

# Restructuring the Introductory Computer Science Course with Topics from AI

**Timothy Huang**

Department of Computer Science  
Middlebury College  
Middlebury, VT 05753  
huang@middlebury.edu

## Abstract

The traditional CS1 syllabus focuses almost entirely on elementary programming concepts and leaves little time to explore the broad discipline of computer science, including the many exciting developments in artificial intelligence. Hence, some students who take the CS1 course develop the misconception that computer science involves little more than programming, and they decide not to study it any further. We have developed a unified approach to CS1 that integrates aspects of the traditional CS0 and CS1 syllabi. This approach draws examples from AI to help illustrate the enormous potential for and broad societal impact of advances in computer science.

## Rethinking CS1

A typical introductory computer science curriculum offers a focused CS1 course for potential majors that covers elementary programming concepts, along with a general CS0 course for non-majors that provides a broad introduction to the discipline of computer science (LACS 2007). Although AI provides a rich source of examples that convey the excitement of studying computer science and the opportunities available to those who do, the traditional CS1 syllabus leaves little room to explore real-world applications or the broad impact of computer science on society and culture. Hence, some potential majors who take CS1 develop the misconception that studying computer science involves little more than developing increasingly larger and more complex programs, and they turn their attention to other fields.

At Middlebury College, a small liberal arts college in Vermont, we hope to change the way in which students perceive computer science by providing a single introductory course that integrates content from the traditional CS0 and CS1 syllabi (Middlebury 2007). In selecting syllabus topics, we focused on those that we would want all students to see, regardless of whether they

took additional computer science courses in the future. Besides covering introductory programming concepts using the Java language, we also discuss topics such as the digital representation of media, the structure of computer hardware and operating systems, applications of computing in medicine and business, and social concerns of computing.

Like other fields such as psychology or economics, this approach allows students to take introductory computer science without having to decide a priori whether they are “CS1” types or “CS0” types. Students who choose not to study computer science any further will have been exposed to the key ideas and societal implications of the discipline, as well as the beauty and rigor of computer programming, while those who choose to major or minor in computer science will have gained not only a sound foundation for future study but also a broader perspective that they would have gotten with our previous curriculum only after looking back and synthesizing topics sprinkled throughout their computer science courses.

Of course, this approach means that we cannot cover all of the topics in the typical CS1 syllabus. By eliminating some topics, particular those that involve GUI and graphics programming, and by moving other topics to the syllabi of other courses in our curriculum, primarily the CS2 course, our new CS1 course covers almost all of the AP Computer Science A syllabus (College Board, 2008) during about 60% of the lectures and covers broader computer science topics in the remaining 40%.

## Motivating CS1 with Topics from AI

The CS1 course at Middlebury College consists of three 50-minute lectures and one 75-minute lab section per week for 12 weeks. There are typically 15 to 25 students per lecture section, and about 10 students per lab section. With this schedule and with the flexibility afforded by our new syllabus, we devote three or four of the 36 lectures and one of the 12 labs to AI-related topics. We select these topics to highlight the excitement of working in computer science

today, to expose the diversity of opportunities available for computer science graduates, and to address the wide-ranging societal impact of the rapid advances in technology.

In one lecture, we survey some real-world AI applications and the key ideas underlying their development. Such applications have included the Deep Blue chess-playing computer, the Google search engine, the Sportvision technology for superimposing a “1st and Ten” line on the playing field in televised football games (Sportvision 2007), face detection in digital cameras, the MIT robots Cog and Kismet, the RoboCup project, and the DARPA Grand Challenge.

In another lecture, we focus on some of the philosophical implications of AI. We summarize the Turing Test, Searle’s Chinese room argument, and some ideas from films such as *Artificial Intelligence*, *The Iron Giant*, and *I, Robot*. We then turn the lecture into a large discussion section by inviting students to express and defend their own thoughts on what it means to be intelligent, to be conscious, and to be human.

Throughout the course, we try to refer to the societal impact of computer science whenever appropriate. Nevertheless, we also devote one or two lectures to issues such as privacy and security. For example, we discuss how data mining enables companies to more effectively recommend specific products to individual consumers and how face recognition systems may help improve security at airports. At the same time, these kinds of systems may infringe on privacy rights. In the ensuing discussion, we encourage students to consider how knowledge or ignorance of the underlying technology affects one’s ability to address potential problems associated with it.

In the weekly lab session devoted to AI, students interact with programs that play the game of Hex, that recognize handwritten characters, that play the game of “twenty questions,” and that attempt to carry on a written conversation. The lab work requires students to explore the strengths and limits of each program and consider how they compare to each other. For example, we encourage students to try playing against the Hex player until they can beat it, and we ask them to record the most sensible and least sensible comments generated in their interactions with the chatterbot. At the end, we have students choose which application seems to exhibit the most “intelligence.” This, of course, challenges them to think about what it means to be intelligent and how to test for intelligence.

All students choose a final project that involves either research into some area of computers and society or more advanced programming. For the programming option, students implement a version of the Sokoban maze game (Bruce, et.al. 2005). For extra credit, we invite students to construct a computer solver for the game. While we don’t

expect students to tackle such an advanced AI problem, we hope the enthusiasm generated by writing the game will encourage them to consider the algorithmic challenges involved in constructing a computer player.

## Initial Assessment

The restructuring of our introductory curriculum is still in trial mode, but we are pleased and encouraged by our initial efforts. In course evaluations, many students welcomed the balance of breadth-based topics and programming concepts, and some specifically identified the AI-related lectures and lab as especially stimulating.

Our experiences to date have reinforced our belief that a unified CS1 course provides a suitable introduction for all students to the field of computer science. Regardless of whether they take future computer science courses, students will be exposed to the challenge, power, and beauty of designing and implementing algorithms, as well as the wide-reaching impact of computer science and the myriad opportunities available for those who study it.

Moving forward, we plan to continue identifying ways in which the traditional CS0 and traditional CS1 material relate to each other, and we plan to regularly update our discussion of real-world applications with new developments in AI and other areas.

## Acknowledgements

Special thanks to Amy Briggs, Matt Dickerson, Bob Martin, and Daniel Scharstein, all colleagues in the Middlebury College Department of Computer Science, for insightful discussions and feedback, and to Amy Briggs for helping to develop the first offering of the new CS1 course.

## References

- Middlebury College. 2007. CSCI 101: The Computing Age, at <http://www.cs.middlebury.edu/~cs101>.
- Liberal Arts Computer Science Consortium (LACS). 2007. A 2007 Model Curriculum for a Liberal Arts Degree in Computer Science, in the *ACM Journal on Educational Resources in Computing*, Vol. 7, No. 2, Article 2.
- College Board, The. 2008. AP Computer Science A, at [http://www.collegeboard.com/student/testing/ap/sub\\_comp\\_scia.html?comp\\_scia](http://www.collegeboard.com/student/testing/ap/sub_comp_scia.html?comp_scia).
- Sportvision, Inc. 2007. 1st and Ten Line, at <http://www.sportvision.com>.
- Bruce, K., Danyluk, A., and Murtagh, T. 2005. Sample lab exercises at <http://eventfuljava.cs.williams.edu/sampleLabs.html>, for *Java: An Eventful Approach*. Prentice-Hall.