MLA 321 — Course Information

Professor: Eric Roberts

Email:	`eroberts@cs.stanford.edu`
Offices:	Gates 202
Phone:	723-3642
Drop-in hours:	Tue 9:30–11:00; Wed 4:00–5:00

Course description

MLA 321. Great Ideas in Computer Science—Covers the intellectual tradition of computer science emphasizing the ideas that have enabled the most important milestones in the history of the discipline. Topics include programming and problem solving; implementing computation in hardware; algorithmic efficiency; the theoretical limits of computation; cryptography and security; and the philosophy behind artificial intelligence. No prior experience with programming is required.

Class meetings

This seminar is scheduled for Mondays and Wednesdays from 7:00 to 9:30 in 200-105. In a small seminar, regular attendance is essential.

Readings

Most of the reading for this class consists of chapters from the book I'm writing for people interested in the intellectual foundations of computer science. I'm handing out the current draft of the book as a reader along with the wonderful E. M. Forster short story "The Machine Stops" and essays by Vannevar Bush and Alan Turing.

Course requirements

The required work for this course consists of the following:

Class participation Most of what you learn in the seminar will take place during the seminar meeting, so it is critical that you take an active role. You are expected to complete the assigned reading and to think about the issues raised by those readings so that you can participate fully.

Weekly assignments In the first five weeks, there will be a short assignment given out each Tuesday and due the following Tuesday. The goal of these assignments is to make sure that you are actively engaged in the topics and not simply reading about them. Concepts that you think you understand from the reading don't really sink in until you've had a chance to put them into practice.

Final project The single most important component of the course is the final project, in which you investigate in detail some aspect of computer science that seems to qualify as a "great idea." The deliverables include a paper that explains your topic to someone with no more

background than this class along with a 15-minute presentation at the last class session on March 8. I will describe the project assignment in more detail and offer several possible topic ideas in a separate handout distributed on January 26.

Grading

Final grades for the course will be determined using the following weights:

25% Class participation
35% Weekly assignments
40% Final project

Week-by-week schedule

January 5	Introductions; course overview; Babbage machines; Ada Lovelace *Readings:* Chapter 3
January 12	Karel the Robot; beginning JavaScript; algorithms *Readings:* Chapters 1 and 2
January 19	Binary arithmetic; digital logic *Readings:* Chapters 4 and 5
January 26	Stored-program machines; the Toddler machine *Readings:* Chapter 6
February 2	Turing machines; the Busy Beaver problem; undecidability *Readings:* Chapters 8 and 9
February 9	Computational complexity; the P = NP question *Readings:* Chapters 7 and 10
February 16	Cryptography; public-key cryptography; digital signatures *Readings:* Chapters 11 and 12
February 23	Networking; networking algorithms; the Google page-rank algorithm *Readings:* Vannevar Bush, "As We May Think"
March 1	Artificial intelligence; promise and peril of the digital age *Readings:* Alan Turing, "Computing Machinery and Intelligence"; E. M. Forster, "The Machine Stops"
March 8	Final presentations