

# SCIENCE MATHEMATICS AND TECHNOLOGY - COMPUTER SCIENCE CONCENTRATION - FOR STUDENTS MATRICULATED BEFORE JAN. 1, 2009

Computer science focuses on aspects of the computer field that have a sufficiently well-defined set of abstract concepts and principles to be characterized as "scientific." The objective of a concentration in computer science is to master those abstract concepts and principles — the theory of the field — and to implement, demonstrate and test this theory via a computer.

A concentration in computer science would be a good choice for someone interested in computers and computing, including the general principles of computing and in how computers and computer programs work most efficiently.

Computer science is still a relatively "young" science, and its theory, while powerful, provides only limited help in finding usable solutions for many common problems. Therefore, computer science, as a disciplinary concentration, probably would not be the best choice for someone primarily interested in specific, real-world problem solving, particularly in the business world. For such individuals, concentrations in computer information systems, applied computer systems, computer programming, computer applications systems or computer studies would generally be more worthwhile.

## Mathematics

Mathematics has provided much of the context from which the field of computer science has been emerging. Because mathematics continues to provide this background, as well as a variety of specific, immediately usable tools and concepts for computer scientists, intensive study of mathematics is required as part of the concentration in computer science.

This includes:

- Traditional math, based upon continuous variables (including algebra, trigonometry, analytic geometry, calculus).
- More recent understanding of discrete or finite (non-continuous) sets, groups, trees, graphs, matrices, linear programming, linear algebra.

Until recently, only a few topics from this latter tradition of discrete mathematics have been included in the school mathematics courses taken by most students. Thus, adult students returning to college often have not studied much, if any, discrete mathematics.

Because of the central and prerequisite position of discrete mathematical ideas within the field of computer science, study of discrete math is required. If a student does not have prior learning in discrete mathematics, study should begin in an early contract (though not necessarily the first).

The recommendations for formulating concentrations in computer science are summarized below.

### A. The degree program should include study of the following areas:

1. Mathematics:
  - Differential and integral calculus (8 or more credits).** Inclusion of concepts and computational techniques related to computers (e.g., Newton's algorithm series) is recommended. Differential equations is recommended, but not required. Knowledge of algebra, trigonometry and analytic geometry (sometimes called precalculus) is necessary to begin calculus; students should include a contract in these prerequisites if they do not have sufficient previous mathematics background.
  - Discrete mathematics (at advanced level, requiring understanding of axioms, theorems and proofs).** This may require study of discrete math at introductory level (involving primarily concepts and techniques) first for students who have not encountered the subject previously.
2. Structured programming in at least one language, preferably two.
3. Data structures (including at least an introduction to runtime analysis and efficiency, or a separate, more thorough component in algorithm analysis).
4. Computer hardware, architecture, organization, and/or assembly language programming.

### B. The degree program should also include study of at least two of the following areas, or of closely related subjects.

1. Operating systems.
2. Database principles, file processing.
3. Analysis of algorithms.
4. Artificial intelligence.
5. Numerical methods, statistical and scientific computing.
6. Communications and/or networks.
7. Principles of programming languages.
8. Theory of computation.
9. Compilers.
10. Simulation.
11. Software engineering.
12. Computer graphics.
13. Logic and switching theory.

### C. The degree program should include an explicit, real-world project of some sort, in which the student is involved in the planning, implementation and evaluation of a practical application involving a specific, working hardware and software system.

This component can be part of or grow out of one of the other above components.

### Note:

- Any of all of the above subjects can appear as part of the advanced-standing or contract-learning sections of the degree program.
- Mentors may wish to encourage students to explain in their degree-program rationale how their program meets the guidelines, with specific reference to the contracts or components that include the study areas in A, B and C above.