CMSI 182

Introduction to Computer Science

Fall 2007

Computer Science Topics

The introductory nature of this course allows us to "customize" it somewhat. Want to learn about something right away? Want to save it for last? Here's your opportunity. Of course, as with any other discipline, subtopics do overlap and interact…I'll point those out whenever they come up.

Topics are listed in the order used by the textbook. Approximate durations for each topic are also included, though of course those are subject to change depending on level of interest and/or degree of difficulty.

Data Storage — 1 week

You may have heard that today's computers use 1s and 0s for everything from plain e-mail to the fanciest 3D effects…but *how???* This topic covers the secrets behind digital data.

Data Manipulation — 1 week

Data storage is one side of the "low-level" coin; *data manipulation* looks at how today's computers work with this data, including their most important parts and fundamental activities.

Operating Systems — 1 week

Working with bits is one thing, but writing a paper is another: *operating systems* bridge the gap between a computer's "primitive" activities and the work that we humans ask of them.

Networking and the Internet — 2 weeks

Bits cross boundaries through *networks*, and these days, the largest such network is the Internet. This topic covers how computers communicate with each other, and how this communication then helps *us* communicate better with each other.

Algorithms — 3 weeks

An *algorithm* defines a process for solving a problem or performing a task, in a manner that allows machines to do this work for us. It is computer science's primary object of study, and not surprisingly, it is this topic that gets the most time.

Programming Languages — 1 week

"Teaching" a computer how to perform an algorithm is perhaps the signature activity of computer science: *programming*. But computers don't "speak" the way we do, so we need special *languages* that bridge the gap between human and machine.

Software Engineering — 1 week

A program that performs a single task is one thing, but a full-fledged software system that fulfills the needs of an entire real-world activity is something else entirely. The computer science subfield of *software engineering* seeks ways to produce complex and effective software successfully and consistently.

Data Abstractions — 1 week

Many algorithms require that information be structured in certain ways — structures with unassuming names like *arrays*, *stacks*, *trees*, and more. Knowing what these *data abstractions* are and how to *implement* them is an essential computer science skill.

Database Systems — 1 week

Data never stop growing, so the study of how to store and manage databases while keeping them fast, secure, *and* easy to search at the same time is an ever-relevant computer science subfield.

Artificial Intelligence — 1 week

After seeing what computers can do and how they work, one can't avoid comparing these machines to ourselves. *Artificial intelligence* is the subfield of computer science that formally studies and experiments with this idea.

Theory of Computation — 1 week

Are there "eternal truths" in computer science? Are there limits to what can be computed? What aspects of computation can be proven with rigor and logic, regardless of the latest technology? If any of these questions intrigue you, *theory of computation* is your computer science subfield of choice.