CS 2C Syllabus - Spring 2023

Algorithms and Data Structures in C++

"*You cannot beast mode code your way through this class*" - John Vicino

Hey there. My name is Anand. Please read this syllabus carefully. You should especially read it if you have never taken a class from me before.

Jump To

Intro

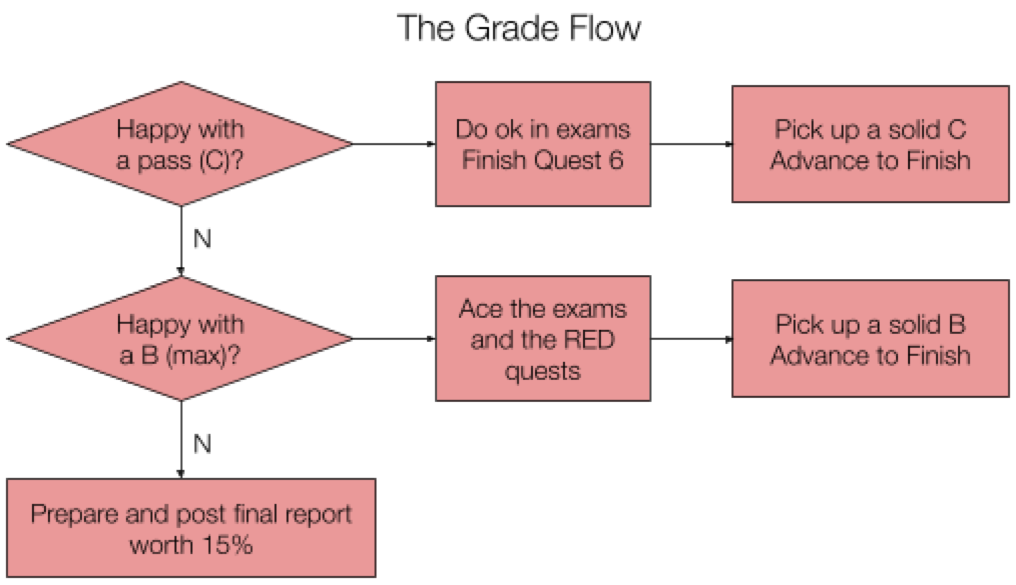
This class is different from other traditionally taught classes. I find more students tend to like it this way than not. However, it takes a little getting used to.

Note the following **essential** prerequisite skills.

- Looking up, evaluating, and using information on the net
- Following simple directions correctly (e.g. creating a subreddit user according to some requirements)

In my classes, you will largely self-teach yourself most of the material with help from your classmates. Your tutors and I will be watching, careful not to step in and give you the answer. Yes - You will NOT be given the answer. You cannot ask anyone to debug your programs for you. You have to find it yourself, although we will give you all the help we think you need to succeed.

Read the advice for incoming students by senior students who took this class.


The Grade Flow
Happy with a pass (C)?
Do ok in exams Finish Quest 6
Pick up a solid C Advance to Finish
N
Happy with a B (max)?
Ace the exams and the RED quests
Pick up a solid B Advance to Finish
N
Prepare and post final report worth 15%

If you just want to be able to fill ordinary (resource-type) programming jobs that don't require heavy-duty knowledge of CS fundamentals or Data Structures, you can pass this class by completing Quests 1-6, and doing ok in the exams/quizzes. At the end of Quest 6, you should be intimately familiar with at least the hash table ADT, which will be behind many applications you may create (e.g. using a python dictionary).

For a B, you should do well in your quizzes and exams, and also complete all 9 RED quests, without necessarily acing anything.

To get into A territory, you absolutely need to prepare the best final report to happily showcase on our sub.

CS 2C Readiness Self-Calibration

This is a **challenging** class according to most former students. How do you know if you will be successful? Try this:

In this class, there are two required technical tasks to test your preparedness. If you cannot complete them both in 10 days you will be dropped for your own benefit, so you can return when you are better prepared to succeed.

- 9 BLUE quests (essentially ALL the homework of CS2A) is due on day 3 (Wed of week 1). You only get 3 days to complete what CS2A students get 3 whole months for.
- 9 GREEN quests (essentially ALL the homework of CS2B) is due on day 10 (Wed of week 2). You only get 7 days to complete what CS2B students get 3 whole months for.

Even though you may already be proficient in these concepts or have passed CS2A and CS2B (or their equivalents) elsewhere, you still need to complete these quests. I allocated grade points to them commensurate with the amount of effort they should take CS2C beginners.

If it takes you longer than 2 days to complete all the BLUE quests, or longer than 7 days to complete all the GREEN quests, you should seriously reevaluate if you want to do CS2C this quarter or later or review the amount of effort you are willing to put in each day.

Important note: To keep your self-evaluation meaningful, you must **NOT** refer to ANY past-subreddit posts, nor search for solutions online. You may converse freely with any *current* quester in the appropriate subreddit. Each quest is designed to be solved using only the tools you should already be familiar with when you reach them (plus your own creative thinking). *This is the way the very first batch of questing students had it, and I don't believe that you are less capable than any of them.*

What does the data say?

Students who complete GREEN 9 by day 10 have an 73.9% chance of a A- grade or better

One by-product of this necessary exercise (if done sincerely) is that you will get a taste of the pace of work required for this class. You cannot return to the class from time to time to make progress. It has to become a central theme of your life for the remainder of this quarter and you must either be programming or thinking about programming at least 4+ hours each day. To make it more worthwhile, my participation points award system is tuned to reward predictable periodic forum behavior and non-bursty progress.

Administrative stuff

CS 2C continues where CS 2B left off. Students already comfortable with intermediate-level C++ programs will have an opportunity to play with and master important data-manipulation algorithms.

This course provides a systematic treatment of advanced data structures, algorithm analysis and abstract data types in the C++ programming language. Coding topics include building ADTs on top of the STL templates, vectors, lists, trees, maps, hashing functions and graphs. Concept topics include searching, big-O time complexity, analysis of major sorting techniques, top down splaying, AVL tree balancing, shortest path algorithms, minimum spanning trees and maximum flow graphs.

A strong familiarity with algebra and good written English comprehension skills are both advisories. Please arrange tutors/helpers right away if you need assistance with either. Contact the STEM Center or the TLC.

Course outline and SLOs

You can access the official course outline of record for all CS courses here. Student Learning Outcomes for this course are:

1. A successful student will be able to write and debug C++ programs involving advanced data structures such as Lists, Trees, Graphs. They will be familiar with the use and implementation of algorithms for balancing binary trees, creating splay trees, minimum spanning graphs, finding the shortest path through a graph, and maximum flow through a network. They would also be familiar with the most common sorting algorithms and know the advantages and tradeoffs of each.
2. A successful student will be able to reason about the running time and derive properties of computer programs using precise mathematical terminology. Specifically they will be conversant with the Big-O notation and be able to craft efficient algorithms using the appropriate data structures to solve non-trivial computational problems.

Canvas and **required tasks**

Students at Foothill College doing this class for credit should use Canvas to coordinate some activities and take online exams.

Most of the rest of our work will happen at other public locations, including youtube, zoom, quests, reddit, etc.

You will start your adventure in Foothill's class by completing the following **required** tasks.

1. posting an introductory note about yourself in Canvas (You can simply reply to my own introductory post if you prefer) and
2. scoring at least 90% on the syllabus quiz (in Canvas). This doesn't need knowledge of CS or c++.

☆ These **must be completed before the end of the first Friday** of the quarter to prevent being dropped.

Assessment

If you're doing this course for kicks, or other fun reasons, you can skip this section.

If this course is offered for a grade, and you are taking it for a grade, then, your final grade will be based on programming quests (homework), exams, and conceptual understanding/ability to help other programmers.

Worksheet

In the worksheet below, activities marked with an asterisk (*) are **required** activities. This means that if that activity is not completed then nothing below it will count towards your grade. *Pupping* a quest means progressing until you get the password to move on. *Dawg* means *detailed attentive work given*, which is proxied by collecting the stated number of trophies for that color).

Activity	**Worth**	**Your score**
Syllabus quiz*	1%	
BLUE quests (pup)*	3%	
BLUE quests (dawg = 180 trophies)	2%	
GREEN quests (pup)*	5%	
GREEN quests (dawg = 240 trophies)	4%	
RED quests (265 trophy cap)	45%	
Midterm and Final	25%	
Final report	15%	
Total	**100%**	

Then use the absolute grading scale below:

For an	A+	A	A-	B+	B	B-	C+	C	D	F
You need (%)	97	91	88	86	80	78	75	67	60	< 60

Quests need to be pupped in sequence before their freeze dates. However, once pupped, they can be dawged any time before Friday of W11. Email me when you're sure you have dawged a RE quest.

Exams

Exams are objective style and will be administered via Canvas. You will typically have a window of time (18+ hours) during which you can begin these exams. But once you begin, the current version of Canvas does not allow you to *pause* your exam and come back to it. The 1h (or 2h for final) timer cannot be stopped once you start it, until you hit finish.

They can be taken anywhere you get a decent Internet connection. But I don't recommend taking them on mobile devices. I also suggest that you don't actually try out code-fragments from your question sheet in an IDE.

Final report

The final report should be a post in the subreddit summarizing your contributions during the entire quarter.

To score high on this, your report should be prepared as if it is a portfolio for a potential hiring manager who may look you up and review your reddit posts to gauge your level of expertise and ability/willingness to learn new things. You can use this as an opportunity to build a compelling portfolio of technical musings and discussions you might want to present to someone.

It should contain links to *selected* posts and comments (with short descriptions) spanning the entire quarter, not just all made in a few weeks.

For examples and templates, you can look at some of the excellent reports from Winter 2023 students:

- CS2A - Participation log by Namrata Keskar
- CS2B - Participation log by Dylan Slocum
- CS2C - Participation log by Yamm Elnekave

I just picked random examples, of course. You can go to the respective subreddit (r/cs2a, r/cs2b or r/cs2c) and simply search for posts with the word "participation" in the title. You are bound to find many fine examples.

Subreddit username

Even if you already have a reddit username, you must create a new one for this class. Your reddit avatar name **must** start with your first name (as on Canvas) and an underscore, followed by your initial (or full last name) + some optional digits (example: *ramanujan_s1729*). If you already have a conformant reddit username from your CS2A or CS2B enrollment, I highly recommend you use it so that people who click on the username can see your contributions over the entire series.

Only posts and comments made by usernames matching the above format are eligible to be linked into your final report.

IMPORTANT: Even though our subreddits contain content created by past students, you must NOT review any past content on a quest - honor code. You can view and comment as much as you want on any content for a particular flair after you have *dawg*ed that quest.

Do not publish tip sheets or cheat sheets. Do not share actual quest code.

Extra Credit Work

Extra credit work will be offered to selected students who manage to dawg all 9 quests before the end of week 7 and maintain a good, helpful and kind profile on our subreddit.

Schedule and Homework Deadlines

Programming, like all art, is not a 9-5 job. Sometimes you're on a roll and killing it. Other times, not so much.

I know how it is.

So there are no regular papers or labs due every day or week in this course. Rather, like real projects, there are deadlines you should strive to meet. You can plan your own time in your own way. Below is one suggestion:

Week	Read References	Complete	Notes
1	Algorithms and ADTs review	All BLUE some GREEN	
	Algorithm analysis	WRAP UP GREEN	
3	Time complexity and Big-O	Mystery Quest 1	Q1 Freezes Mon 12am
	General trees (and BSTs)	Mystery Quest 2	Q2 Freezes Mon 12am
5	AVL Balancing and Splaying	Mystery Quest 3	Q3 Freezes Mon 12am
	Review/Midterm (Canvas)	Mystery Quest 4	Q4 Freezes Mon 12am
7	Hash tables Quadratic probing	Mystery Quest 5	Q5 Freezes Mon 12am
	Sorting	Mystery Quest 6	Q6 Freezes Mon 12am
9	Priority Queues, Heaps, Heapsort	Mystery Quest 7	Q7 Freezes Mon 12am
	Dijkstra's and Kruskal's algorithms	Mystery Quest 8	Q8 Freezes Mon 12am
11	A Maxflow algorithm	Mystery Quest 9	Q9 Freezes Mon 12am
	Final Exam (Canvas)		

You only get 1 week to complete each quest - hard-deadlines in CS2C

Every week, give yourself one or more topics to study and one or more programming quests to complete. By topics I mean either chapters in the assigned text, or better - a distillation of essential representative concepts from the text, prepared by ex-prof Michael Loceff. You will find links to them in the Resources section.

Make sure you have adequate time to code, experiment, revise and recode EVERY week. This takes a LOT longer than many students estimate. Also see: The bell rings at 11:59.

☆ Note that you may NOT speed solve several quests in a week. You cannot take off from class, disappear for a week or more, and then come back to catch up on missed quests without prior permission.

Finally, note that you cannot leave a hole in your questing trail. Your trophies for Quest$_n$ only count after you have at least pupped all quests < n-1

Extensions

Extensions don't make sense because the quests are self-paced. You just have to complete each by their "freeze" date to get credit. After their freeze dates, you should still complete them, but not for credit. There's a LOT of time to complete these quests even if you have to take some breaks. So, there is no need to ask for extensions.

If you miss a deadline for a quest, you must still complete it and submit a petition with valid cause after you pup Q9 for your frozen scores to be considered.

Disability Resource Center

Foothill College is committed to providing equitable access to learning opportunities for all students. Disability Resource Center (DRC) is the campus office that collaborates with students who have disabilities to provide and/or arrange reasonable accommodations.

If you have, or think you have, a disability in any area such as mental health, attention, learning, chronic health, sensory, or physical, please contact DRC to arrange a confidential discussion regarding equitable access and reasonable accommodations.

If you are registered with DRC and have a disability accommodation letter of accommodations set by a DRC counselor for this quarter, please use Clockwork to send your accommodation letter to your instructor and contact your instructor early in the quarter to review how the accommodations will be applied in the course.

Students who need accommodated test proctoring must contact the DRC immediately if they cannot find or utilize your MyPortal Clockwork Portal. DRC strives to provide accommodations in a reasonable and timely manner. Some accommodations may take additional time to arrange. We encourage you to work with DRC and your faculty as early in the quarter as possible so that we may ensure that your learning experience is accessible and successful.

To obtain disability-related accommodations, students must contact the Disability Resource Center (DRC) as early as possible in the quarter. To contact DRC, you may:

Visit DRC in Building 5400, Student Resource Center (physical visits suspended during college closure).

- On the web: http://www.foothill.edu/drc/
- Email DRC at drc@foothill.edu
- Call DRC at 650-949-7017 to make an appointment

Important Dates

For a list of important dates for the fall quarter, see the official college page here.

Virtual Catch-up Circles

Even though your section may be purely online, we will have **weekly check-in virtual meetings.** These meetings may last from anywhere between 1 to a couple of hours. During the first week of class, a time slot that accommodates the most number of students will be chosen for this meeting.

Consider this meeting the equivalent of lectures for those who are familiar with my online lectures. Professionals in the field taking this class out of interest may find these meetings similar to their Weekly Status meetings, except that ours will take place in a supportive, moderated environment in which we can point you to information or lessons that can help unblock you.

Interested to see what these meetings may be like?

See https://www.youtube.com/watch?v=qdHskvhqHH8&list=PL7PAwfkmE8mqnokMLBJaZAwJ6F9V9Poen

What if you cannot make these online catch-up circles via zoom? Then you must schedule private 1-1 meetings with me during which I'll get a chance to review your progress, give pointers on what to do next, and enter a suitable proxy for this score in my private spreadsheet.

Note: *I try to keep everything public, including 1-1 meetings you may have with me unless it involves confidential matters. Foothill offers a number of qualified counselors who can discuss confidential matters with you to suggest good solutions.*

Any class-related discussion that may be generally useful will be made available as transcripts or recordings at publicly accessible media sites (e.g. YouTube or Reddit). Make sure you are ok with this policy before enrolling.

Communication

Please use our sub for any question or comment that relates to the quests (except questions of a private nature). If you have a confidential question (grades or registration) you can email me. If you have a question that only makes sense of material you can find in Canvas (e.g. modules, syllabus, exams, etc.) then it makes sense to post that question in Canvas rather than our sub.

Try to meet with each other after class (even if virtually), set up private study and programming groups and work on independent (non assignment) programming challenges outside of class. I'll give you a few interesting challenges from time to time. Some of these may earn you extra credit.

You can reach me via messaging in Canvas, Reddit or by email. While on campus, my room number is `0x113d` (in hex). In week 1 of CS2A, you would have learned how to decode that into decimal.

Infinite Office Hours!!!

Well, within reason, ofc.

One time I happened to be in my office at 2AM. Someone knocked on my door.

> "Who is it?" I asked.
>
> "Quick question prof" said a female voice.
>
> I opened the door. "Hey sorry. My office hours are at 10am. See the sign?" I pointed at the print out I had stuck to my door. It clearly said 10AM - 11AM in **BIG BOLD BLACK** letters.
>
> "Yeah" she said. "I read it. But I'm a binary janitor. So I waited up until now to come here."
>
> "That's cool. You definitely deserve an answer" I said. "What's your question?"
>
> "Would you like me to come back later to empty your trash can?"

This quarter, I've decided to do away with this confusion. I have open office hours ANYTIME (virtual). I am very flexible in being able to adjust to your availability. But you have to pre-arrange it with me via email first.

I will likely be able to point you in the right direction for further experimentation. But I will not look at your questing code directly, nor debug it for you. Nor can you ask anyone else to do it (honor code).

Appendix A - How to Quest

This part of the syllabus is optional. It is meant to ease your journey on the way to a successful completion of this class.

Mystery Quests (capped at **265** for CS2C this quarter)

You will solve these at the public questing site (https://quests.nonlinearmedia.org).

These quests are meant to be solved **in sequence.** Each quest will give you a certain number of trophies. You can check your total trophy count at any time by visiting your personal scoreboard at the /q site (It will be wiped on the 1st of Jan, Apr, Jul, and Oct). *It will show you all your trophies, but only the ones you earned for 2C (RED ones) count for this course.* Your secret handle is your Student ID

The quests are set up such that the password to each quest is given out upon scoring a certain number of trophies in the preceding quest. However, I found that a few students were getting stuck in the lower numbered quests pounding away at them to eke out every remaining trophy before moving on, even though they had already earned the password. This is a bad strategy. Keep moving when you get a password. You can always come back to polish your previous quests when you have free time before the freeze date. You get about ONE week (7 days) before each quest freezes.

At the end of the quarter, your total trophy count will be capped at **265** and fed into the grade crunching system. If you spend a lot of effort getting up to high numbers by the time you get to Quest 7 already, then you'll be close to getting burned out right in time for two of the funnest quests of all. So plan your time and effort wisely. It's not like your old quests are going to disappear when you move on.

Pupping and Dawging quests

Solving minis until you get the password to move on from a quest is called pupping it. This is what I recommend you do in your first pass through the questing trail.

Dawging it means you review the code and the spec in detail and try to get all available trophies. I don't think anyone knows how many trophies there are per quest, but it's usually easy to tell if you've got everything obvious.

Dawging a quest gives you 2 extra trophies for it. With 9 quests, that's 18 extra trophies. That's why the trophy cap is higher this quarter than the last.

What is a Quest Freeze?

A freeze is when I will review your submissions and transfer scores from a particular quest into Canvas.

If a quest has frozen, you still have to complete it in order to move forward into the next quest, but your trophies for it will not be counted towards your grade!

Bummer… Yes? If I were you I would try my best to avoid that situation by staying on top of things and taking this class ***seriously***.

Getting started on your mystery trail

The password for the first quest is A Tiger named Fangs. Passwords for subsequent quests will be automatically revealed upon *satisfactory progress* (as the machine sees it) in each preceding quest. The first five BLUE quests range from trivial to simple and are meant to give you an opportunity to get familiar with basic control structures and the questing system. You're best to finish them in a single sitting.

What does the data say?

Students who don't complete Quest 4 by week 6 have an 61.7% chance of a C-grade or worse

In order for rewards from a quest to count towards your total, you must have completed all previous quests. If you leave a hole in your trail of completed quests, then your total reward earnings is the sum of all rewards you earned before the first incomplete quest.

You can quest as much as you want. Everything you do is logged. Quest with your ID and we can watch your struggles and progress, and know when it is best to help.

Bugs in your code?

Getting your code debugged by someone else is NOT allowed. That includes me, tutors, teachers, friends, enemies and relatives. Debugging your own code is an essential skill that aspiring programmers must learn and enjoy - Yes, enjoy!

Of course, I can't police this. But your enrollment in this class signifies acceptance of this condition (in addition to being bound by Foothill's Academic Integrity Policy). You cannot send your code to me, a tutor, or someone else and ask them what the issue is. What you can do is:

1. Explain (in our **subreddit**) what you're trying to do
2. Describe in English the steps you think might work, or if you have no idea and would like someone to explain the requirement better.
3. Or describe the behavior of your program and ask why it is not working as expected.

Sometimes it is also ok to post your code on our **subreddit**. Mostly, exercise good judgment regarding what can be shared. You want a fun and fulfilling learning experience. The best way to get it is to keep it fun and fulfilling for everyone. You wouldn't give away a movie's ending to a friend who's going to watch it. Why give them the solution to a problem when they can feel good finding it themselves?

Programming style and compilers

My personal preference for program formatting is the C++ equivalent of the classic K&R style for C. It's not imperative that you follow the K&R style. I'm ok with any consistent and clean styling/formatting of your programs.

Use an IDE/compiler of your choice. But you'll find better support from me and the STEM center if you stick to one of the environments we know about (ask).

Appendix B - Recommendations for Reddit

Don't say anything in the forums that you'll end up regretting later in your life. OTOH do try to let your natural genuine curiosity shine through for others to seek out in a sea of wannabe programmers. Maintain your profile on our subs as you would if you were a professional and it will free up a lot of your time.

I will try to remove posts that I deem (in my subjective opinion) to be a liability to your future self. But you can't rely on it. Best to be helpful, courteous, informative and only post useful and interesting observations without overtly giving the answers away. They usually have more lasting value.

👉 ANY user anywhere in the world can quest and post/discuss in our subreddits. So you may see posts and replies by users with anonymous names like *coding_lion*, *bat_girl* and such. All posts are subject to the same rules like *Johnny be good*, but only the ones with avatar names matching the spec in this syllabus will get participation credit:

Can you review past subreddit posts?

Access to past posts in the subreddits **AND to posts giving tips/hints from current students** is off-limits to current students (also recommended for current non-student questers).

Imagine you are on equal footing with past students who did NOT have these resources but were still able to do well in this class.

You are expected to use the subreddit to communicate with your current questers, and to demonstrate both your conceptual understanding and willingness to help others technically.

After you successfully complete a quest on your own (with possible contemporary help), you can refer to all past posts for that quest.

Can you write Tip Sheets (or read those by others)?

No. NO points will be awarded for writing tip sheets. I also found that some weaker students tend to wait until their stronger trailblazer classmates publish hints/tips in order to follow these tip sheets and pass quests. This doesn't help their learning. Therefore you may also NOT refer to tip sheets of any kind in solving the quests and must instead be open with your struggles as you conquer each new quest.

Those who pass quests without showing evidence of having understood the core concepts will now be flagged for individual review and private exam/quiz.

Final report

A good final report has links to the following types of content with a 1-line summary of each link. See examples on our sub.

- Posts that sparked rich discussion (lots of good comments)
- Posts that say something insightful
- Posts that offer helpful and kind advice to other students who may be struggling
- Posts that share personal work that are relevant to 2C (along with an explanation why)
- Posts over the entire duration of the quarter, rather than in spotty bursts once in a while

Appendix C - Resources

Text recommendation

My first resource suggestion is the book: *Data Structures and Algorithm Analysis in C++*, any edition ≥ 2nd, by Mark Allen Weiss.

Online modules (free, courtesy of Michael Loceff)

I also have a fork of CS2A/B/C modules that ex-prof Michael Loceff created when he taught this course. Thanks to Michael, I'm able to make these available to everyone completely free.

Click on the appropriate link to access them: cs2a, cs2b, cs2c

Although a couple of revisions behind, much of it is still relevant to this course. It is essentially a *distillation* of selected topics from the text. But be aware of salient differences between the content of his modules (or the text) and what some of our quests require. This shouldn't be a problem if you understand the concepts. But it will be a problem if you don't.

As always, hit our ***sub****, when in doubt.*
Actually, that's not quite it.
When in doubt, try it out.
If you still don't get it, then hit our ***subreddit*** *(to ask - not to look up)*

Lane's Lane

The Foothill STEM Center already provides fantastic assistance by making experienced CS tutors available for 1-1 real-time (synchronous) assistance almost 24/7 (via zoom) and generous hours in the STEM Center when the campus is open. Within the STEM Center, Lane Johnson hosts two special workshops each week focused especially on helping questers. Look for their actual hours on our **sub**, or simply check into the STEM Center sometime and ask for Lane.

Other Resources

The department maintains a blog called Opportunities for CS students. It has announcements of internships, scholarships, free software offers, public lectures, etc. You can also check out our numerous SLI opportunities.

Bottom line

Computer science is a **hard science**, not a soft science. A significant investment of your time and effort will be required in this class. To succeed in CS2C, you must expect to code at least 2 hours EACH and EVERY single day for the next 12 weeks. Make sure your schedule allows it before you start. Now smile, and get ready to get wasted (in a good way). If you apply yourself sincerely, you will learn a REALLY USEFUL skill for a happy life.

Happy Hacking!

&