

COMPUTER SCIENCE, M.S.

Learning Outcomes

- At the time of graduation a Master of Science/Engineering student should be able to describe the techniques and principles for the development of high performance computer systems.
- At the time of graduation a Master of Science/Engineering student should be able to describe the details of extant computer architectures.
- At the time of graduation a Master of Science/Engineering student should be able to quantitatively analyze aspects of computer architecture and draw conclusions about performance.
- At the time of graduation a Master of Science/Engineering student should be able to use models of languages, such as regular expressions and context-free grammars, to develop parsers for specific languages and construct intermediate representations such as abstract syntax trees.
- At the time of graduation a Master of Science/Engineering student should be able to decorate abstract syntax trees according to the scope and type rules of a language, and use the decorated tree to generate intermediate code.
- At the time of graduation a Master of Science/Engineering student should be able to recognize optimizations that compilers apply to transform intermediate code into more efficient code.
- At the time of graduation a Master of Science/Engineering student should be able to work with basic aspects of discrete math related to the analysis of algorithms and data structures, e.g., sums, probability, basic properties of trees and graphs, asymptotic analysis and amortized analysis.
- At the time of graduation a Master of Science/Engineering student should be able to analyze the time and space resources used by complex algorithms.
- At the time of graduation a Master of Science/Engineering student should be able to analyze the time and space resources used by complex algorithms.
- At the time of graduation a Master of Science/Engineering student should be able to use high-order principles of algorithm construction, e.g., divide-and-conquer, dynamic programming, greedy algorithms, graph algorithms.
- At the time of graduation a Master of Science/Engineering student should be able to understand the concept of NP-completeness and be able to find simply polynomial reductions between decision problems.
- At the time of graduation a Master of Science/Engineering student should be able to integrate components to form coherent well designed system.
- At the time of graduation a Master of Science student should be able to independently explore a research topic.

Degree Requirements (30 Hours)

The Master of Science in Computer Science (MSCS) degree requires 30 credit hours beyond the BS. Students in the MSCS program may elect either the thesis or the non-thesis option. The course work must include:

Core (10 hours)

Course	Title	Credits
CSCE 513	Computer Architecture	3
CSCE 531	Compiler Construction	3

CSCE 750	Analysis of Algorithms	3
CSCE 791	Seminar in Advances in Computing	1

Total Credit Hours 10

Electives (20 hours)

A maximum of six hours in non-CSCE courses and at most three hours of CSCE 798 may be applied toward the degree. CSCE 797 may not be applied toward the degree.

Course	Title	Credits
CSCE 799	Thesis Preparation	1-12

Total Credit Hours 1-12

Thesis Option

Students who choose the thesis option may substitute 6 hours of thesis preparation (CSCE 799) for electives. In addition, students must complete at least 12 hours in CSCE courses numbered 700 and above, and defend the thesis in a public presentation.

Non-Thesis Option

Students who choose the non-thesis option must complete at least 15 hours in CSCE courses numbered 700 and above, and pass a written comprehensive examination offered at the end of Fall and Spring semesters.