internet applications and traffic. Topics include: Cryptographic algorithms tools and concepts; Secure Socket Layer (SSL), Transport Layer Security (TLS) and IPSec protocols; Securing Internet Applications: HTTP, SMTP, UDP and multicast; Hands on socket programming using C and Java.

CS 773. Data Mining and Security. 3 Credits.

Introduction to data mining; Algorithms including naive Bayes, Decision Trees and Rules, Association Rules, Linear classification, and Clustering; Cross validation, Lift charts, ROC Curves; SVM, Bayesian networks, K-means clustering; Data transformation; PCA; Ensemble Learning; Application of data mining to security and privacy including authentication, authorization, and intrusion detection; Privacy-preserving data mining. Prerequisite: CS 471 and CS 455 or CS 555.

CS 774. Distributed System Security with .Net. 3 Credits.

The course provides detailed coverage of security aspects of ASP.Net. It examines distributed system architectures, ASP.Net security framework, cyber attacks, system vulnerabilities, C# and ADO.Net. It also discusses windows and forms authentication, authorization, impersonation, code obfuscation, and advanced concepts, including secure web services, runtime security, operating system security, code access security, role-based access control, and attribute-based access control. Finally, cryptography, XACML and security policies as implemented in ASP.Net are covered. Prerequisites: CS 471 and CS 555.

CS 775. Distributed Systems. 3 Credits.

This course deals with the design issues in distributed computing systems and will discuss the motivation for building distributed systems, various algorithms and protocols proposed in literature for system operability, and some of the experimental distributed systems that have been built in the last few years. Special attention will be paid to the fault-tolerant and performance aspects of these systems. The project component of this course will enable students to get hands-on experience of implementing some of the distributed algorithms.

CS 776. Architectural Support for Cloud Computing. 3 Credits.

Cloud computing requires a great deal of architectural support. This course investigates various types of architectural support that make cloud computing almost infinitely scalable while maintaining efficiency. The course will look at various types of support provided by Google, Amazon, Facebook, Yahoo! and others.

CS 778. Networked Multimedia Systems. 3 Credits.

This course will introduce some of the technical foundations for capturing, transmitting, presentation and storage of continuous multimedia. Students will explore the applications of multimedia and techniques in some areas such as group collaboration and network based education. Topics covered include: Architectures and issues for distributed Multimedia Systems Support for real-time multimedia applications, quality-of-service, synchronization, and presentation of multiple multimedia streams.

CS 779. Design of Network Protocols. 3 Credits.

Understanding the design, implementation and performance of network protocols using TCP/IP protocol suite as a case study. The students will have hands-on experience on low-level tools and will access and study the source code of these protocols and writing networking software applications. Topics include: socket interface, IPv4 and IPv6, routing, UDP, multicasting and IGMP, TCP specification, implementation and performance. Prerequisite: CS 455/CS 555 or equivalent.

CS 791. Graduate Seminar. 1-3 Credits.

Seminar. Prerequisites: permission of the instructor.

CS 795. Topics in Computer Science. 1-3 Credits.

Topics in computer science.

CS 796. Topics in Computer Science. 1-3 Credits.

Topics in computer science. Prerequisites: permission of the instructor.

CS 800. Research Methods. 3 Credits.

Introduction to research methods in computer science. Topics include academic publishing, academic writing, literature reviews, responsible conduct of research, and presenting research results. Research faculty will present overviews of their research and how research is conducted in their labs.

CS 810. Applied Algorithms. 3 Credits.

Laboratory work required. The course will involve solving two or three comprehensive projects anchored in computer science and engineering. Possible topics for projects include: computational issues in network design and analysis; scheduling problems and applications; digital geometry and pattern recognition; image processing and computer vision applications; robotics. The basic thrust is to demonstrate the usefulness and power of algorithm design and analysis in solving real-life problems. Prerequisite: CS 600.

CS 811. Software Validation. 3 Credits.

Laboratory work required. The most common path to improved confidence in a program is via testing. This course explores divergent and sometimes conflicting approaches to conducting testing and to measuring the resulting confidence. Topics include the theoretical basis for testing, common testing methods, statistical measures of program reliability, and the relationship between correctness and reliability.

CS 812. Stochastic Modeling. 3 Credits.

Stochastic processes are ways of quantifying the dynamic relationship of sequences of random events. This course will expose the participants to standard concepts and methods of stochastic modeling, as well as the rich diversity of applications. Topics include, but not limited to, Markov chains in discrete and continuous time, Poisson processes, renewal theory and branching processes.

CS 814. Monte Carlo Simulation. 3 Credits.

This course serves to illustrate important principles in Monte Carlo simulation methods and to demonstrate their power in applications. The course covers Metropolis-Hastings algorithm, Gibbs sampler, Markov Chain Monte Carlo, acceptance-rejection method, Monte Carlo integration, quasi-Monte Carlo, random walk, and random number generation.

CS 822. Machine Learning. 3 Credits.

This course presents both the foundational and the practical aspects of modeling, analyzing, and mining of computerized data sets, including classification, regression, clustering, semi-supervised learning, structured sparsity learning, etc. The course assignments are designed to contain both theoretical and programming components in order to train students to gain hands-on-experience.

CS 823. Introduction to Bioinformatics. 3 Credits.

This course introduces the fundamental knowledge in bioinformatics and the current advances in selected directions. The topics include: fundamental concepts and experimental techniques in molecular biology, computational methods in genomic sequence comparison and analysis, and computational methods in molecular structural modeling.

CS 824. High Performance Computing and Big Data. 3 Credits.

This course introduces parallel and distributed programming principles and has emphasis on hands-on programming and deploying high-performance computing applications with big data for different science and engineering disciplines. Topics include programming on emerging technologies such as NVIDIA GPU, Hadoop Framework, and Apache Spark for large scale data analytics and mining applications.

CS 825. Information Visualization. 3 Credits.

This course covers the theory and application of information visualization. Research on graph design, visual perception, cognition, and interaction will be covered. Research and practical techniques for the display of graphs, networks, hierarchies, text, and complex multivariate data will be addressed. Course projects will require the development of interactive web-based visualizations.

CS 826. Application of Graphs in Bioinformatics. 3 Credits.

This course links the fundamental concepts and algorithms of graphs with the actual biological problems. Various biological problems will be selected to discuss the formulation of the graph, the graph algorithms, and the performance analysis.

CS 834. Introduction to Information Retrieval. 3 Credits.

Laboratory work required. Theory and engineering of information retrieval in the context of developing web-based search engines. Topics include issues related to crawling, ranking, query processing, retrieval models, evaluation, clustering, machine learning, and other aspects related to building web search engines. Students will perform a mix of hands-on development and coding, as well as theoretical exploration of the research literature.

CS 844. Performance Evaluation of Computer Systems and Networks. 3 Credits.

The course will introduce some of the commonly used techniques in the performance evaluation of computing systems. Students will be exposed to a variety of analytical and simulation tools used in this field. The applicability of the techniques will be illustrated through case studies.

CS 851. Introduction to Digital Libraries. 3 Credits.

Digital Libraries (DLs) are an increasingly popular research area that encompass more than traditional information retrieval or database methods and techniques. The course will cover a brief history of DL development, with emphasis on World Wide Web implementations. Case studies will be performed on various DLs. The class will focus heavily on project work. At the end of the course, students will be prepared to develop, evaluate, or apply digital library technologies in their work environment. Topics include: Repositories; Distributed Searching; Metadata Harvesting; Preservation, Reference Linking and Citation Analysis.

CS 852. Wireless Communications and Mobile Computing. 3 Credits.

This course looks at fundamental issues in the area of wireless networks and mobile computing. The course material is organized around the following broad themes: Basics of mobile and wireless communications; Cellular communications: Bandwidth allocation and reservation, Location management, Call admission strategies and QoS issues; Mobile IP and Mobile TCP; Mobile Ad-Hoc NETWORKS (MANET): Routing, Multimedia and QoS support; Sensor networks.

CS 862. Memory Analysis and Forensics. 3 Credits.

This course is based on the Intel processor architecture employed in Windows, Linux and MacOS operating systems. Students will learn how memory is assigned to processes and how it is addressed, how memory data structures can be exploited by malware, and what is available for forensic analysis of memory. The course involves several hands-on lab work on recognizing process data structures in memory, memory acquisition, and use of a set of tools to catch the malware while preserving evidence from live memory analysis. Course requires a set of assigned reading and lab work. Prerequisite: CS 270, CS 471, or instructor's permission.

CS 864. Blockchains and Cryptocurrencies: Fundamentals, Technologies, and Economics. 3 Credits.

This course covers different aspects of cryptocurrencies, including P2P networks, distributed consensus, Bitcoin and Ethereum, blockchain technologies, cryptographic techniques (secure hashing, encryption, decryption, digital signatures), privacy and anonymity, mining and mining puzzles, wallets, smart contracts, case studies, cryptocurrency ecosystem, legal aspects, implications and impact on economy and finance, and future of cryptocurrencies. Prerequisites: CS 471, CS 455/CS 555 or equivalent experience.

CS 871. Advanced Operating Systems. 3 Credits.

This course covers principles, design decisions, design techniques, policies, and mechanisms in the design and implementation of general-purpose multiprogramming and distributed operating systems. Topics to be covered include: concurrency, interprocess communication, threads, access control, protection and authentication, multiprocessor operating systems.

CS 872. Network Security: Concepts, Protocols and Programming. 3 Credits.

This course deals with the basic protocols, techniques and programming issues to secure internet applications and traffic. Topics include: Cryptographic algorithms tools and concepts; Secure Socket Layer (SSL), Transport Layer Security (TLS) and IPsec protocols; Securing Internet Applications: HTTP, SMTP, UDP and multicast; Hands on socket programming using C and Java.

CS 873. Data Mining and Security. 3 Credits.

Introduction to data mining; Algorithms including naive Bayes, Decision Trees and Rules, Association Rules, Linear classification, and Clustering; Cross validation, Lift charts, ROC Curves; SVM, Bayesian networks, K-means clustering; Data transformation; PCA; Ensemble Learning; Application of data mining to security and privacy including authentication, authorization, and intrusion detection; Privacy-preserving data mining. Prerequisite: CS 471 and CS 455 or CS 555.

CS 874. Distributed System Security with .Net. 3 Credits.

The course provides a detailed coverage of security aspects of ASP.Net. It examines distributed system architectures, ASP.Net security framework, cyber attacks, system vulnerabilities, C# and ADO.Net. It also discusses windows and forms authentication, authorization, impersonation, and code obfuscation, and advanced concepts, including secure web services, runtime security, operating system security, code access security, role-based access control, and attribute-based access control. Finally, cryptography, XACML and security policies as implemented in ASP.Net are covered. Prerequisites: CS 471 and CS 555.

CS 875. Distributed Systems. 3 Credits.

This course deals with the design issues in distributed computing systems and will discuss the motivation for building distributed systems, various algorithms and protocols proposed in literature for system operability, and some of the experimental distributed systems that have been built in the last few years. Special attention will be paid to the fault-tolerant and performance aspects of these systems. The project component of this course will enable students to get hands-on experience of implementing some of the distributed algorithms.

CS 876. Architectural Support for Cloud Computing. 3 Credits.

Cloud computing requires a great deal of architectural support. This course investigates various types of architectural support that make cloud computing almost infinitely scalable while maintaining efficiency. The course will look at various types of support provided by Google, Amazon, Facebook, Yahoo! and others.

CS 878. Networked Multimedia Systems. 3 Credits.

This course will introduce some of the technical foundations for capturing, transmitting, presentation and storage of continuous multimedia. Students will explore the applications of multimedia and techniques in some areas such as group collaboration and network based education. Topics covered include: Architectures and issues for distributed Multimedia Systems Support for real-time multimedia applications, quality-of-service, synchronization, and presentation of multiple multimedia streams.

CS 879. Design of Network Protocols. 3 Credits.

Understanding the design, implementation and performance of network protocols using TCP/IP protocol suite as a case study. The students will have hands-on experience on low-level tools and will access and study the source code of these protocols and writing networking software applications. Topics include: socket interface, IPv4 and IPv6, routing, UDP, multicasting and IGMP, TCP specification, implementation and performance. Prerequisite: CS 455/CS 555.

CS 891. Graduate Seminar. 1-3 Credits.

Seminar. Prerequisites: permission of the instructor.

CS 895. Topics in Computer Science. 1-3 Credits.

Topics in computer science.

CS 896. Topics in Computer Science. 1-3 Credits.

Topics in computer science. Prerequisites: permission of the instructor.

CS 899. Doctoral Dissertation. 1-9 Credits.

Research for the doctoral dissertation. Departmental permission required.

CS 998. Master's Graduate Credit. 1 Credit.

This course is a pass/fail course for master's students in their final semester. It may be taken to fulfill the registration requirement necessary for graduation. All master's students are required to be registered for at least one graduate credit hour in the semester of their graduation.

CS 999. Doctoral Graduate Credit. 1 Credit.

This course is a pass/fail course doctoral students may take to maintain active status after successfully passing the candidacy examination. All doctoral students are required to be registered for at least one graduate credit hour every semester until their graduation.