MASTER OF SCIENCE IN COMPUTER SCIENCE (MSCS)
Department of Computer Science and Information Technology (CSIT)
University of the District of Columbia

Faculty
Chair: Byunggu Yu
Professors: David Barnett
Associate Professors: Li Chen, Gail Finley, Lily Liang, LaVonne Manning, Byunggu Yu, Sherali Zeadally
Assistant Professors: Duane Shie
Graduate Faculty: Barnett, Chen, Yu, Liang, Zeadally

The CSIT Department has a graduate program in computer science which leads to the Master of Science in Computer Science (MSCS) degree. The program is offered at the University of the District of Columbia's Van Ness campus.

The MSCS program is tailored to meet the needs of traditional domestic and international students as well as working professionals in the greater Washington DC area. The program emphasizes a practitioner-oriented curriculum which includes the study of advanced algorithms, network security, artificial intelligence, computer graphics, digital image processing, software systems, and database applications.

The program offers a thesis option and a non-thesis option. Students in both options must take four graduate computer science (CS) core courses. Students who wish to write a thesis take four graduate CS electives and sign up for 6 thesis credits (counts as two graduate electives) whereas students in the non-thesis option must take six graduate CS electives. One of these six electives can be 3530 599 Master's Project.

Students in the thesis option must find a thesis advisor in the department based on their interests in order to comply with the thesis submission requirements of the UDC Graduate School.

Students in the non-thesis option, but wishing to do a special project as part of their Master's program, need to find a faculty member (project supervisor).

Qualified students with little or no formal education in computer science at the undergraduate level can be accepted into the program based on the faculty's decision; however, such students must successfully complete a sequence of background or migration courses with a grade of B or better before they can enroll in the regular graduate CS courses. No credit will be given for these background courses toward the MSCS degree.

Admission Requirements for the MSCS Program

All students (US and international) must submit the following documents if they wish to be considered for admission into the MSCS Program:

Completed/signed application form;
Non-refundable application fee;
Official transcripts from each college or university attended;
Graduate Record Examination (GRE) Basic test scores;
Two professional references;

International students must also submit TOEFL [Test of English as a Foreign Language] scores and test scores on the advanced portion of the GRE [Graduate Record Examination] as part of their application. It is the policy of the graduate admissions committee in the CSIT department to carefully consider every applicant's previous academic and professional qualifications, test scores and achievements before an admission decision is made. Students admitted into the graduate program may start either in the fall semester or the spring semester.

Applicants accepted for graduate study will be informed in writing, at the time of admission, whether they need to enroll in background courses and/or prerequisites requiring completion before commencing their graduate studies. Each student admitted into the program will be assigned a graduate advisor and the student is responsible for discussing any special needs they may have with their adviser. Please note that, unless otherwise stipulated, every course in the MSCS program carries 3 credits.

Graduation Requirements
In order to obtain the MSCS degree, students must successfully complete *a minimum of 30 graduate credit hours in computer science* with a grade of B or better in each of the courses. Students receiving grades lower than a B in any course will have to retake the course and obtain a grade of at least a B. Students must complete all program requirements within six years of their initial enrolment in the program.

The course of study is as follows:
30 credit hours required for graduation. Four required 3530 core courses and six 3530 electives. In addition, students who are inadequately prepared in computer science but who have been admitted into the graduate program will also need to take up to 4 background courses in computer science.

Students enrolled in the thesis option will need to submit a thesis and defend it successfully in an oral examination conducted by the student's thesis committee. Students must follow the requirements of the Graduate School in preparing the thesis. Students who elect to do a project must consult with their project advisor on project submission requirements.

Background Courses
The background courses must be taken or may be waived without replacement based on previous college courses. These courses do not count toward the minimum 30 credit hours required by the program.
3530 500 Discrete Mathematics
3530 501A or 501B Object-oriented Programming
3530 502 Systems Software
3530 503 Data Structures

Core Courses
All students in the MSCS program must take four of the following seven courses.
3530 504 Algorithm Design and Analysis
3530 505 Foundations of Computer Architecture
3530 506 Principles of Operating Systems
3530 507 Principles of Database Systems
3530 508 Principles of Data Communication Networks
3530 509 Foundations of Software Engineering
3530 510 Principles of Artificial Intelligence

Electives
Students are advised to choose an appropriate set of **six** courses with their advisors. The specific prerequisites for the various courses are given in the course description section of the catalog. Students may enroll in the 3530 5X8 Topics course more than once, provided different special topics are involved. Students may, however, enroll in 3530 599 Master's Project only once; students electing to write a thesis must enroll in 3530 600 Master's Thesis twice.

Course Descriptions

Background/Migration Courses
The following courses must be taken by students with no formal educational preparation in computer science prior to their enrolling as regular students in the graduate program in computer science at UDC.

3530 500 Discrete Mathematics
This course emphasizes the relationships between certain mathematical structures and various topics in computer science. Topics include set theory, graphs and trees, algorithms, propositional calculus, logic and induction, functions, relational algebra, and matrix algebra.
Prerequisites: none
NOTE: This course DOES NOT count toward the Master of Science in computer science degree.

3530 501A Introduction to Java Programming
The course is designed to facilitate students without a background in computer programming to become proficient Java programmers. Topics include language syntax, data typing, control constructs, arithmetic, logical and string expressions, elementary and composite data structures, methods, and exception handling. Object oriented classes, polymorphism and inheritance. Prerequisites: none
NOTE: This course DOES NOT count toward the Master of Science in computer science degree.

Or

3530 501B Object-Oriented Programming with C++
This course introduces the fundamental principles and technology of object-oriented programming, with emphasis on the C++ programming language. Topics include data abstraction, information hiding, encapsulation, inheritance, polymorphism, templates, exceptions, and object-oriented considerations for software design and reuse.
Prerequisites: none
NOTE: This course DOES NOT count toward the Master of Science in computer science degree.

3530 502 Introduction to Systems Software
This course provides a fundamental understanding of the design and structure of system programs such as assemblers, compilers, interpreters, linkers, loaders, and operating systems. The compilation of block-structured languages is used as a basis for understanding lexical, syntactic, and semantic analysis as well as code generation and optimization. The course concludes with a study of operating system topics such as process scheduling, memory management, and file systems.
Prerequisites: none
NOTE: This course DOES NOT count toward the Master of Science in computer science degree.

3530 503 Data Structures
Topics include dynamic allocation, recursion, the implementation of stacks, queues, linked lists, and trees, as well as algorithms for sorting and searching. The emphasis is on the trade-offs associated with the use of alternative data structures. This course includes several significant programming assignments.
Prerequisites: 3530 500
NOTE: This course DOES NOT count toward the Master of Science in computer science degree.

Required Core Courses
Students enrolled in the MSCS program who do not have a computer science degree at the bachelor's level are required to take **four** of the following seven courses and pass them with a grade of B or better.
Students holding a BS degree in computer science may have taken one or more of these seven courses in their undergraduate program; if so, they only need to take those courses that they have not had in their undergraduate program. However, since they still need to complete a total of 10 courses for their degree, they can fulfill this requirement by choosing from the list of elective courses.

3530 504 Design and Analysis of Algorithms
This course focuses on the design and analysis of algorithms to solve various classes of computational problems. Algorithmic techniques to be studied include divide-and-conquer, dynamic programming, greedy methods, amortized analysis, branch-and-bound, randomizing, and backtracking.
Prerequisites: 3530 500, 503

3530 505 Foundations of Computer Architecture
The internal structure and operation of modern computer systems is examined in this course. Topics to be discussed include the design and operation of the ALU, FPU, and CPU; micro programmed control vs. hardwired control, pipelining, RISC vs. CISC machines, and various memory systems including caches and virtual memory; An introduction to parallel and vector processing, multiprocessor systems and interconnection networks will also be presented. System performance will also be addressed.
Prerequisites: 3530 502

3530 506 Principles of Operating Systems
In this course theoretical and implementation aspects of operating system design are presented from both developer and user perspectives. Parallelism or concurrency aspects are explained using the concepts of process management, synchronization, deadlocks, job and process scheduling. Detailed techniques of real and virtual storage management are discussed for a variety of processing environments such as multi-programming, multi-processing, etc. Students will be designing simulated operating system components and implementing them using a high-level language.
Prerequisites: 3530 502

3530 507 Principles of Database Systems
This course focuses on theoretical and design aspects of database management system software. Topics include the entity-relationship model, database system architectures, data models, and file organization and access methods. A variety of database models including the relational, object-oriented and network models

will be discussed. Other topics include normal forms, concurrency management, query languages and query optimization.
Prerequisites: 3530 503

3530 508 Principles of Data Communications Networks
This course provides a unified treatment of data communications networks from the perspective of data communication principle, components and services, line control techniques and network requirements and design. Topics include transmission principles and media, data encoding and channel capacity, modems and modulation techniques, error and line control techniques, protocols, data compression techniques, switching technologies, common carriers' services and facilities and regulatory requirements
Prerequisites: graduate student standing with no deficiencies.

3530 509 Foundations of Software Engineering
Fundamental software engineering techniques and methodologies commonly used during software development are studied. Topics include various life cycle models, project planning and estimation, requirements analysis, program design, construction, testing, maintenance and implementation, software measurement, and software quality.
Prerequisites: 3530 501A or 501B, 503

3530 510 Principles of Artificial Intelligence
In this course, the highly diverse field of artificial intelligence is explored from a theoretical and practical perspective. A variety of schemes for representation and reasoning will be discussed. Topics focusing on representation include symbolic, rule-based, frame-based, object, and semantic net systems. Topics focusing on reasoning include inductive, abductive and deductive systems, non-monotonic reasoning, temporal reasoning, model-based reasoning, and planning. Common LISP and Prolog will also be briefly discussed.
Prerequisites: 3530 503

Electives
Courses listed in this section can be used by students in designing a program which meets their specific needs and interests. Students may concentrate on one of five areas – theoretical computer science, computer design and system software, specialized applications, network security, and intelligent systems.

3530 511 Automata Theory and Formal Languages
This course covers finite state machines and their limitations, tape automata and their limitations, Turing machines and basics of recursive functions, Post and Thue systems, word problems, phrase-structure grammars, and the different versions of the halting problem.
Prerequisites: 3530 504

3530 512 Computational Complexity
Computational complexity and its applications in computer science and cryptography are explored. Basic concepts of polynomial, NP, and NP-Complete problems are developed in both intuitive and rigorous forms. Methods for determining the tractability of problems, the polynomial hierarchy, techniques and complexity of approximation algorithms, and current topics in complexity are also covered. The course also covers complexity topics in cryptography.
Prerequisites: 3530 511

3530 513 Parallel Algorithms
This course introduces students to parallel computation and algorithm design for parallel machines. Topics include adapting conventional algorithms to fit parallel execution models and stochastic methods suitable for massively parallel machines. Selected readings from the literature will be required.
Prerequisites: 3530 504

3530 518 Topics in Theoretical Computer Science

3530 521 Advanced Computer Architectures
In this course novel computer architectures are explored. Topics include parallel machines, multiprocessor and multi computer machines, dataflow machines, biologically inspired architectures, quantum computers and various interconnection structures. Performance evaluation aspects will also be considered. Selected readings from the literature will be required.
Prerequisites: 3530 505

3530 522 Advanced Operating Systems
This course presents the design principles and applications of advanced operating systems. Topics include communications in distributed systems based on layered protocols, asynchronous transfer mode networks, the client-server model, remote procedure call, synchronization and deadlock in distributed systems; Various concurrency algorithms will also be presented.
Prerequisites: 3530 506

3530 523 Advanced Database Systems
This course investigates the principles of object-oriented and distributed database systems, with an emphasis on algorithms and protocols for handling the complexity of managing data in a distributed environment. Topics include object-oriented and extended relational data models, object identity and persistence, replication, distributed concurrency control, distributed query processing and optimization, data security, semantic integrity control, optimal resource allocation, reliability, and failure recovery.
Prerequisites: 3530 507

3530 524 Human-Computer Interfaces
This course covers the principles, concepts, and objectives of human engineering for interactive systems. Topics include definition of human factors, syntactic and semantic models of user behavior, design principles for user interfaces, interface presentation techniques, and evaluation methods. Selected readings from current research literature will be assigned.
Prerequisites: 3530 509

3530 525 Compiler Design
This course explores the principles, algorithms, and data structures involved in the design and construction of compilers. Topics include context-free grammars, lexical analysis, parsing techniques, symbol tables, error recovery, code generation, and code optimization. Each student will implement a compiler for a small programming language.
Prerequisites: 3530 502

3530 531 Principles of Computer Graphics
Techniques and algorithms for creating and displaying a variety of 2-d and 3-d objects on raster-scan devices are discussed. The mathematics underlying 2-d and 3-d rotations, reflections, scaling and perspective transformations will be presented. Algorithms for clipping lines and polygons, curve fitting, surface rendering, etc. will also be presented.
Prerequisites: 3530 503

3530 532 Image Processing
Fundamentals of image processing are covered, with an emphasis on digital techniques. Topics include digitization, enhancement, segmentation, the Fourier transform, filtering, restoration, reconstruction from projections, and image analysis including computer vision. Concepts are illustrated by laboratory sessions in which these techniques are applied to practical situations, including examples from biomedical image processing.
Prerequisites: 3530 531

3530 533 Computational Geometry
Computational Geometry is used to developing algorithms for solving geometric problems in continuous spaces. It has deep connections to classical mathematics, theoretical computer science, and practical applications such as computer vision, graphics, and engineering such as CAD. The problems dealt with are typically posed as spatial decompositions such as polygon partitioning and triangulation, convex hulls, Voronoi diagrams and Delaunay triangulations, geometric search, and curves and surfaces.
Prerequisites: 3530 504, 531

CSC 534 Bioinformatics
A variety of algorithms for the representation and visualization of genetic data will be presented in this course. Appropriate material drawn from the fields of biology, physics and chemistry will also be presented so that the nature of genetic data can be understood. Extensive readings will be required.
Prerequisites: 3530 531. May be taken concurrently.

3530 538 Topics in Specialized Applications

3530 551 Computer Network Architectures and Protocols

This course covers the architecture and principles of operation of integrated broadband networks particularly those capable of supporting different types of traffic (voice, video, data, graphics) over local and wide area networks. The focus in this course is on high-speed networks (LANs, WANs), switching designs and architectures, router designs and routing protocols, MPLS, IPv6, optical networking, satellite communications, and network performance evaluation. Hands-on practical projects are an integral part of the course
Prerequisites: 3530 505

3530 552 Network Programming
This course provides programming skills useful for network designers and network application developers. It first covers a brief introduction to networking concepts and protocols. The course then covers topics including: the UNIX model, socket programming (TCP/UDP/raw sockets) for client-server systems, Internet addressing, application protocols (SMTP, DNS, Telnet, ftp), Remote Procedure Calls (RPCs), multicasting, secure protocols (e.g. IPSec). The course places a strong emphasis on the completion of hands-on projects.
Prerequisites: 3530 508

3530 553 Network Security
This course provides students with a comprehensive overview of fundamental network security concepts, techniques, and issues. The course covers topics including: security basics and fundamentals, attackers and their attacks, secure data transmission protocols, cryptography, key management, security management, intruders and intrusion detection, operational security policies and procedures. This course also covers security approaches deployed in local and wide area networks. Hands-on practical projects are an integral part of the course.
Prerequisites: 3530 551

3530 554 Wireless and Mobile Computing
Ubiquitous access of information anywhere, anytime, from any device is being made possible to a large extent by wireless and mobile computing technologies. This course discusses key concepts of wireless communications, wireless networks including WiFi, Bluetooth, WiMax, ad hoc networks, cellular technologies (CDMA, UMTS, etc), mobility protocols (including mobile IP, SIP, SCTP), internetworking design architectures for heterogeneous wireless networks, mobility management techniques (handoff and location management), wireless Web (WAP), energy management algorithms, and sensor networks. The course places a strong emphasis on the completion of hands-on projects.
Prerequisites: 3530 551

CSC 558 Special Topics in Network Security

3530 571 Logic Programming
This course provides an introduction to Prolog, the theoretical foundations of logic programming, and current research on applications of logic within artificial intelligence. Topics include a review of first-order logic, the resolution principle, semantics of logic programs and alternative proof procedures. Alternatives to first-order logic such as modal logics for representing and reasoning about knowledge and belief, and non monotonic and default logics will also be discussed. Assignments include problem sets and a number of Prolog programs.
Prerequisites: 3530 510

3530 572 Evolutionary Computation
This course focuses on concepts and techniques from genetic algorithms, genetic programming, and artificial life for modeling and developing software agents capable of solving problems as individuals and as members of a larger "community" of agents. Algorithms for solving optimization and learning problems will be stressed.
Prerequisites: 3530 510. Some background in probability and statistics.

3530 573 Neural Networks
This course provides an introduction to concepts in neural networks and connectionist models. Topics include parallel distributed processing, learning algorithms and applications. Specific networks discussed include Hopfield networks, bidirectional associative memories, perceptrons, feed forward networks with back propagation, and competitive learning networks, including Kohonen and Grossberg networks.
Prerequisites: 3530 510. Good mathematical background in calculus and differential equations.

3530 574 Natural Language Processing

This course covers the concepts and methods for the automated processing of natural language. Topics include pattern matching, parsing, dictionary and lexical acquisition, semantic interpretation, anaphoric reference, discourse analysis, and text generation and understanding.
Prerequisites: 3530 510. 3530 525 highly recommended. Prior exposure to linguistics would be helpful.

3530 575 Speech-based Computing
Topics addressed in detail in this course include the anatomy, physiology and physics of speech generation and reception, speech signal analysis/synthesis and computer representations of spoken data. Systems to be discussed include text-to-speech, speech to text, multilingual speech software and speaker identification/verification.
Prerequisites: 3530 574

CSC 578 Special Topics in Intelligent Systems

CSC 598 Master's Project

CSC 600 Master's Thesis [3 credits/term; 6 credits maximum]