CS 101 - Introduction to Computing
Semester, year
Section, Time, Room #

Your name and Phone Number
Office and Office Hours
E-Mail address (and Course Web Page address if available)

Course Objective. The goal of this course is to teach students with no prior experience in computing to think like computer scientists.
The course is designed to enable students to appreciate, use and understand ideas at the core of computer science. This course covers ideas that will be useful and interesting to students, whether or not they major in computer science.

Course Description. In the past hundred years, computer science has changed the world more than any other field. Without nuclear Physicists, World War II may have lasted longer, but without computer scientists the other side may have won. Without computer science, humans would not have walked on the Moon, modern medicine would not exist, and Wal-Mart would be a small store in Arkansas.

But this course is not about the pragmatic impact of computer science; it is about something even more fundamental: how computer science changes the ways we think, solve problems and understand the world. Despite its name, computer science has very little to do with the beige boxes we call computers, and it is far from being a science. It has more in common with music and mathematics than it does with science or engineering. At its core, computer science is the study of imperative knowledge. Whereas mathematics is all about declarative knowledge ("what is"), computer science is all about "how to" knowledge.

Computer science is the study of how to describe and predict properties of information processes. Most of what we know about describing information processes stems from three simple ideas:

1. You can define things in terms of themselves (recursive definitions).
2. You can treat procedures and data as one and the same (first class procedures).
3. When you give something a name, it becomes more useful (abstraction).

Although these ideas are simple, they have profound implications that it takes many years to fully appreciate.
The kinds of properties we want to predict about information processes include whether or not there is a procedure that can always solve a given problem (computability), and how much time and space will be required to solve a given problem (complexity).
Expected Background: This course is open to students with no prior background in computer science or programming, but appropriate and challenging for experienced CS majors also. The only background we assume is:

- Language: reasonable proficiency in reading and writing English
- Math: understanding of whole numbers and addition, subtraction, multiplication, divison and exponentiation.
- Logic: familiarity with logicial and and or and not.
- Computer Literacy: ability to use email and browse the web.

Meetings: Mondays, Wednesdays and Fridays at 2:00-2:50 pm in Hannon 341.
Books: There are two required books for this course:

- Structure and Interpretation of Computer Programs ("Wizard Book"), by Harold Abelson and Gerald Jay Sussman with Julie Sussman (referred to as SICP) [MIT Press] [Amazon]
- Gödel, Escher, Bach: An Eternal Golden Braid, by Douglas R. Hofstadter (referred to as GEB) [Amazon]