

Defense of Intro (say CS 180) as an LAS course:

Intrinsic value:

Since about the middle of the 20th century, the word "computer" has been used to refer to general purpose devices that can be programmed to perform logical and arithmetic operations. Prior to that time the word was used to refer to humans (most often women, by the way) who performed repetitive calculations in the service of producing numeric tables or summarizing business data.

This class is an introduction to the fundamental ideas involved in programming computers to solve problems. Rather than just solving the problems directly, the students are taught to think carefully through the steps required for problem solving and describe those steps in a formal language. These formal descriptions take the form of "algorithms" for solving particular kinds of problems. It is similar in many ways to describing a cooking recipe for someone who has no familiarity with even the most basic cooking techniques. It is also similar to the rigor and formalism required of mathematics students when they are learning to write effective proofs.

It is an enlightening experience to have to describe in precise detail exactly how to perform a task to an agent that will do no more and no less than what you ask it to do. Learning to do this opens student minds to the ideas of how to think like a computer scientist, meaning (among other things) how to think in a precise, formal way.

Practical intellect:

This class requires and amplifies students critical thinking skills. Figuring out how to take a few simple programming constructs (sometimes called design patterns, especially as they get more complex) and combine them in intricate patterns to solve complex problems requires significant thought.

Documenting programs by writing non-executable comments to explain program performance to humans is absolutely necessary and requires the development of an ability to explain complex technical systems in understandable ways. This class emphasizes the development of these skills.

Building character:

Pair programming (which is pretty much what it sounds like, students working in pairs on programming tasks) has been shown to be helpful for reducing student anxiety and increasing productivity, but it also requires a level of tolerance of differences, willingness to listen

carefully, and cooperation that helps build student character.

Access to the Internet gives students the ability to find many resources online that might help them solve assigned problems, or might just allow them to copy a program with only minor changes. Addressing this ethical question is a big part of teaching them to be responsible for their own learning and avoid copyright and plagiarism issues.