

CMSI 182

INTRODUCTION TO COMPUTER SCIENCE

Fall 2009

Midterm Review Sheet

The midterm will take place as scheduled, on October 8. It will be open book, notes, and handouts, but not open computer. This guide should help you to prepare for it properly.

Covered Material

The midterm covers Chapters 0 and 1 in the Brookshear textbook. Coverage also includes all handouts and sample code that have been distributed in support of this content, plus prominent material from class discussions and activities.

Covered Objectives

The questions on the midterm are meant to measure where we are in terms of the following course objectives based on L. Dee Fink's *taxonomy of significant learning*, as presented in the syllabus and on the first day of class:

- How well you remember and understand the central ideas discussed so far — at this point, we have covered a brief history of computer science, a variety of algorithms, a mechanism for expressing these algorithms so that a computer can perform them (JavaScript), and a study of how today's machines represent information
- How easily you recognize these concepts in your daily life, in current events, and in new products: can you see algorithms in the things you do everyday? In the news? In other courses?
- Do you see how the need to represent information solely as **1s** and **0s** might influence certain machine behaviors or idiosyncrasies, such as capitalization or exact punctuation?

Sample Tasks and Questions

With these objectives in mind, these are the types of tasks and questions that are fair game for the midterm. Questions and tasks are meant to be *forward-looking*; that is, they represent the kinds of things that I hope you'll still know and/or be able to do loooooong after this course ends:

- The big picture view of computer science — what it is about, what it studies, what its concerns are, and what its [current] tools are
- A historical overview of computer science — key people, events, and milestones, their significance, and their general historical vicinity (as in, I hope no one confuses “ancient Babylonian algorithms” as pertaining to the 20th century!)
- Given some [simple] problem or task, specify an algorithm that solves that problem, possibly in plain English or [basic] JavaScript
- Given an algorithm, possibly in plain English or [basic] JavaScript, figure out and explain what it does (i.e., what information it requires, what computation it performs)
- Describe or use some algorithmic concept that has been covered so far, such as assignment (=), loops (*while*), conditions (*if*), and lists ([])
- Given some [simple] scheme for representing information as **1s** and **0s**, convert information from one form to the other
- Apply some known *standard* way for representing information as **1s** and **0s** (binary, ASCII, etc.) to convert information from one form to the other
- Ethical, philosophical, or societal considerations relating to any of the topics covered so far

Questions concerning these topics may range from brief explanations to applications of these ideas to real-world problems and/or situations.