Computer Science 50

Software Design and Implementation - or, how to be a hacker

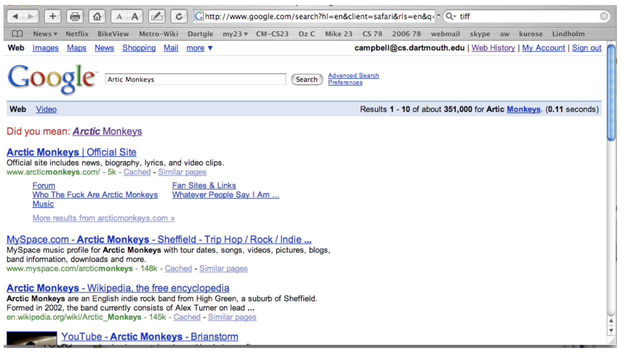
Robot racing on demo or die day with the hackers - Winter 2011. Photo by Joseph Mehling

Course

"How can I really call myself a CS major and not be a hacker?" - CS50 is your answer.

The ORC: Techniques for building large, reliable, maintainable, and understandable software systems. Topics include UNIX tools and filters, programming in C, software testing, debugging, and teamwork in software development. Concepts are reinforced through a small number of medium-scale programs and one team programming project.

More specifically: The course teaches Linux, C and shell programming, the use of GNU development tools (gcc, gdb, make, valgrind, gprofile, cvs) and design and implementation techniques. As part of the course you will design and implement the **TinySearchEngine** as part of a series of programming assignments - then you can take on google! The course includes a group project based on programming **embedded nanocopters and leap motion**. So you will implement a search engine and contol nanocopters using hand gestures! You will also be exposed to thread, socket and embedded systems programming. All in one course.


http://www.google.com/search?hl=en&client=safari&rls=en&q=
tiff
News Netflix BikeView Metro-Wiki Dartgle my23 CM-CS23 Oz C Mike 23 CS 78 2006 78 webmail skype aw kurose Lindholm
Web Images Maps News Shopping Mail more
campbell@cs.dartmouth.edu | Web History | My Account | Sign out
Google
Artic Monkeys
Search
Advanced Search
Preferences
Web Video

Results **1** - **10** of about **351,000** for **Artic** Monkeys. (**0.11** seconds)

Did you mean: ***Arctic*** Monkeys

Arctic Monkeys | Official Site
Official site includes news, biography, lyrics, and video clips.
www.arctic**monkeys**.com/ - 5k - Cached - Similar pages
Forum
Who The Fuck Are Arctic Monkeys
Music
Fan Sites & Links
Whatever People Say I Am ...
More results from arcticmonkeys.com »

MySpace.com - **Arctic Monkeys** - Sheffield - Trip Hop / Rock / Indie ...
MySpace music profile for **Arctic Monkeys** with tour dates, songs, videos, pictures, blogs, band information, downloads and more.
www.myspace.com/arctic**monkeys** - 148k - Cached - Similar pages

Arctic Monkeys - Wikipedia, the free encyclopedia
Arctic Monkeys are an English indie rock band from High Green, a suburb of Sheffield. Formed in 2002, the band currently consists of Alex Turner on lead ...
en.wikipedia.org/wiki/Arctic_**Monkeys** - 145k - Cached - Similar pages

YouTube - **Arctic Monkeys** - Brianstorm

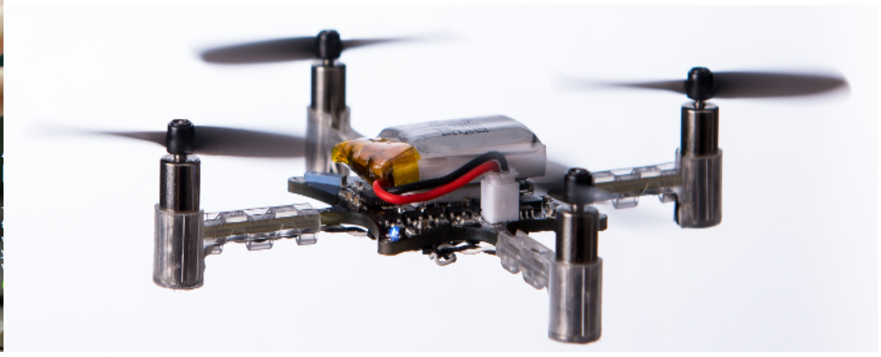
Note, there is a significant amount of programming in this course requiring a significant time commitment on the part of the student.

The course breaks down into seven/eight weeks of lectures and labs with the remaining time held over for group projects. The projects will be organized around a design review, code review and demo or die event. Something similar to what you would experience in the software industry when you have to ship a beta.

For a summary of the course see syllabus.

What you will learn

Some Linux, bash shell programming, C programming, programming tools, how to design, implement and debug a complex software system by working together in small groups. We hope that the skills and experience gained will reflect what you would expect in the software industry when working in small teams. The two main programming challenges are the TinySearchEngine specifically developed for this Dartmouth CS class - which represents a complex standalone software system - and real-time distributed client/server programming using the robots. The group project is a key part of the CS50 experience. We program embedded devices, specifically, embedded nanocopters and leap motion.



The nanocopters run RTOS. We control these tiny copters using the leap motion contoller (left figure). So with a swipe of your hand (left figure) you can launch the nacocopter (right) into the air and control it using the c code you write on your laptop and on the copter.

We will be flying our nanocopters "hand free" on the Green -- but don't tell anyone or I'll get fired.

Before nanocopters we raced robots in Sudikoff.

This is a fun, exciting but challenging class. Note, there are no midterm or final exams - just design, programming, and a lot of debugging; you will learn all about segfaults, memory leaks, and core dumps. Also, we don't use any fancy IDEs (e.g., Eclipse - integrated development environment) **in this class - everything is on the command line, no IDEs**. We expose details to gain knowledge rather than promote abstraction.

In brief, C50 is the class I never had as an undergrad - it's a crash course in design, C/Unix programming, process/threads, GNU tools, socket programming and the implementation of complex standalone and embedded software systems - it's a course where you will write approximately 3000 lines of C and 500 lines of bash script for the programming assignments and 1000 lines for the joint project; as a result, you will gain tangible programming skills; it will turn you into a systems hacker or your money back!

When and where

10-11.05 am Monday, Wednesday, Friday (Life Sciences Center Room 200)

Tu 12:00-12:50 pm X-hour Thursday   (Life Sciences Center Room 200)

Lab hours in L001 Sudi: Thursday 4-7 PM, Friday 4-7 Saturday 4-7 PM, Sunday 6-10 PM

We are using Piazza for the class for all announcements and posts. If you are not signed up you should contact the TA.

Team

Instructor: Andrew T. Campbell
campbell AT cs.dartmouth.edu
260 Sudikoff
Office Hours: Monday and Friday 4-5 PM.

Teaching assistants:

Fanglin Chen (TA)
Fanglin.Chen.GR@dartmouth.edu

Tianlong Yun (TA)
tyun@cs.dartmouth.edu

Section leaders: Ethan Yu, Mehdi Oulmakki, Shuyang Fang, Robin Wang, Chandrasekar Ramesh, Samuel Tan Jun Jie

Our CS50 hackers.

Grading

75% - Laboratory exercises

There are 7 weekly laboratory assignments over the first 8 weeks. These labs are designed to help you learn the languages, tools, and design skills you will need for your final project. Only six labs are graded. Some labs are harder than others and we grade as shown below. These assignments are to be done individually. The schedule is online - plan a head. You need to be organized to get through the labs and stay on schedule.

0%	Lab1 - Shell Commands (Not graded)
10%	Lab2 - Shell Programming
10%	Lab3 - C Programming
15%	Lab4 - Crawler
15%	Lab5 - Indexer
15%	Lab6 - Query Engine
10%	Lab7 - Socket Programming

We will provide source code solutions to all labs. The TAs/section leaders will grade your solutions and write up a grade sheet on the correctness, simplicity, and clarity of your code. The instructor will review the TA's grading and grade sheets. Your grade and grade sheet and our source code solution will put in your svn repository (which is an open-source revision control system) one week after the lab assignment is turned in. If you have questions about the grade or grade sheet please talk to the TA/section leader first. If you are still have concerns see the instructor. **Please do not distribute the source code solutions we provide you with (see honor code).**

25% - Team project

The project is made up of a small team (three or four people) and requires strong collaboration and a problem solving mindset to get the job done. The instructor will put the teams together with each member being responsible to deliver against a part of the overall system design, implementation, testing and integration. The goals of this activity are to help you develop the confidence, skills, and habits necessary to write large computer programs while part of a multi-person team. You will become conversant in software engineering paradigms, and be exposed to various public-domain and open source tools that make the software development process easier. In addition, you will develop vital skills in self-directed learning, problem solving, and communication. The project will have a design and code review as well as the demo. A project report that captures the design and implementation will be submitted as part of the assessment.

Policy for late assignments

Unless prior arrangements have been made, or in case of medical or family emergencies, or specific disabilities, all assignments are subject to the following policy regarding late submissions:

Programming assignments must be submitted according to the schedule; if late, the following grade penalties are assessed:

Late < 8 hours
  Ten percent (10%) will be deducted from the final score.
Late ≥ 8 hours and < 24 hours
  Twenty percent (20%) will be deducted from the final score.
Late ≥ 24 hours and < 48 hours
  Forty percent (40%) will be deducted from the final score.
Late 48 or more hours
  Speak to the lecturer.

Students are given two free passes for 48 hour extension with no penalty. Keep these for the more challenging programming assignments. But note that by taking these free passes you are running behind on a very tight schedule.

Caveat: Students get stressed with the demanding programming assignments that is why I provide the two free passes. Key thing is to come talk to me if you feel you are not making progress on assignments. The most important thing for you to do is to to finish the programming assignment even if late.

Lab access

In order to obtain access to Sudikoff after hours, and to get into Sudikoff's Lab 001, you will need to have your Dartmouth ID card activated for the appropriate access. To do this, stop by the front desk at 101 Sudikoff on a weekday between 8:30am-12:00pm, or 1:00-4:00pm, and bring your Dartmouth ID card. and say you are taking CS50, and require access to Lab 001. You will have to fill out and sign a form stating that you understand the various policies about access to the labs in Sudikoff.

Keep in mind that it may take 24 hours for access to be activated, so please plan ahead!

You will also require a computer account on the CS Linux machines. Contact the TA Fanglin Chen (Fanghao.Chen.GR@dartmouth.edu) informing him of your preferred (new) account name.

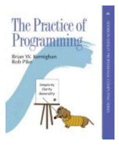
Please note that the exterior doors of Sudikoff are automatically locked after 6:00pm weekdays, and also every weekend and holiday. In addition, the laboratory doors are locked at all times. You will need your access card to pass through locked doors.

Books

There is no one book that caters for the material in the course. However, we do recommend a book on the C language - this is a very good book:

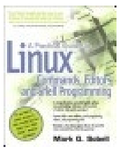
A First Book of ANSI C, Fourth Edition by Gary J. Bronson

We do not recommend you buy any other books for the course. But If I were to recommend a book on design and programming for Unix and C it would be this **classic** text:



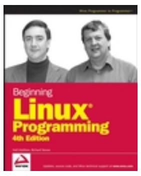
The Practice of Programming (Addison-Wesley Professional Computing Series) by Brian W. Kernighan, Rob Pike

If I where to recommend a hands on book on Linux and shell programming it would be this one (lots of good stuff in this):



A Practical Guide to Linux Commands, Editors, and Shell Programming by Mark G. Sobell

Another really good book covering, debugging, processes, threads, and socket programming in clear and easy manner to grasp:



Beginning Linux Programming, 4th Edition by Neil Matthew, Richard Stones

Just to make things clear. Buy the ANSI C book for the course. But if you are interested in hacking C, shell scripts, Linux you might like the other books for your library. They are all really nice books - I love them.

Policy on Joint Work and the Dartmouth Honor Code

The assignments and project are all about writing great code - shell scripts and C. Here is the policy for joint work in CS50:

First, you may discuss and help each other (e.g., help in debugging, sharing knowledge, giving moral support, getting coffee, etc.) - I promote that as the type of team spirit and joint problem solving skills that is the essence of the course and necessary to do a great project. **However, you cannot work jointly on coding up (i.e., writing) your programming assignments**. You can talk, discuss solutions, even show snippets of code on the white board (not the computer) to solve a problem but you cannot jointly work on the code development and writing. Submitted code for the labs has to be yours and yours alone.

The project phase is different. You can work jointly on writing code. But you cannot take code from anywhere (e.g., the web or any other source). It has to be the joint product of the team. One caveat: no sharing of code between teams. As above, teams can discuss code, show each other snippets on the white board, but not share source code. The project phase of the course is a friendly competition so there isn't a lot of incentive to share code.

We hate to state this since you all know it but it is necessary to be explicit here (citation - culled and extended from CS8 Policy on joint work):

You would be amazed at how easy it is to tell when people work together on problem sets, particularly coding exercises. Think about the simple shell commands we run against your source code from labs and projects to compare your lab assignments and projects against every other assignment and project ever submitted since this revision of course started in 2008 - it takes less that a millisecond to run these checks - no effort on our behalf. Similarly, we know how to use google too. **You should not under any circumstance look at or use code from students that have previously taken CS50**. The message is simple - please don't make life unpleasant for all of us by breaking these rules. The penalties for cheating at Dartmouth are severe, starting with suspension and including expulsion. If you are unsure about anything, please ask.

We take the Honor Code very seriously, so please, if you are unclear on any matter regarding this policy, do not hesitate to see me in office hours, and we will be more than happy to answer your questions.