C SC 397a: Advanced C++
The University of Arizona
Spring 2010

Instructor

Name:	William H. Mitchell
E-mail:	`whm` (on the CS machines)
IM:	`x77686d` on AIM, Skype, and Yahoo
Office:	Gould-Simpson 842
Office hours:	*See Office Hours section below*
Phone:	577-6431 (mobile; 8am to 9pm, seven days a week; no texts!)

Grader

Name:	Rebecca Bailey
E-mail:	`rlbailey` (on the CS machines)

Prerequisites

C SC 335 and C SC 352

Website

The class website is `www.cs.arizona.edu/classes/cs397a/spring10`.

Textbooks

It is intended that the lectures, handouts, notes to the mailing list and materials on the class website will provide all the information needed to successfully complete this course. There is no required text for this class.

However, if you'd like a well-written and comprehensive C++ book, my top recommendation is *C++ Primer, 4th edition* by Lippman, Lajoie, and Moo. Bruce Eckel's *Thinking in C++, 2nd edition* is available for free on the web: `http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html`. It is very good. Tim Budd's *C++ for Java Programmers* is focused on programmers who know Java, and does a good job with that approach. I have a copy of it in my office, if you'd like to get a peek at it. *C++ for Java Programmers* by Mark Weiss looks promising from a distance but I don't have a copy of it yet. I think *The C++ Programming Language* by Bjarne Stroustrup, the creator of C++, is a great book once you have a handle on the language but I don't think of it as a great book for learning C++.

The C++ language has been very stable since 1998; any C++ book published after that stands a chance of being a good resource for this class. The library has dozens of books on C++.

Course Description

This course covers the fundamentals of the C++ programming language, building on prerequisite knowledge of C and Java. Major topics include class definition, object creation and interaction, aggregation of objects, single and multiple inheritance, operator overloading, I/O streams, exception handling, and templates. The Standard Template Library is introduced with coverage of the principles of containers, iterators, algorithms, and function objects.

An underlying theme of the course will be the tensions that arise from C++ simultaneously providing support for object-oriented programming, type extensibility, high-level language facilities, very strong compile-time type-checking, and efficient execution.

Note: The course title in the catalog, *Advanced C++*, is perhaps misleading since no prior knowledge of C++ is assumed. Perhaps a better title would be *Fundamentals of C++*.

Topic Sequence

This is the anticipated set and sequence of topics:

Introduction and Overview
Class and Object Basics in C++
Fine Points on Classes and Objects
Miscellany—References, `const`, Friends, and more
Aggregations of Objects
Operator Overloading
IO Streams
Inheritance
Multiple Inheritance
Templates
Exceptions
Run-Time Type Information
Introduction to the C++ Standard Library

Grading Structure

This is the composition of the final grade:

Assignments	90%
Quizzes	10%

Note that there are no examinations and no attendance/participation component in the grade.

This course has Pass/Fail grading; the only possible final grades are "P", "F", and "I". **An overall average of 75% is required to earn a "P".**

It is my goal that you will need to spend on average no more than four hours per week, counting lectures, to maintain an average over 90% in this course. If you find that you're averaging more than *three* hours per week on this course, let's talk about it.

You are strongly encouraged to contest any assigned score that you feel is not equitable.

Assignments

Homework assignments will largely consist of programming problems but other types of problems, such as short answer questions, may appear as well. About ten assignments are anticipated. There will be a total of 900 points worth of assignments. Thus, a ten point homework problem corresponds to one point on your final average.

For programming problems, great value is placed on the ability to deliver code whose output exactly matches the specifications. Failing to achieve that will typically result in large point deductions, sometimes the full value of the problem. When reference versions of assigned programs are made available, use `diff` to ensure that your output matches that of the reference versions for all examples of input that are provided.

Submitted solutions for C++ programs will be compiled with `g++` on `lectura`, and run on `lectura`. You're of course free to develop solutions elsewhere, perhaps using Cygwin's `g++` on your Windows machine, but be sure to allow yourself adequate time to ensure that your solutions work on `lectura`, too. Be especially careful if you use a C++ compiler outside of the `gcc` family, like the Visual Studio C++/CLI compiler.

My view is that it's a Bad Thing to give credit for code that doesn't work. **Programs that don't compile or that mostly don't work will earn a grade of zero, regardless of how close to correct they might be.** Additionally, non-general solutions, which typically have the expected output "wired-in", will very likely earn a grade of zero.

My view is that programming assignments are to help you learn the course material—I don't view an assignment as a take-home exam. As a rule, you'll learn more if you can get through an assignment without asking for a lot of help. However, if you reach a point where you simply aren't making progress and you're running out of ideas or time, then you surely should ask for help.

Each homework assignment will specify a due date and local time. **As a rule, late homework solutions, even if only one second late, are not accepted and result in a grade of zero.** Assignments will be submitted using `turnin` on `lectura`. The time on `lectura`, as reported by the `date` command, is considered the official time. (Let me know if you should ever see `lectura`'s clock deviate by more than a second or two from an accurate and credible time source such as `time.gov`.)

Deadline extensions may be granted to the class as a whole if problems arise with an assignment or with departmental or university computing facilities.

Extensions for individuals may be granted in certain cases. **The key factor that can lead to an extension is that due to circumstances beyond the student's control, the student's work is, was, or will be impeded and it is impractical or impossible for the student to make up for the lost time.**

Accident, illness, and personal/friend/family crises are examples of circumstances that I generally consider to be beyond a student's control. On the other hand, for example, an extension due to lots of work being due in other classes is very unlikely. Travel, such as an interviewing trip, may merit an extension but pre-trip approval of the extension is required. Ultimately, however, each situation is unique; you are strongly encouraged to contact me if you believe an extension is warranted. If you believe an extension is warranted, DO NOT work on an assignment (or even think about it) after the deadline is passed; set the assignment completely aside and wait until an extension is granted.

Extensions for individuals are granted in the form of an amount of time, such as eight hours. An eight-hour extension can be pictured as a count-down timer with an initial setting of eight hours. The timer runs whenever you are working on the assignment, whether that be typing in code or simply thinking about it. You will be on your honor to keep track of that time and not exceed the amount granted.

All holidays or special events observed by organized religions will be honored for those students who show affiliation with that particular religion. Absences pre-approved by the UA Dean of Students (or Dean's designee) will be honored.

Quizzes

There will be some number of "pop" quizzes. Quizzes will typically have a handful of short questions, be allocated three minutes or less, and be worth five to fifteen points. There will be a total of 100 points worth of quizzes during the semester. Quizzes may be conducted at any time during the class period. In some cases one or more quiz questions may be "exploratory"—perhaps to see what portion of the class grasped some just-presented topic, for example. Simply handing in a paper will earn full credit for such questions.

If circumstances beyond your control (as outlined above for assignments) cause you to miss a quiz, let me know.

Bug Bounties

A "bug bounty" of three homework points of extra credit will be awarded to the first student to report a particular bug in an assignment. Bugs might take the form of errors in examples, ambiguous statements, incomplete specifications, etc. As a rule, simple misspellings and minor typographical errors won't qualify for a bug bounty, but each situation is unique—you are encouraged to report any bugs you find. Any number of bug bounty points may be earned for an assignment and will be added to the grade for that assignment.

Bug bounty points may also be awarded for first reports of bugs in the slides, e-mail messages, and this syllabus. Such points are added to the next homework assignment.

Office Hours

I put forth an interesting deal when I proposed to teach this class: (1) I'd do it for no pay; (2) Time spent on the class would not count toward my 40 hours/week for my position on the AnimalWatch project; (3) I would try to minimize impact on my AnimalWatch work.

I truly enjoy working with students whether it is in person, via e-mail, IM, or the phone. I believe that being available to work with students outside the classroom is a vital responsibility when teaching a course. I will do everything possible to make myself accessible to you but AnimalWatch is my top priority.

What I'd like to try at first for office hours is a simple "open door" policy—feel free to drop in any time my office door is wide open. I'm usually "in" by 9am every day, often earlier, and usually don't leave before 6pm, aside from lunch somewhere in the middle. Here's the catch: if an AnimalWatch issue needing my immediate attention should arise while we're talking, our conversation may need to pause for a few minutes or even be terminated. Typically, mornings are more subject to urgent needs than afternoons; late afternoons provide the best chance of being interruption-free. If should you find me in a nearby colleague's office you should think of me as "out"; in some cases I may be able to offer an estimated wait time, but not always.

We'll see how this policy works out, both for the class and for AnimalWatch, and make changes if necessary, perhaps instituting limited but guaranteed blocks of office hours during the week. I welcome your feedback.

I prefer to conduct office hours in a group-style, round-robin manner. You needn't wait in the hallway if I'm working with another student; you may join us and listen in if you so desire. If several persons each have questions, I will handle one question at a time from each person in turn. I will often give priority to short questions (i.e., questions with short answers) and to persons having other commitments that constrain their waiting time. (Speak up when you fall in either of these categories.) If for some reason you would like to speak with me in private, let me know and I will clear the office.

Students who make proactive, not reactive, use of office hours usually achieve the best results. Proactive use of office hours includes asking questions about material on the slides and in the texts, asking questions about how to use tools, discussing how to approach problems, etc. If you're familiar with Covey's *The Seven Habits* *of Highly Effective People* you might view those as "Quadrant II" activities—important, but not urgent.

Reactive use of office hours is typically centered around simply getting a program to work one way or another ("Quadrant I"—important and urgent) rather than viewing it as an exercise to put new concepts and techniques into practice.

E-Mail

E-mail is a great way to handle some types of questions; feel free to use it. When answering mail, I give priority to well-focused questions. And, it's often the case that the task of developing a well-focused question will lead you to an answer on your own.

My goal is to respond to mail promptly, overnight at worst. If you think a message may have been overlooked, feel free to send it again.

The first step I often take when a mysterious problem is reported is to attempt to reproduce the problem. A common mistake is to send only a "relevant" excerpt of the code when in fact additional code is needed to reproduce the problem. The best thing to do is to `turnin` the full set of involved files—so I can compile and possibly run your code—and then send me a note about the problem, mentioning that you've turned in the code.

It's sometimes a bad idea to put multiple unrelated questions in a single message—I usually answer a message as a whole; the time required to address the full list may delay my response. If you have several independent questions, it's often best to send them in separate messages.

If you mail me with a question but then resolve it before I get a chance to reply, I'll surely appreciate it if you send a follow-up note to let me know that the issue is closed, sparing me from writing an unneeded response.

Phone Calls

It's sometimes the case that a five minute phone call can accomplish the same result as an e-mail message that takes fifteen minutes to write and ten minutes to answer. If you think a phone call might be the best way to pose a question, feel free to give me a call

Please don't worry about "bothering me" with a phone call; if I don't want to deal with a phone call at a particular time I turn off the phone or simply don't answer it. The window in which I'm most likely to answer the phone is 8am to 9pm, seven days a week. If I don't answer, I'll appreciate it if you'll then e-mail me instead of leaving a voice message.

Mailing List

A Google Groups mailing list will be used as the electronic forum for this class. Messages from me might include information such as supplemental material, assignment corrections, lecture clarifications, due-date changes, etc. **The material I post to the list is considered to be part of the course material** so it is important that you be on the list. Go to `http://groups.google.com/group/cs397a/subscribe` to join.

Please don't view the mailing list as read-only! I hope you'll participate with questions and comments related to the course material, testimonials for handy tools, URLs for interesting websites or blogs, and the like.

Original Thoughts

In the movie comedy "Broadcast News" a 14 year-old high-school valedictorian receives a post-commencement beating from a group of bullies. After picking himself up, one of the verbal spears he casts at his attackers is, "You'll never have an original thought!" That notion of an "original thought" has stayed with me. I hope that you'll have some original thoughts during this semester.

I offer an award of one point on your final average for each Original Thought that you claim as such and that strikes me as significant. Observations, analogies, quotable quotes, and clever uses of tools and language constructs are some examples of things that have qualified as Original Thoughts in my classes. Note that an Original Thought does not need to be something that's probably never been thought of before; it just needs to be something that I consider to be reasonably original for you.

Of course, an Original Thought needs to be something you've thought up yourself—don't submit something you found elsewhere, like a quote, just because it strikes you as being original!

Distracting Classroom Behavior

A student doing anything that distracts or otherwise impairs the learning of others may be asked to either stop that behavior or leave the classroom for the balance of the period. Here are some examples of things that might distract students that are beside or behind you: gaming or other laptop usage unrelated to the lecture, more than a little texting, and watching cell phone videos. If there's a distraction I'm unaware of, sometimes a frown and a prolonged look at the offender is enough to let me know that there's a problem.

Academic Integrity

It is unfortunate that this section need be included but experience sadly shows that some students are willing to cheat their way to the grade they desire. For those students who would never do such a thing, I apologize for the inclusion of this section.

Nutshell summary: Don't cheat in my class and don't make it easy for anybody else to cheat. One strike and you're out!

You are responsible for understanding and complying with the University's Code of Academic Integrity. It can be found at `http://dos.web.arizona.edu/uapolicies`. Among other provisions the Code demands that the work you submit is your own and that submitted work will not subsequently be tampered with. Copying of another student's code or answers is prohibited when they are part of an assignment; it is immaterial whether the copying is by e-mail, IM, pencil and paper, or other means. Allowing such copying, thus facilitating academic dishonesty, is also a Code violation.

In addition to ruining one's grade and damaging one's future, the processing of an academic integrity case requires hours of work by myself and others. I am happy to spend hours helping a student who is earnestly trying to learn the material, but I truly loathe every minute spent on academic integrity cases.

A violation of the Code will likely result in all of the following: (1) failure of the course, (2) the following permanent transcript annotation: "FAILING GRADE ASSIGNED DUE TO CHEATING, and (3) disallowance of GRO for the failing grade.

It is difficult to concisely and completely describe where helping a friend learn stops and cheating starts, but here are some guidelines:

- I consider it to be reasonable to work together on assignments to get to the point of understanding the problem and the facilities that are required for the solution.

- I consider it to be reasonable to help another student find a bug only if your solution for that problem is further along than the one being examined. For example, if you haven't started on a problem, you shouldn't be hunting bugs in another student's code for that problem.

- I consider it to be reasonable to exchange test cases, unless test cases are one of the deliverables for a problem.

- It is surely a mistake to give another student a copy of a solution or a significant portion thereof.

- If you receive help on a solution but are unable to fully explain how it works, it is likely a mistake to submit it as your own work.

- If your gut feeling is that you're cheating on an assignment or helping somebody else cheat, you probably are.

- If in the heat of the moment you submit a solution that is not fully yours, or give your work away, and you later reconsider your action, your only consequence will a grade of zero for the involved problems **if** you confess before your act is discovered. Conversely, further dishonesty when confronted, which invariably increases the time expenditure, raises the likelihood of more extensive penalties, including a recommendation for suspension or expulsion from the university.

Cheating almost always starts when one student has easy access to the solutions of another. You are expected to take whatever steps are necessary to guard your solutions from others in the class. For example, you should have a non-guessable password and never share it. Hardcopy should not go into even a locked recycling bin—take it home and dump it in a recycle bin when the course is over. Personal machines, be them laptops or home desktops, should be behind a hardware firewall or running a software firewall. If you and another student share a computing facility off-campus, perhaps as co-workers or roommates, using that system to develop solutions may be of very questionable merit, depending on the privileges of the other student.

Failure to take reasonable precautions to ensure the privacy of your solutions may be construed to be facilitating dishonesty, a Code violation. For example, having a guessable password such as, but not limited to, your first name, your last name, your initials, or your pet's name could be viewed as facilitation of dishonesty. Leaving a logged-in and unlocked machine unattended in the presence of another person, even a friend, is very questionable.

Be very careful when typing your password in the presence of others, be them friends or strangers. The single worst cheating case I've ever handled started when the password of a two-fingered typist was surreptitiously observed. Don't hesitate to ask someone, even a friend, to turn their head when you're typing your password. Of course, using a common password makes theft by observation easier—avoid them! See http://www.openwall.com/passwords/wordlists/password.lst for one list of common passwords.

Accommodation

Students with disabilities, who may require academic adjustments or reasonable accommodations in order to participate fully in course activities or to meet course requirements, must first register with the Disability Resource Center. The DRC staff will qualify students for services, and provide a letter to me listing accommodations to be made. This letter should be submitted by the student directly to the me as soon as possible during the first week of classes. The student should meet as soon as possible with me by appointment or during office hours to discuss accommodations and how course requirements and activities may impact the student's ability to fully participate.

Threatening Behavior

Threatening behavior is not tolerated. University policies on threatening behavior can be found at `http://policy.web.arizona.edu/~policy/threatening.pdf`.