

Class 54: What is Computer Science, Revisited?

Held: Tuesday, 5 December 2006

Summary: Today we begin the conclusion of CS by looking beyond this course to what kinds of topics computer scientists often study.

Related Pages:

- EBoard.

Notes:

- EC: The final Tuesday Extra of the semester is Thursday at noon.
- I anticipate distributing grades tomorrow.

Overview:

- What is CS?
- Subfields of CS.
- Related Disciplines.

What is CS?

- We started the course with this question.
- We therefore start to end the class with a similar question.
- I've given you a simple definition: *Computer science is the study of algorithms and data.*
- We've seen a bit what an algorithm is and what data are.
- But how do we study them?
- How have we studied them in this course?

Subfields of Computer Science

- In part, the ways in which we study algorithms lead to different subfields of computer science. The domains we study also lead to different classes of algorithms.
- Computer scientists who emphasize *organization and architecture* study the ways in which algorithms and data may be implemented in hardware and the implications of particular implementations.
 - Example: Representing integers and adding integers.
 - Grinnell's course: CSC211
- Computer scientists who emphasize *operating systems* study algorithms and data representations that permit programs to use the common resources (file system, processor, mouse, etc.)
 - Example: Dining philosophers problem.

- Grinnell's course: CSC213
- Computer scientists who emphasize *software engineering* look at the design of large computer applications. They may consider process, program segmentation, team aspects, or even social aspects.
 - Example: Waterfall vs. Agile
 - Grinnell's course: CSC223
- Computer scientists who emphasize *artificial intelligence* look for ways to either model the way the brain works or to build alternate simulations.
 - Example: Genetic programming.
 - Grinnell's course: CSC261
- Computer scientists who emphasize *algorithms* tend to look for interesting problems for which to design algorithms. They also investigate general algorithm design strategies and prove things about the characteristics of problems and algorithms.
 - Example: Lower-bound on the running time of a sorting algorithm.
 - Example: The traveling salesman problem.
 - Grinnell's course: CSC301
- Computer scientists who emphasize *programming languages* look at the design of the languages in which we express algorithms.
 - Example: Paradigms
 - Grinnell's course: CS302
- Computer scientists who emphasize *theory* consider models of computation and the limits of these models.
 - Example: The halting problem.
 - Example: The Church-Turing thesis.
 - Grinnell's course: CSC341
- Computer scientists who emphasize *computer graphics* write algorithms related to that problem domain.
 - Example: Ray tracing.
 - Grinnell's course: Special Topics
- Computer scientists who emphasize *human-computer interaction* consider the relationship of the programs we write to the people who use them.
 - Example: Menu ordering.
 - Grinnell's course: Special Topics
- Computer scientists who emphasize *databases* look carefully at the models for representing and accessing large amounts of data.
 - Example: Relations
 - Grinnell's course: Special Topics
- Computer scientists across the board consider *social and societal implications of computing*, although some focus more on this area than others. (And yes, some make it their specialization, and we still consider them computer scientists.)

Related Disciplines

- There are many fields that are like computer science, but differ a bit.
 - Here are some variants that you may hear.
 - *Computer Engineering* emphasizes the construction of computational devices.
 - *Software Engineering* (mentioned above) emphasizes the construction of software, particularly processes, using some of the approaches of the discipline of engineering.
 - *Informatics* is either (a) a better, European, name for CS or (b) the application of CS to a particular problem domain.
 - *Information Science* is the study of information. It tends to focus more on representation or the softer sides of CS.
 - *Computer Programming* is a profession not always closely related to CS.
-

Copyright © 2006 Samuel A. Rebelsky. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.