

CS 5 Fall 2007

Course Information

Instructor

Tom Cormen

Office: 204 Sudikoff Laboratory (Computer Science) or 208 Baker Library (Writing Program)

Email: tom.cormen@dartmouth.edu or thc@cs.dartmouth.edu

Telephone: 646-2417 (Computer Science) or 646-9749 (Writing Program)

Office hours (September 26 through December 4), in 204 Sudikoff.

- Monday, 11:00 am-12:00 noon
- Tuesday, 4:15-6:15 pm
- Thursday, 10:00 am-11:30 am

and by appointment.

Additionally, any time my door is open and I'm not talking to anyone, feel free to knock and come in. I try very hard to be available to students. I am often around on campus during evenings and weekends; call or Blitz to verify that I'm around.

Teaching Assistants

- Jan Sriram
Office: 220 Sudikoff Laboratory
Office hours: Tuesday 2:00-4:00 pm
Email: janani.sriram@dartmouth.edu
Telephone: 646-8752
- Nick Foti
Office: 157 Sudikoff Laboratory
Office hours: Tuesday 10:30 am-12:30 pm Email: nick.foti@dartmouth.edu
Telephone: 646-8743

Section Leaders

- Deliana Dalton '10
- Tilman Dette '10
- David Kopec '09
- Lisa Marchessault '10
- Kristin Ricci '08
- Jarman Rogers '09
- Gemma Ross '08

- [Loren Sands-Ramshaw '10](#)
- Justin Sanford '08

Lectures

Lectures are in Filene Auditorium in Moore Hall in the 2 hour: MWF, 1:45-2:50, x-hour Thursday 1:00-1:50.

x-hours

You must keep x-hours available for this course. I expect to use the x-hour *every* week during this term (except for Thanksgiving, of course). We'll use x-hours for lectures (and you are as responsible for material covered in x-hours as you are for material covered during regular lecture hours), review sessions, and in-class quizzes. The [syllabus](#) page will tell you what we're using the x-hour for in each week.

Questions

I like it when you ask questions in lecture. I understand that it can be intimidating to raise your hand for a question in a fairly large lecture, but please do not hesitate to ask questions. Don't worry about what anyone thinks of you for asking questions. I'll think more of you if you understand the material, and if you need to ask questions, that's fine. And don't worry about what the other students might think. It is likely that if you have a question, then at least one other student—and possibly many more—have the same question. You're doing the other students a favor by asking!

If you still are reluctant to ask a question in lecture, even after all the encouragement in the last paragraph, feel free to write it down and ask me, a TA, or a section leader after class, by blitz, in office hours, or in lab hours. We want to help you!!

Occasionally, I might elect to answer a question later or after lecture. That will usually be because the answer to the question is tangential to the material we're covering, or simply because I am pressed for time. But please feel free to ask; I'll let you know nicely if I cannot answer the question then and there.

Recitation Sections

You must attend a one-hour recitation section each week. Attendance is mandatory; section leaders will actually take attendance! The recitation section schedule will be determined by the end of the day on Friday, September 28, and your section assignment will be on the Recitation Sections page by then. This page will be accessible only through the [Blackboard site](#). You are responsible for checking this page and attending your recitation section starting on October 3, 4, or 5.

You need to download the [Section Time Availability Form](#), fill it out, and email to me by 3:30 pm on Friday, September 28. If you say you are available during a one-hour slot, I expect you to honor this commitment. Please take this expectation into account as you make other commitments on and off campus this term.

We recognize that occasionally (which means "not on a regular basis"), you might not be able to attend the

recitation section to which you have been assigned. In that event, please find another section to attend that week. Make sure that your regular section leader knows that you'll be attending another section, and make sure you get permission *in advance* from the section leader whose section you'll be visiting.

Labs

Room 005 in Sudikoff has a number of iMacs that are available for your use. They run OS X version 10.4.10 or later.

005 Sudikoff is locked at all times, but CS 5 students will be able to get in by doing the following:

1. Go to 101 Sudikoff during a weekday between 8:30 am and 12:00 noon or between 1:00 and 4:00 pm.
2. Tell the person there that you're a CS 5 student who will need access to the Mac lab.
3. Bring your Dartmouth ID card with you.
4. **The next day**, your Dartmouth ID will get you access to the labs. *Because it takes a day to activate your access, make sure to plan ahead.* If you want access for over the weekend, go to 101 Sudikoff on Thursday.
5. Once your access is activated, merely holding your Dartmouth ID up to the black-box card reader outside 005 Sudikoff will unlock the door for you.

The entrances to Sudikoff are locked after 6:00 pm on weekdays and all weekends and holidays. Therefore, if you think you'll need to enter the building after hours—even if you won't be going into 005—you'll want to perform the above procedure so that you can get in. If your recitation section is on a Wednesday evening and is located in Sudikoff, you'll certainly want to make sure you have access.

Note that there are several restrictions that come with access to 005 Sudikoff. Among them are that you are not to lend your Dartmouth ID to anyone else in order to allow them into the building or labs. Also, *you must not bring food into the lab*. There are tables in the hallway outside the lab for you to eat at. If we find that people are bringing food into the lab, we will close down the lab. We really will. We've done it before and although we don't want to do it again, we will if we have to. Sorry to be so adamant about no food in the lab, but food in the lab can make the machines unpleasant or even impossible to use.

There will be course staff members in 005 Sudikoff to help you during the following hours:

- Sunday, 7:30-10:30 pm
- Monday, 7:30-10:30 pm
- Tuesday, 7:30-10:30 pm

Note: On Sunday evenings, we share the lab with CS 22, 3D Digital Modeling. Some CS 22 students might be in the lab at other times, too.

You may find it useful to do some of the assignments right in the lab. Many of the problems that will stump you and waste your time if you are working alone can be cleared up in moments by the course staff. Then again, you might not find it useful to work in the lab, depending on how many of your fellow students are competing with you for machines.

In addition, whenever a short assignment is due on Friday, we will have a course staff member in 005

Sudikoff during the hours

- Thursday, 8:00-10:30 pm

Later in this page, we refer to some homework assignments as "lab assignments." You may work on lab assignments wherever you like. You need not work on them in 005 Sudikoff.

Text

The course text is John Lewis and William Loftus, *Java Software Solutions: Foundations of Program Design, Fifth Edition*. This book contains much of the basic information about Java. We will often use class time to elaborate, clarify, and expand on topics that are discussed in the text.

Please note that we are using the **fifth edition** of the Lewis and Loftus text. If someone offers to sell you a Lewis and Loftus book that is a previous edition—even at a deep discount—you should decline their kind offer.

Both Wheelock Books and the Dartmouth Bookstore carry the book. I have put one copy on 4-hour reserve in Baker.

I have put all examples from the text on [our own web site](#) (and there's a link on the top-level page of the CS 5 web site).

The [syllabus](#) page will tell you the reading for each lecture. Some of the assignments will *remind* you what to read, but the ultimate authority on what to read for each lecture is the syllabus page.

The Lewis and Loftus text is recommended, rather than required. That's because in past course evaluations, several students commented that they found the lecture notes I publish on-line to be wholly sufficient and that they wished they had not purchased the textbook. The text contains material that we do not cover in lecture, but that you should know. That's why I recommend purchasing the textbook.

Software

We will be using Eclipse 3.3 for Java. This software is a free Integrated Development Environment (IDE) that runs on both Windows and Mac OS X.

Eclipse 3.3 uses Mac OS X version 10.4 and Windows XP as its "reference systems." That means that these are the operating systems that new versions are tested and validated on. In fact, Eclipse 3.3 requires Mac OS X version 10.4 or later; if your Mac runs version 10.3 or earlier, you will have to either upgrade to 10.4 or use a different machine. Note that the iMacs in 005 Sudikoff all run Mac OS X version 10.4.

Unfortunately for Windows users (and, yes, we know that the overwhelming majority of you run Windows), the Mac will be the "primary" machine supported. The CS undergraduate lab for this course contains only iMacs. The demonstration machine that I use is a Mac. Some of the course staff have experience with Windows, however, so you may find it a good idea to ask them your Windows-specific questions. Because I don't run Windows, I will be unable to help you with Windows-specific problems. Indeed, one great asset of our TAs and some of our section leaders is that they can help you with Windows issues.

Note that you can always Blitz your program to yourself. So even if you normally run Windows but you want to work on an iMac in the lab, you'll be able to work on your own code.

The Web

All documents, class examples, lab assignments, short assignments, and sample solutions related to the course will be on the web. You can get at them through the course [Blackboard site](#), or you can dial direct and go to <http://www.cs.dartmouth.edu/~cs5/>. Presumably you know how to find the web pages, since you're reading this one. The first short assignment tells you how to find the CS 5 information on the web.

Blitzmail

I will assume that you know how to use electronic mail, such as Blitzmail. I will send various kinds of information by way of electronic mail addresses, and I will respond to questions from you sent by Blitz. Please check your Blitzmail frequently, and before working on an assignment, check whether I have made any changes.

I will assume that when I send out a Blitz to the class, you will read it.

Help

There are many ways for you to get help. You can visit me during office hours, you can go to 005 Sudikoff when course staff is there, or you can make an appointment with me, your section leader, or a TA. You should also feel free to ask for help by Blitzmail. You can Blitz a specific person, including me. For a faster response, you can get an answer from the first member of the course staff (again including me) to read your message by Blitzing to cs5help@cs.dartmouth.edu; this account is a mailing list consisting of me, the TAs, and all section leaders. (Note that you have to put cs. after the @ in the address.) We will try to respond as soon as possible.

If you visit me or any of the course staff for help with a program that you are writing, make sure that you can get to an electronic copy of your program. (Either Blitz it to whom you're visiting or to yourself, or turn on file sharing, or store it in Vault or some other publicly accessible file server.) That way, we can try to compile and run it. It's nice to have a listing, but we cannot compile and run a listing.

When you Blitz a program to one of us, you just need to add your program as an enclosure to your Blitz, along with additional information as to what problems you have observed. Make sure you send your program as an enclosure; *do not copy and paste your program into the message*.

The Tutor Clearinghouse is another source of help. They will have private, one-on-one tutors available for this class. The tutors are recruited on the basis that they have done well in the subject, and they are trained by the Academic Skills Center. If you are on financial aid, the College will pay for three hours a week of tutoring. To get a tutor, go to 301 Collis and fill out an application.

Material That Will Be Turned In

On many Mondays, Wednesdays, and Fridays, one of two types of assignment will be due. The [Homework Requirements](#) page explains what quality is expected of homework that you submit.

Short assignments

These will be relatively brief exercises that will be turned in at the very next class. They usually will consist of one or two short programs that help you understand the concept being discussed. They will be turned in via the CS 5 HW HAND IN boxes in the lobby outside Filene. You will hand in only hardcopy for short assignments. The short assignments will be graded and returned to the CS 5 HW RETURNED boxes. Most of the short assignments will be earlier in the course.

Lab assignments

During the term, there will be five longer lab assignments. These will be handed out usually on Wednesdays and will be due usually one week later.

Note that although we refer to the longer assignments as "lab assignments," you may work on them wherever you like. You need not work on them in 005 Sudikoff.

Because CS 5 contains substantial lab assignments, it satisfies the laboratory distributive requirement (i.e., it's a TLA).

Homework Policy

Your assignments will be graded by your section leader.

For each assignment, you must place your hardcopy materials in a large envelope (8.5 x 11 inches), with your name and your section leader's name appearing conspicuously on the outside of the envelope. (You can omit your section leader's name for [Short Assignment 1](#), since you won't know who your section leader is at the time you hand it in.) Your name should appear as well on each program listing, output, and any other written material that you submit. If you cannot get a large envelope, see Professor Cormen or a TA, and we will get you one. But do so *in advance*. We do *not* bring envelopes to class.

The **short assignments** are to be turned in at the next class time (see the lateness policy below) with whatever materials are needed to show that you can, and have, done the assignments. This will be a bit more informal since the exercises will be much shorter. Normally, you will turn in hard copy of program listings and runs, or a separate sheet of paper on which you have given written responses to questions posed. All such materials should be placed in a large envelope and turned in at class time on the due date. You must turn them in on time so that your section leader can check them and arrange for help in cases where you need it. If you do not get full credit on a short assignment, you may resubmit it one time, provided that you resubmit an assignment within two days of when you got it back (or if you get it back on a Friday, you must resubmit by the next Monday.) You may resubmit up to six short assignments of your choosing.

When you resubmit a short assignment, we ask that you do a couple of things as a courtesy to your grader. First, please make it clear that it is a resubmission. Second, please include your original submission along with the resubmission, so that your grader can see your progress.

We will ask you to be more formal when submitting your work on the **lab assignments**. When you turn in your lab assignments, you should include hardcopy listings, output, and any other required written materials.

All such materials should be placed in a large envelope and turned in at class time on the due date. Place your lab assignments, packaged as noted, in the CS 5 HW HAND IN boxes across the lobby from Moore B03.

For the lab assignments, you will also turn in the programs electronically. Each lab assignment will tell you how to submit your programs electronically. Your section leader will run your programs, if necessary, as she or he reads your printed listings and runs.

Please note that you will submit electronically *only* the lab assignments. You will hand in only hardcopy for your short assignments.

Assignment Grading

Short assignments will be graded on a scale of 0-2. Your section leader will allow you to resubmit if you get a 1, but not if you get a 0. In the past, we allowed resubmits for 0s. Several students abused the system, repeatedly submitting nothing of substance and then resubmitting later. By allowing resubmission only if you get a 1, we encourage you to put in an honest effort on the short assignments.

Short assignments are due at the start of class on the day they are due. (They will be collected a few minutes after the class starts.) We will not accept short assignments handed in late. If you do not hand in a short assignment on time, you will get a score of 0 for that assignment and you may not resubmit. Solutions to short assignments will be posted on the web right after the class in which they are due. *Yes, you may refer to the posted solution when preparing a resubmission.*

Each of the five lab assignments is scored on a basis of 40 points. The electronic version and the envelope with your hardcopy will be due at the start of class on the due date. Solutions will be posted on the web at 5:00 pm the next day.

By the start of class, we mean 1:45 pm *sharp*. **If your assignment is submitted after 1:45 pm (i.e., 1:46 pm or later) on the due date but by 5:00 pm the next day, there is an automatic 16-point deduction.** That is, we will determine what your score would have been had you submitted your assignment on time and then subtract 16. (If the resulting score turns out to be negative, we will just score it as a 0.) If your assignment is not sent by 5:00 pm the next day, you will get a 0.

It might happen that you submit your program electronically on time but cannot get the printout ready by the start of class. If this happens and you hand in the printout by 5:00 pm on the day that the assignment is due, we will deduct only 1 point. If you hand it in by 5:00 pm the next day (still having submitted the electronic version on time), the deduction is 2 points. If you don't hand in the printout at all, the deduction is 5 points. To hand in the printout of your lab assignment late, you must submit it to me, to a TA, or to your section leader. If you submit the electronic version late, we will not take a further deduction for a late or missing printout.

There are two ways that you can get an exemption from this lateness policy. First, if something happens beyond your control that prevents you from doing your assignment on time and you have a note from Dick's House or a dean, then we can excuse a late assignment. Second, you may be able to arrange with Professor Cormen (and only Professor Cormen) **in advance** to submit your assignment late. You will need a Darned Good Reason to do so. Note that you need to ask **in advance**. The day the assignment is due does not count

as "in advance."

Yes, this lateness policy is harsh. Why? Because in the past, those who have fallen behind have had a devil of a time catching up. So we are trying to prevent you falling behind. Yes, 40% off for a lab assignment handed in at 1:46 pm is inflexible. In the past, students have complained that they could have handed in something substandard on time and gotten more points than if they had handed in something really good a little late. Unfortunately, the Real World works this way as well. Imagine showing the World's Best Software...a week after the trade show. **It is up to you to plan your time carefully and get your work in on time!**

Short assignments and lab assignments will be returned after grading to the boxes labeled CS 5 HW RETURNED in the lobby outside Filene.

Quizzes

We will have three in-class quizzes during x-hours on the following dates:

- Thursday, October 11
- Thursday, October 25
- Thursday, November 29

Exams

- Exam 1: Thursday, October 18, 6:00-9:00 pm, in Filene.
- Exam 2: Thursday, November 8, 6:00-9:00 pm, in Filene.

The exams are designed to take less than the three hours allocated, but I don't want you to feel time pressure.

- Final examination: Friday, December 7, 3:00-6:00 pm, location TBA.

Prior to each exam, I will post a web page to help you review. There will also be review sessions during the x-hours the days of the first two exams. We will hold a review session during reading period, on Wednesday, December 5, 10:00 am to 12:00 noon, in Filene.

Grading

- Short assignments: 15%
- Lab assignments: 45%
- Quizzes: 15%
- Exams: 25%

Your final grade will be based on the percentage of all available points that you receive. To give you an idea of how percentages might translate into letter grades, here is the correspondence from the Fall 2006 term. **I do not claim that the grade cutoffs for this term will be the same. These cutoffs are merely to give you an idea of how I have graded in the past.** (Really. Do not use the cutoffs below as *this term's* cutoffs.)

- A: 95-100
- A-: 91-95
- B+: 88-91
- B: 84.5-88
- B-: 81-84.5
- C+: 77.5-81
- C: 73-77.5
- C-: 70-73
- D: 55-70
- E: 0-55

Extra Credit

Some of the assignments may suggest extra credit work. Extra credit in this course will be tallied separately from regular scores. If you end up on a borderline between two grades at the end of the course (or are being considered for a citation), extra credit will count in your favor. Failure to do extra credit will never be counted against you, however. You should do extra credit work if you find it interesting and think that it might teach you something. It never pays to skimp on the regular assignments or reading in order to do extra credit problems.

One corollary of the way in which we count extra credit is that if you get a 30 on a lab assignment and also 5 points of extra credit, that is *not* the same as getting a 35 with no extra credit. The latter is better.

Disabilities

I encourage students with disabilities, including "invisible" disabilities such as chronic diseases and learning disabilities, to discuss with me after class or during my office hours appropriate accommodations that might be helpful to you.

Students with disabilities enrolled in this course and who may need disability-related classroom accommodations are encouraged to make an appointment to see me before the end of the second week of the term. All discussions will remain confidential, although the Student Accessibility Services office may be consulted to discuss appropriate implementation of any accommodation requested.

Honor Principle

On exams, all work must be your own. You may work on short assignments individually or in groups. **Programs that you turn in, however, should be created, typed, documented, and output generated, yourself.** Similarly for the lab assignments, you may consult freely with instructors and classmates during the phase of designing solutions, but you should then work individually when creating your programs — typing, documenting, and generating output. During the debugging stage you may discuss your problems with others in the class, but you should not copy code to "fix" bugs. **To do otherwise is a violation of the Academic Honor Principle. If you work with a classmate on any assignment, you should tell us who you worked with in a comment at the beginning of your program.**

You should attribute the proper source in any code that you submit that you did not write yourself. This includes code that you take from outside references—for example a book other than the course text. And it even includes code that you take from class examples, the course text, or the assignments. (I agree that may be tedious to attribute the source in code that we have given you, but we want you to be in the habit of attributing your sources.)

If you resubmit a short assignment and use code from the published solution, you *should* attribute that. Note also that proper respect for copyright laws as it applies to printed and software products is part of the *Computing Code of Ethics*.

Whenever we ask you to turn in sample runs of your program, the runs you turn in must be the result of actually running your program. **It is a violation of the Academic Honor Principle to falsely represent a printout as being output from your program.** If you change your program, make sure to generate output from **the version of the program that you hand in.** It's amazing how a seemingly minor change to the code can cause a big change to the output of a program. Also, make sure that when you are running a program, that it is *your* program; it is easier than you might think on a public Mac to run a program that someone else had left on the machine.

In the past, we have had a few incidents in which students turned in output that did not come from the program handed in. In each case, it turned out that the student had made a foolish mistake (in not rerunning the program or handing in an old version of the program or the output) and had not intended to misrepresent the work. Yet it caused many an uncomfortable moment for the student and also for the student's section leader and for me as well. So please—pretty please with sugar on top—endeavor to verify that you're handing in output that comes from the very program you're handing in.

It is not easy to come up with good homework problems that help you learn a concept, are interesting, and require an appropriate amount of work. Over the years we have developed and refined a number of homework problems, and I plan to reuse some of them for this class. You should **not** look at any solutions to homeworks assigned in previous terms, including sample solutions, or solutions written by other students.

We have had some uncomfortable situations occur in the past, and I want to make it clear what the policy is. Two students, Alice and Ralph, discuss an assignment and design their code together. That is fine. But then they decide that since their programs would be so similar, they might as well have Alice type in the code, have Ralph make his own copy of the file containing the code, and then have Ralph make his own minor changes. **This is a violation of the Academic Honor Principle.** Although you may discuss and design with others, the code you hand in must be entirely your own.

Here's another situation that occurred. Trixie and Ed start working independently on a program. Trixie finishes and has a working version. Ed has trouble with his. Trixie helps Ed debug. That is fine. But then Trixie realizes that Ed has a section of code that is all wrong and the program she wrote has just the right code for that section. She shows Ed her program. Or worse, she gives Ed an electronic copy of her program so that he can just paste in the correct code. **Either action is a violation of the Academic Honor Principle.**

I realize that it can be hard to decide when you might be violating the Academic Honor Principle when we let you collaborate to a limited extent. Here is a good rule of thumb. If you are talking in normal English (or Chinese or German or some other natural language) you are probably OK. If you find yourself talking in Java code, you have crossed the line. So saying, "Your program runs forever because you have the wrong

condition in the while loop" is OK. But saying, "Change the `while (x == 0)` to `while (x != 0)`" is not.

If you have any question about whether what you're doing is within the Academic Honor Principle, do not hesitate to check with me. If it's late and you can't find me, you're better off erring on the side of caution.

Most violations of the Academic Honor Principle come down to **failure to cite work that is not yours**. If you copy any portion of your program from your friend Elvira and represent it as your work, then you either intended to deceive or were careless about citing. Either case is a violation of the Academic Honor Principle. If you copy your entire program from Elvira but include on the printout, "This code was copied in its entirety from Elvira," then you cited properly, though you didn't actually do the work. In this latter case, I would not report a violation of the Academic Honor Principle, though your grade on the assignment would look a lot like a Krispy Kreme doughnut. (I'm talking about the original glazed doughnuts, not the creme-filled ones. In other words, your grade would be 0.) But that would be *far* preferable to a COS hearing.

The same goes for code that you find in some other book or on the Internet. You are in violation of the Academic Honor Principle if you fail to attribute your sources.

You don't need to cite if you wrote the code yourself. You don't need to cite just because you're using a construct you saw elsewhere. For example, you need not cite for using `System.out.println(something)`, even though it was in the class examples. That would be like citing "printing press" in an essay! Nor do you have to cite just because you use a for-loop, even though you saw a for-loop in a class example. It's when you lift several lines of code from elsewhere that you need to cite.

To cite, include in a comment—near the top of your file is fine—stating where you got the code from:

```
// Based on Fact6.java in the September 26 lecture.
```

or

```
// Based on the Addition.java program on page 67 of Lewis and Loftus.
```

Please do not cheat. Cheaters—whether or not they are caught—bring dishonor upon themselves and upon everyone else at Dartmouth. To do that, for just a few lousy points in a course, is [insert your favorite strong adjective meaning "stupid" here]. Furthermore, if you cannot solve the short assignments and lab assignments on your own, then you will be nailed on the quizzes and exams.

I have served on the COS, and I have seen what happens to students who are caught cheating: they are sent on the "Parkhurst FSP." I have brought several Academic Honor Principle cases to the COS, and I deeply dislike having to do so. Please don't make me.

Special note for when you work on a computer that anyone else might use

If you are working on a computer that is not yours—especially a Mac in 005 Sudikoff—or that someone else in the course might use, you should be very careful to remove your code from the computer when you are all done. You should probably Blitz your code to yourself before you remove the code.

Why do we tell you to do this? Because if you leave your code on a computer, and someone else can see it,

then they can copy it and hand it in. If that happens, then we have a bad situation involving you (the copy-ee) and the other person (the copy-er), and it's difficult—if not impossible—to tell who was the copy-ee and who was the copy-er. By removing your code from the computer when you're done, you can avoid getting yourself into that situation.

To remove your code, you'll want to delete it from the Eclipse workspace. And you'll also want to move any other copies on the computer to the Trash (or the Recycling Bin) and empty it.

Advice

Read the material I ask you to read, and follow the instructions. Many students have wasted a lot of their valuable time, and their grades have suffered, *simply because they did not follow directions*. **Start all assignments early.** With few exceptions, at the time you receive an assignment, you'll know everything you need to do it. There is no reason to procrastinate.

At the same time, do not be afraid to get help. The purpose of this course is not to waste your time. If you are spinning and not making progress on a problem, please see me, a TA, or a section leader. We can point you in the right direction without giving away the store.

Three final pieces of advice

1. Don't fall behind in this course.
2. *Don't fall behind in this course.*
3. ***DON'T FALL BEHIND IN THIS COURSE.***

The material in CS 5 builds on itself, and the pace is fast. As a result, it's easy to fall behind in this course, and if you do it's very difficult to recover.

[Thomas H. Cormen <thc@cs.dartmouth.edu>](mailto:thc@cs.dartmouth.edu)

Last modified: Tue Oct 2 08:51:25 2007

[CS 5 Fall 2007](#)

Syllabus

Schedule subject to change with minimal notice.

All readings are from Lewis and Loftus.

[\[Week 1\]](#) [\[Week 2\]](#) [\[Week 3\]](#) [\[Week 4\]](#) [\[Week 5\]](#) [\[Week 6\]](#) [\[Week 7\]](#) [\[Week 8\]](#) [\[Week 9\]](#) [\[Week 10\]](#)
[\[Week 11\]](#)

Week 1

[Wednesday, September 26:](#)

Why study Computer Science? Administrivia. What makes a good program?

[Thursday, September 27](#) (x-hour):

What makes a good program, continued.

Reading: Chapter 1.

[Friday, September 28:](#)

Simple types and values, variables.

Reading: 2.1-2.3.

Week 2

[Monday, October 1:](#)

Simple types: operators, Scanner class, type conversions, and other basic stuff. References and objects. Using builtin classes: DecimalFormat.

Reading: 2.4-2.6, 3.1, 3.6.

[Wednesday, October 3:](#)

Using builtin classes: String, Math, Random. More on operators. Enumerated types.

Reading: 2.5, 3.2-3.5, 3.7.

[Thursday, October 4](#) (x-hour):

Flow of control: if-else statements, loops.

Reading: 5.2-5.5, 5.7-5.8.

[Friday, October 5:](#)

Boolean expressions. Nested loops. Introduction to writing classes.

Reading: 5.1, 4.1-4.5.

Week 3

[Monday, October 8:](#)

More on writing classes. Return values from methods.

Reading: 6.7-6.8.

Wednesday, October 10:

Problem decomposition.

Reading: 6.1-6.2, 6.4.

Lab #1 out.

Thursday, October 11 (x-hour):

In-class quiz.

Friday, October 12:

Method call mechanisms. Garbage collection.

Reading: 5.0.

Week 4Monday, October 15:

Scope of names. Static methods and static variables. Wrapper classes and autoboxing.

Reading: 3.8, 6.3.

Wednesday, October 17:

Debugging and testing. Graphical applets.

Reading: 2.7-2.9, 6.9.

Lab #1 due.

Thursday, October 18 (x-hour):

Open review session.

Exam at 6:00 PM in Filene.

Friday, October 19:

No class.

Week 5Monday, October 22:

Objects and applets. Interfaces.

Reading: 6.5, 7.9.

Wednesday, October 24:

Events and listeners. Manual animations. Animations with timers.

Reading: 8.8.

Thursday, October 25 (x-hour):

Arrays. Arrays of objects. Multidimensional arrays.

Reading: 7.1-7.6.

Friday, October 26:

Arrays, continued. Multidimensional arrays. Objects containing arrays. ArrayList class. Generic types.

Reading: 7.7-7.8.

Week 6Monday, October 29:

Graphics using arrays. Inheritance.

Reading: 8.1-8.3, 8.5.

Wednesday, October 31:

Inheritance, polymorphism, dynamic binding, and graphics.

Reading: 8.4, 8.7, 9.1-9.3, 9.7.

Lab #2 out.

Thursday, November 1 (x-hour):

In-class quiz.

Friday, November 2:

Object class. GUI components.

Reading: 3.9-3.10, 4.7-4.9.

Week 7Monday, November 5:

More on GUI components. Recursion.

Reading: 5.10-5.11, 11.1-11.3.

Wednesday, November 7:

More recursion.

Reading: 11.4.

Lab #2 due. Lab #3 out.

Thursday, November 8 (x-hour):

Open review session.

Exam at 6:00 PM in Filene.

Friday, November 9:

Swing: containers, components, and layout managers.

Reading: 6.11-6.13, 8.6.

Week 8Monday, November 12:

Interval bisection. Case study: Amortizing a loan.

Wednesday, November 14:

Searching: linear search, binary search.

Reading: 9.5.

Lab #3 due. Lab #4 out.

Thursday, November 15 (x-hour):

Searching: binary search. Orders of growth.

Friday, November 16:

Sorting: selection sort, insertion sort.

Reading: 9.4.

Week 9Monday, November 19:

Sorting: merge sort, quicksort.

Wednesday, November 21 and Friday, November 23:

No class. Thanksgiving break.

Week 10

Monday, November 26:

Data structures: heaps and priority queues. Heapsort.
Lab #4 due. Lab #5 out.

Wednesday, November 28:

More heapsort. Data structures: doubly linked lists.

Thursday, November 29:

In-class quiz.

Friday, November 30:

Data structures: doubly linked lists and singly linked lists.
Reading: 12.1-12.2 (skim; we'll use a different structure).

Week 11

Monday, December 3:

Singly linked lists and ADTs.

Tuesday, December 4:

Lab #5 due at 5:50 pm.

Wednesday, December 5:

Exam review session, 10:00 am-12:00 noon in Filene

Friday, December 7:

Final exam 3:00-6:00 PM in Filene.

[Thomas H. Cormen <thc@cs.dartmouth.edu>](mailto:thc@cs.dartmouth.edu)

Last modified: Sat Dec 1 13:57:35 2007