

Computer Science

Richard LeBlanc, PhD, Chair

Degree Offered

Master of Science in Computer Science (MSCS)

Master of Science in Computer Science

The Master of Science in Computer Science (MSCS) program offers a curriculum based on theoretical foundations and practical applications. The program is designed for computing professionals seeking advanced study in computer science. Full-time students, registering for two courses per quarter, can complete the program in five quarters. The program also accommodates part-time students who can complete the program in three years.

The curriculum includes three required advanced courses in core areas of computer science. A set of elective courses ensures students develop a solid, broad understanding of computer science and software engineering. Under the supervision of a faculty member, students are required to complete a project spanning two to three quarters. The project consists of either implementing a software system or carrying out research within computer science. Software implementation projects enhance students' programming skills and provide hands-on experiences in system design and building. Research projects culminate in a report suitable for publication, and provide excellent preparation for those who are pursuing research careers or further education in a doctoral program.

Seattle University has long been a leader in computer science and software engineering education. It established its pioneering Master of Software Engineering program in 1979, and awarded the world's first MSE degree in 1982. Our graduate faculty has strong academic credentials and is well connected to the computing industry in the Seattle area.

Degree Requirements – Master of Science in Computer Science

Minimum requirements for the degree are 45 graduate credits. These include 12 credits of required courses, 15 credits of breadth electives, 8-9 credits of elective courses, and 9-10 credits of a graduate project. Students may be admitted provisionally with a requirement that they complete specified undergraduate courses before obtaining regular status in the program. A maximum of 12 credits taken in non-matriculated status may be applied to this program. Students are allowed up to 10 transfer credits as described earlier in this *Bulletin*. All degree requirements must be completed within six years after course work has begun.

I. Required Courses

12 credits, including:

CPSC 510 Algorithms	4
CPSC 532 Software System Design	4
CPSC 545 Computing Systems	4

II. Breadth Electives

15 credits, from the following:

CPSC 522 Design Patterns and Refactoring	5
CPSC 534 Software Testing and Debugging	5
CPSC 544 Distributed Systems	5
CPSC 560 Computer Networks	5
CPSC 565 Computer Graphics	5
CPSC 570 Artificial Intelligence	5

CPSC 571 Physical Database Design and Optimization	5
CPSC 572 Security in Computing	5
CPSC 585 Compiler Principles and Techniques	5
CPSC 591-593 Special Topics	5

III. Graduate Project

9-10 credits, consisting of a two to three quarter project sequence. Choose either:

- a. CPSC 586 Graduate Project** **9-10**
- b. CPSC 587 Graduate Research Project** **9-10**

IV. General Electives

8-9 credits, additional CPSC breadth electives and/or MSE courses from the following list:

- SEGR 523 Human-Computer Interaction**
- SEGR 534 Software Testing**
- SEGR 543 Applied Formal Methods**
- SEGR 550 Distributed Computing**
- SEGR 551 Embedded Systems**
- SEGR 553 Artificial Intelligence**
- SEGR 572 Software Security**

NOTE: Students may also take a maximum of three credits of Internship (CSPC 595), Independent Study (CSPC 596) and/or Directed Research (CSPC 599) for general elective credit, without department approval.

Minimum credits required for degree 45

Graduate Project

Satisfactory performance (B- or better) in graduate project is required of all MSCS students. A project consisting of 9-10 credits carried out in either a two or three quarter sequence. The project requires either implementation of a software system or carrying out research within computer science.

Computer Science Courses

Eligibility to remain in courses for which students are registered will be based on the criteria listed within each course description and the program admission criteria, and will be determined by the instructor after the first day of class.

CPSC 510 Algorithms..... **4**

Algorithm analysis, algorithm design techniques (such as dynamic programming and greedy methods), randomized algorithms, NP-complete problems, approximation algorithms, and parallel algorithms. Application of algorithms to real world problems.

CPSC 522 Design Patterns and Refactoring **5**

Categorization of standard design patterns, their use, expected benefits and associated costs. Explication and analysis of creational, interface, structural and behavioral patterns. Use of patterns relative to legacy code, including characteristics of poorly designed code as well as the misapplication of patterns. Examination and application of refactoring, techniques used to modify existing code in order to improve structure or performance.

CPSC 532 Software System Design..... **4**

Design of large complex software systems. Understanding, interacting, and extending legacy systems. Design and analysis of software systems functional and non-functional properties. System integration, middleware technologies, and software quali-

ty. Students may not get credit for both CPSC 532 and SEGR 532.

CPSC 534 Software Testing and Debugging 5

Overview of testing and debugging principles. Topics include program analysis, testing adequacy, functional testing, structural testing, unit testing, integration testing, and systematic debugging. Students may not get credit for both CPSC 534 and SEGR 534.

CPSC 544 Distributed Systems 5

Distributed systems help programmers aggregate the resource of many networked computers to construct highly available and scalable services. Abstractions, design and implementation techniques used in the construction of scalable, high performance distributed systems. Topics include (but not limited to) multithreading, network programming, consistency, fault tolerance, security and several case studies of distributed systems. Students may not get credit for both CPSC 544 and SEGR 550.

CPSC 545 Computing Systems 4

Graduate-level introduction to operating system designs and implementations as well as operating system support for networking. Concurrent processes, process communication, resource allocation, process scheduling, concurrent programming (threads and synchronization), and an introduction to security and networking. Selected materials will be added as case studies.

CPSC 560 Computer Networks 5

The study of computer networks and the services built on top of them. Topics include packet-switch and multi-access networks, reliable data transfer, routing and multicasting, Internet protocols (IP, TCP, BGP), Internet architecture, the client-server model, network programming, peer-to-peer networks and security.

CPSC 565 Computer Graphics..... 5

Fundamentals of image representation and computer graphics. Techniques for computer image synthesis, including line drawing, color representation, surface shading, texture mapping, and programming graphics processors.

CPSC 570 Artificial Intelligence 5

Concepts and techniques of artificial intelligence, with an emphasis on building intelligent agents, environments and systems. Methods and tools for building systems that can interact intelligently with their environment by learning and reasoning about the world. Students may not get credit for both CPSC 570 and SEGR 553.

CPSC 571 Physical Database Design and Optimization..... 5

Analysis and conversion of a logical data model to a physical model using a range of storage techniques including partitioning, creating new data types, index organized, clustered, multi-dimensional tables, etc. A major focus is query optimization and retrieval strategies used by major commercial database vendors in both transaction processing and dimensional databases. Other topics to be covered will be security, privileges, distributed processing, and transaction control.

CPSC 572 Security in Computing 5

Fundamental principles of security with emphasis on various aspects of security related to operating systems, networks, databases, and web applications. Discussion on privacy concerns and social and ethical issues related to security. Students may not get credit for both CPSC 572 and SEGR 572.

CPSC 585 Compiler Principles and Techniques..... 5

Lexical analyzers, grammars, top-down and bottom-up parsing, symbol tables, internal forms and intermediate languages, code generation, code optimization, semantic specifications, error detection and recovery. Use of software tools and standard architectures for compiler construction.

CPSC 586 Graduate Project3 to 5

CPSC 587 Graduate Research Project3 to 5

A project consisting of 9-10 credits carried out in either a two or three quarter sequence. Students can either implement a software system (CPSC 586) and/or carry out research within computer science (CPSC 587). During the first quarter of the sequence students will study background material, complete a project proposal, and begin working on the project. Students registering for CPSC 587 will be required to write a research paper suitable for publication in a workshop or conference during the final quarter of the project.

CPSC 591-593	Special Topics.....	1 to 5
CPSC 595	Internship.....	1 to 3
CPSC 596	Independent Study	1 to 5
CPSC 599	Directed Research.....	1 to 5