

 <h1>COMPUTER SCIENCE</h1> <p>An Interdisciplinary Approach</p> <p>ROBERT SEDGEWICK KEVIN WAYNE</p>	
INTRO TO PROGRAMMING	
1. Elements of Programming	
2. Functions	
3. OOP	
4. Data Structures	
COMPUTER SCIENCE	
5. Theory of Computing	
6. A Computing Machine	
7. Building a Computer	
BEYOND	
8. Systems	
9. Scientific Computation	
RELATED BOOKSITES	
	
WEB RESOURCES	
FAQ	
Data	
Code	

This page provides information about online lectures and lecture slides for the course. These materials are appropriate for use by instructors as the basis for a “flipped” classroom.

Flipped classroom. If you are an instructor teaching introductory computer science at a weekly cadence, as follows:

- Each week, send an email to all students in the class that briefly describes the material (from this booksite).
- Students watch the lecture videos at their own pace, do the reading assignments, and work on exercises.
- Schedule a weekly “class meeting” for discussion of the material, review of exercises, and problem sets.

This is just one suggestion—this material can support many different teaching styles.

Important note: A common mistake in teaching a flipped class is to add too much reading for students for success on programming assignments and exams. If an instructor expects students to read, most students will do so. Class meetings then can involve interactive problem sets with potential exam questions is an excellent activity.

Self-study. An effective way to learn the material on your own is to watch the lecture videos, do the exercises in the book or on the booksite on your own. If you get stuck on an exercise, look at the solutions there.

The lecture videos are available for purchase through the course website. If you are bored; if it is too fast, you are likely to get lost. A

Lecture videos are available in .pdf format by clicking the

Dialogue—A Simple Machine. This lecture introduces the story of computer science. The story motivates the study of computer science throughout the course.

Lecture 1: Basics. Why program? This lecture addresses that question by writing a program in Java using either virtual terminals or a program development environment. Java's built-in data types, with example programs for each.