Computer Science 50: Introduction to Computer Science I
Harvard College
Fall 2007

Syllabus

v. 1.0.15

Instructor

David J. Malan
`malan@post.harvard.edu`
+1-617-523-0925

Description

Introduction to the intellectual enterprises of computer science. Algorithms: their design, specification, and analysis. Software development: problem decomposition, abstraction, data structures, implementation, debugging, testing. Architecture of computers: low-level data representation and instruction processing. Computer systems: programming languages, compilers, operating systems. Computers in the real world: networks, security and cryptography, artificial intelligence, social issues. Assignments include extensive programming in the C language and PHP.

This course, when taken for a letter grade, meets the Core area requirement for Quantitative Reasoning.

Prerequisites

None. CS 50 does not assume any prior programming experience.

Motivation

"Demanding, but definitely doable. Social, but educational. A focused topic, but broadly applicable skills. CS 50 is the quintessential Harvard course."

Computer Science 50: Introduction to Computer Science I is a first course in computer science at Harvard College for concentrators and non-concentrators alike. More than just teach you how to program, this course teaches you how to think more methodically and how to solve problems more effectively. As such, its lessons are applicable well beyond the boundaries of computer science itself. That the course does teach you how to program, though, is perhaps its most empowering return. With this skill comes the ability to solve real-world problems in ways and at speeds beyond the abilities of most humans.

To be sure, CS 50 has a reputation that, each year, does scare some humans away (including, back in 1995, at least one member of the staff). The course indeed involves upwards of ten hours of work per week. But programming is simply one of those arts that you learn only by doing. The more you put into this course, the more you get out of it. Plus, with your final project, you might even leave your mark on campus.

Computer Science 50: Introduction to Computer Science I
Harvard College
Fall 2007

Table of Contents

Computer Science 50: Introduction to Computer Science I
Harvard College
Fall 2007

Expectations

You are expected to attend all lectures and sections, complete eight problem sets, take three quizzes, and produce a final project.

Grades

Final grades will be determined using the following weights.

Problem Sets (best 7 out of 8):	65%
Quizzes (best 2 out of 3):	20%
Final Project:	15%

You may take this course pass/fail. Realize, though, that some departments (Computer Science, Applied Math, *et al.*) do not allow courses taken pass/fail to satisfy concentration requirements. Moreover, you'll certainly learn more by immersing yourself in this course, in which case you might just want to take it for a grade.

`nice.fas.harvard.edu`

If you haven't one already, this course requires that you obtain an FAS (Faculty of Arts and Sciences) Computer Account by visiting the URL below and following the on-screen instructions.

```
https://www.fas.harvard.edu/computing/utilities/activate-pin/
```

This account will allow you to access via SFTP and SSH `nice.fas.harvard.edu`, FAS's New Linux Computing Environment, use of which will be taught by the course and required by the course's problem sets.

Website

The address of this course's website appears below.

```
http://www.fas.harvard.edu/~cs50/
```

To save a few keystrokes, you can also use the address below; you'll be automatically redirected to the above.

```
http://cs50.org/
```

Visit the course's website to access the course's bulletin board, enter the virtual terminal room, check your grades, watch digital videos of lectures and supersections, download handouts and software, and follow links to other resources.

Staff

To pose a question or comment to the staff, email the address below.

```
cs50@fas.harvard.edu
```

However, you are encouraged to post most questions to the bulletin board on the course's website instead so that others might benefit as well.

Lectures

Lectures will take place in Sever Hall 113 on most Mondays, Wednesdays, and Fridays from 10:00 A.M. until 11:00 A.M.

Each lecture will be filmed and made available within 72 hours via podcast (for download to iTunes and iPods) and via the course's website in Flash, MP3, and QuickTime formats. Once posted, these recordings will remain available until semester's end. You are welcome to watch or listen to a recording if you are unable to attend some lecture in person. You are encouraged to watch or listen to these recordings for the sake of review.

A schedule of lectures, subject to change, appears below.

Week 0
Monday, 17 September 2007 – Friday, 21 September 2007

Introduction. Bits. Binary. ASCII. Programming. Algorithms. Scratch. Statements. Boolean expressions. Conditions. Loops. Variables. Threads. Events. C.

Week 1
Monday, 24 September 2007 – Friday, 28 September 2007

C, continued. Source code. Compilers. Object code. SSH. SFTP. GCC. Functions. Comments. Standard output. Arithmetic operators. Precedence. Associativity. Local variables. Types. Casting. Standard input. Libraries. Boolean expressions, continued. Conditions, continued. Loops, continued.

Week 2
Monday, 1 October 2007 – Friday, 5 October 2007

Functions, continued. Global variables. Parameters. Return Values. Stack. Frames. Scope. Arrays. Strings. Command-line arguments. Cryptography.

Week 3
Monday, 8 October 2007 – Friday, 12 October 2007

No lecture this Monday.

Linear search. Binary search. Asymptotic notation. Recursion. Bubble sort. Selection sort. Merge sort.

Week 4
Monday, 15 October 2007 – Friday, 19 October 2007

Quiz 0 this Wednesday.

Debugging software. Designing software.

Week 5
Monday, 22 October 2007 – Friday, 26 October 2007

Structures. Dynamic memory allocation. Pointers. Heap. Digital forensics. File I/O.

Week 6
Monday, 29 October 2007 – Friday, 2 November 2007

Linked lists. Hash tables.

Week 7
Monday, 5 November 2007 – Friday, 9 November 2007

Quiz 1 this Wednesday.

Binary search trees. Heaps. Heapsort.

Week 8
Monday, 12 November 2007 – Friday, 16 November 2007

No lecture this Monday.

Huffman coding. Preprocessing. Compiling. Assembling. Linking. CPUs. Ant-8.

Week 9
Monday, 19 November 2007 – Friday, 23 November 2007

No lecture this Wednesday or Friday.

Secure coding.

Week 10
Monday, 26 November 2007 – Friday, 30 November 2007

TCP/IP. HTTP. XHTML. PHP. SQL.

Week 11
Monday, 3 December 2007 – Friday, 7 December 2007

Designing Web-based software. Cybersecurity. Sneak preview of CS 51.

Week 12
Monday, 10 December 2007 – Friday, 14 December 2007

Quiz 2 this Wednesday.

Sneak preview of CS 61. Fun. Exciting conclusion.

Sections

Lectures are supplemented by weekly, 60- to 90-minute sections, each led by a teaching fellow. Sections provide you with an opportunity to review and discuss course materials in a more intimate environment, with only your teaching fellow and a handful of classmates present. Moreover, the teaching fellows often will supplement lecture materials with additional examples and implementation details as well as provide further guidance for problem sets and quizzes.

A schedule of sections appears on the course's website.

Supersections

Sometimes it helps to hear things twice, if not from different people.

In addition to their own sections, the teaching fellows will hold one "supersection" each week that students from all sections are welcome to attend. Each supersection will review material from the most recent section in greater detail, at a slower pace, and/or with additional examples.

Moreover, these supersections will also be filmed and made available within 96 hours via the course's website in Flash, MP3, and QuickTime formats. Once posted, these recordings will remain available until semester's end. You are encouraged to watch or listen to a recording if you are unable to attend some supersection in person.

A schedule of supersections appears on the course's website.

Office Hours

Throughout the week, the staff hold office hours in Science Center B14 (the "terminal room") during which you can receive hands-on, one-on-one assistance. The staff also hold virtual office hours in the course's "virtual terminal room," where you can receive remote, one-on-one assistance via the Web. Thanks to technology, the staff can help you troubleshoot bugs by observing or sharing control of your screen while chatting with you via IM or VOIP, whether you're in your dorm room, at Brain Break, at Starbucks, or beyond!

A schedule of office hours appears on the course's website.

Problem Sets

Eight problem sets will be distributed during the semester. Each will be due via electronic submission on `nice.fas.harvard.edu` seven or more days after its date of distribution. However, you have eight "late days" that you may "spend" during the semester, each of which provides you with an extension of twenty-four hours. In other words, you may submit one problem set eight days late, each problem set one day late, *etc*. It is expected that you will alert your teaching fellow to your use of any late days for some problem set prior to or upon its actual deadline. Lateness of electronic submissions will be determined down to the minute by those submissions' timestamps on `nice.fas.harvard.edu`. Submitting one minute late is equivalent to submitting twenty-four hours late. Late work will not be accepted once you have exhausted your eight late days, except in cases of emergency. Technical difficulties will not constitute emergencies. These late days cannot be spent on the course's final project.

In order to accommodate students with different backgrounds, some problem sets will be released in two editions: a standard edition intended for most students and a "Hacker Edition" intended for some students. Both editions will essentially cover the same material. But the Hacker Edition will typically present that material from a more technical angle and pose more sophisticated questions. Hacker Editions will be graded separately from standard editions, but those students who submit the former will not receive any form of extra credit outright. When awarding letter grades at term's end, however, we will bear in mind submissions of Hacker Editions.

To be clear, we encourage most students (including aspiring computer scientists) to tackle the standard editions. However, you may choose, week to week, which edition to submit. You may not submit both or some amalgam of the two.

When computing your final grade, we will drop your lowest score on these problem sets; your seven highest scores will be weighted equally.

A schedule of problem sets, subject to change, appears below.

Problem Set 0: Scratch
due by 7:00 P.M. on Friday, 28 September 2007

Create your own animation, game, or interactive art.

Problem Set 1: C
due by 7:00 P.M. on Friday, 5 October 2007

Meet Linux and C.

Problem Set 2: Crypto
due by 7:00 P.M. on Friday, 12 October 2007

Encrypt and decrypt sensitive information.

Problem Set 3: The Game of Fifteen
due by 7:00 P.M. on Saturday, 27 October 2007

Implement a childhood classic.

Problem Set 4: Forensics
due by 7:00 P.M. on Friday, 2 November 2007

Recover lost photos. Solve a murder mystery.

Problem Set 5: Mispellings*
due by 7:00 P.M. on Friday, 16 November 2007

Implement a spell-checker that's faster than your classmates'.

Problem Set 6: Huff'n Puff
due by 7:00 P.M. on Friday, 30 November 2007

Compress your own files.

Problem Set 7: C$50 Finance
due by 7:00 P.M. on Friday, 7 December 2007

Design a database. Build a website.

* Yes, we know.

Quizzes

The course will have three 53-minute quizzes. These quizzes will be "closed-book." However, you may utilize during each quiz one two-sided page (8.5" × 11") of notes, typed or written, and a pen or pencil, but nothing else.

A schedule of quizzes, subject to change, appears below; these quizzes will take place in lieu of lectures on these dates.

Quiz 0
Wednesday, 17 October 2007

Covers lectures through Friday, 5 October 2007.

Quiz 1
Wednesday, 7 November 2007

Covers lectures through Friday, 26 October 2007.

Quiz 2
Wednesday, 12 December 2007

Covers lectures through Friday, 30 November 2007.

You should expect the second and third quizzes to be cumulative, but their content will be based mostly on lectures subsequent to those covered by the first and second quizzes, respectively. Unless arranged with the staff in advance, quizzes may not be taken at alternative times even if missed by accident, except in cases of emergency.

When computing your final grade, we will drop your lowest score on these quizzes; your two highest scores will be weighted equally.

Final Project

Perhaps the most gratifying aspect of this course is its final project. The final project will be your opportunity to take your knowledge of programming out for a spin and develop your very own piece of software. So long as your project draws upon this course's lessons, the nature of your project is entirely up to you, albeit subject to the staff's approval. You may implement your project in any language(s), as long as the staff approves. You are welcome to utilize infrastructure other than `nice.fas.harvard.edu`, provided the staff ultimately has access to any hardware and software that your project requires.

All that we ask is that you build something of interest to you, that you make something useful, that you solve an actual problem, or, best yet, that you somehow impact campus. Strive to create something that outlives this course.

Inasmuch as software development is rarely a one-person effort, you will be allowed an opportunity to collaborate with one or two fellow students for this final project. Needless to say, it is expected that every student in any such group will contribute equally to the design and implementation of that group's project. Moreover, it is expected that the scope of a two- or three-person group's project will be, respectively, twice or thrice that of a typical one-person project. A one-person project, mind you, should entail more time and effort than is required by each of the course's problem sets.

Guidelines for the final project will be distributed by Wednesday, 14 November 2007.

A schedule appears below.

Pre-Proposal
due by 10:00 A.M. on Monday, 26 November 2007

Proposal
due by 10:00 A.M. on Monday, 3 December 2007

Design and Framework
due by 10:00 A.M. on Monday, 17 December 2007

Status Report
due by 7:00 P.M. on Friday, 4 January 2008

Presentation
sometime between Wednesday, 9 January 2008, and Friday, 11 January 2008, inclusive

Implementation and Documentation
due by 8:00 P.M. on Sunday, 13 January 2008

Extensions on this final project will not be granted, except in cases of emergency. Technical difficulties will not constitute emergencies. Problem sets' late days cannot be spent on this final project. Work submitted late without extension will penalized as follows: projects submitted up to one hour late will incur a penalty of 5%; projects submitted up to two hours late will incur a penalty of 10%; projects submitted up to three hours late will incur a penalty of 25%; projects submitted up to four hours late will incur a penalty of 50%; projects submitted more than four hours late will incur a penalty of 100%. Lateness of submissions will be determined down to the minute by timestamps.

Final Exam

This course will not have a final exam.

Books

No books are required for this course.

However, you may want to supplement your preparation for or review of some lectures with self-assigned readings relevant to those lectures' content from either of the books below. The first is intended for those inexperienced in (or less comfortable with the idea of) programming. The second is intended for those experienced in (or more comfortable with the idea of) programming. Both are available for purchase at the Coop and at sites like Amazon.com. Both of these books have been placed on reserve at Cabot Science Library and Gordon McKay Library. Realize that free, if not superior, alternatives to these books can be found among the course's online resources.

For Those Less Comfortable

Absolute Beginner's Guide to C, Second Edition
Greg Perry
Sams Publishing, 1994
ISBN 0-672-30510-0

For Those More Comfortable

Programming in C, Third Edition
Stephen Kochan
Sams Publishing, 2004
ISBN 0-672-32666-3

The book below is recommended for everyone, particularly those interested in understanding how their own computers work, for personal edification. At the Coop, this book is bundled at a discount with each of the books above. It is also available for purchase separately at sites like Amazon.com. It, too, has been placed on reserve.

For Everyone

How Computers Work, Eighth Edition
Ron White
Que Publishing, 2005
ISBN 0-7897-3424-9

This last book below is recommended for aspiring hackers, those interested in "programming tricks" and low-level optimization of code, for applications beyond the scope of this course. It is also available for purchase at the Coop and at sites like Amazon.com. It, too, has been placed on reserve.

For Aspiring Hackers

Hacker's Delight
Henry S. Warren Jr.
Addison-Wesley, 2003
ISBN 0-201-91465-4

Academic Honesty

All work that you do toward fulfillment of this course's expectations must be your own unless collaboration is explicitly allowed (*e.g.*, by some problem set or the final project). Viewing or copying another individual's work (even if left by a printer, stored in an executable directory, or accidentally shared in the course's virtual terminal room) or lifting material from a book, magazine, website, or other source—even in part—and presenting it as your own constitutes academic dishonesty, as does showing or giving your work, even in part, to another student.

Similarly is dual submission academic dishonesty: you may not submit the same or similar work to this class that you have submitted or will submit to another. Moreover, submission of any work that you intend to use outside of the course (*e.g.*, for a job) must be approved by the staff.

All forms of cheating will be dealt with harshly.

You are welcome to discuss the course's material with others in order to better understand it. You may even discuss problem sets with classmates, but you may not share code. In other words, you may communicate with classmates in English, but you may not communicate in, say, C. If in doubt as to the appropriateness of some discussion, contact the staff.