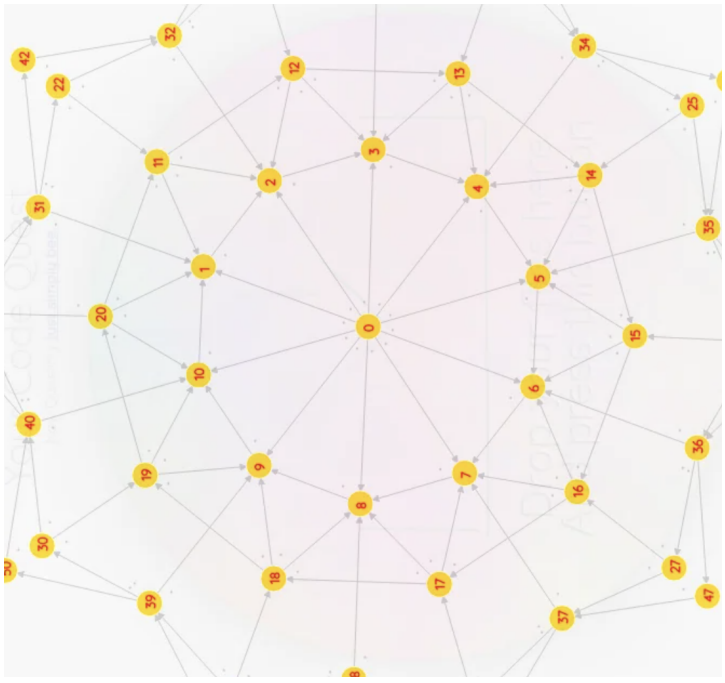
CS 2B Syllabus - Fall 2023

Intermediate OO Programming in C++

Ordinary people have opinions. Extraordinary people have knowledge
Gain Knowledge. Unleash your Genius



Web, by Heath Bacon, Summer 2023
Final homework submission for CS2B
https://www.reddit.com/r/cs2b/comments/15k5kol/quest_9_thoughts_and_picture/

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.

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. It is like a **bootcamp** for those who want to learn C++, one of the most powerful programming languages.

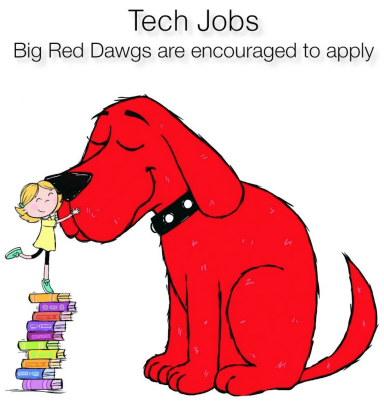
YES - This quarter, CS2A and CS2B students will be required to do their homework assignments at the Genius Bootcamp. Those who are enrolled in this class are auto-enrolled in this bootcamp for free. I have arranged to get notifications and access to all of your code submissions, and for your scores to be transferred to Canvas on their due dates. This bootcamp is an incredibly fun and supportive community experience to learn CS in. However, you should not expect answers or hand-holding by anyone here either (as these are discouraged).

Helping others does not come automatically to MOST people who would rather *do it* for you or tell you *how to do it*. Learning to help without destroying one's joy of finding out for themselves is usually a learned skill, and a SUPER VALUABLE one for those of you who aspire to be future leaders. Practice here.

You should ideally start questing and participating in the bootcamp as early as possible after you get this syllabus in your hands. More details on this later in the syllabus. Here is how my class homework problems align with Genius.

- CS2A Homework gets you through the BLUE level of the bootcamp (***Required*** *for 2B grade points*)
- CS2B Homework gets you through the GREEN level of the bootcamp
- **Belief in yourself**, courage, patience, and perseverance make you a RED DAWG

Read some recent final reports left by your senior students who took my class.



CS2B Self-Calibration

In the first week (summer students only get 3 days) you must complete all 9 BLUE quests[1] for credit. If you have already done so as part of your CS2A or the Genius Bootcamp, you can reuse those credits, but you will have to resubmit your source files into the questing system.

Strong familiarity with the concepts introduced in CS2A is an absolute must in order to succeed in this class. To calibrate yourself you can use the speed with which you completed the BLUE quests (required for grade points) and use the chart below (based on about 2 quarters worth of data):

Time to complete ALL BLUE quests	**Expected 2B difficulty level**
< 1 day	Easy
3 days	Moderately difficult
4 days	Very difficult
5+ days	Likely drop or fail unless serious remedial steps are taken

Important note: 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 open subreddit without sharing code or getting direct debugging help. 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.*

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 2+ hours each day (4+ in summer). To make it more worthwhile, my points award system is tuned to reward predictable periodic forum behavior and non-bursty progress.

[1] More on quests later.

Administrative stuff

CS2B continues where CS2A left off. Students already comfortable with the C++ language will have an opportunity to master essential intermediate programming techniques using C++.

Class inheritance, templates, elementary data structures and the Standard Template Library are among the many topics that will be covered in depth. Successful completion of CS 2B is required to continue with CS 2C, which is the study of algorithmic analysis and data structures, the centerpiece of all C++-based CS degree programs and vocations.

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 which make use of inheritance, i.e., the "is a" relationship, common to all OOP languages. Specifically, the student will define base and derived classes and use common techniques such as method chaining in his or her programs..
2. A successful student will be able to use the C++ environment to define the basic abstract data types (stacks, queues, lists) and iterators of those types to effectively manipulate the data in his or her program.

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. Hybrid students have mandatory f2f classes on campus at 6pm on Tue/Thu.

Start your adventure in Foothill's class by completing the following tasks (the first two are **required**).

1. posting an introductory note about yourself in Canvas (You can simply reply to my own introductory post if you prefer)
2. introducing yourself in the subreddit under the Genius Flair
3. 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 (Wednesday for summer) of the quarter** to prevent being dropped.

Course Workbook

A selection of topics from the recommended text is the basis for your homework problems. These shape the second 9 quests (about 100 pages) of the official Genius workbook - The Enquestopedia.

The above link is just for a high-level reference of what is to come. You will get Foothill-tailored versions of the quest specifications (slightly different from the bootcamp workbook) as and when you complete each quest.

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 fun programming quests[2] (homework), exams, and conceptual understanding/ability to help other programmers (forum).

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* stands for *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 pup*	10%	
BLUE dawg (**strictly** >= 190 blue trophies)	4%	
GREEN pup	35%	
GREEN dawg (**strictly** >= 241 green trophies)	5%	
Midterm and Final	25%	
Weekly participation and final report[3]	20%	
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[4] in sequence before their freeze dates. However, once pupped in time, they can be dawged any time before the Friday of the final week.

Dawg points are **All OR NOTHING**. That means, "240 of 241 GREEN trophies" is not worth any points until one more trophy is won. No exceptions.

---

[2] More on quests in Appendix A.
[3] Assume about 1 forum discussion point on average per week and 3 points more for the final report.
[4] pup = Progress Until Password to next quest. dawg = Detailed Attentive Work Given (usually max trophies) - More on these later.

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. I also recommend you use your post to leave comments and advice for future students.

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

Extra Credit Work and Internship

Extra credit work may be offered to selected students who manage to dawg all 9 quests before the end of week 6 and maintain a good, helpful and kind profile on our subreddit.

I may offer SW Engineering Internships at Nonlinearmedia to a select few RED DAWGS with an awesome reddit profile.

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.

Some students prefer to have more structure (e.g. what to read and what to do each week). Please ask in the subreddit if that is the case, and we will suggest a few structures with more detail than the one below. One of them may hopefully work for you.

Here is one suggestion:

Week	Read References	Complete	Notes
1	Pointers and Memory	ALL BLUE QUESTS	BLUE 9 is like GREEN 1
	Recursion	Mystery Quest 1	
3	Bitwise Operators	Mystery Quest 2	Q1 and Q2 Freeze (**Mon** AM)
	Trees, Deep copies	Mystery Quest 3	
5	Exceptions & Operator overloading	Mystery Quest 4	Q3 and Q4 Freeze (**Mon** AM)
	Review/Midterm	Mystery Quest 5	
7	Inheritance and Chaining	Mystery Quest 6	Q5 and Q6 Freeze (**Mon** AM)
	Polymorphism & Virtual Fx	Mystery Quest 7	
9	Streams and Templates	Mystery Quest 8	Q7 and Q8 Freeze (**Mon** AM)
	The STL	Mystery Quest 9	
11	Sorting		Q9 Freezes (**Mon** AM)
	Review/Final Exam		

You only get 1 week per quest (half a week in summer)

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 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.

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.

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 (otherwise it will count as 0).

Special awards

This quarter, you can solve ANY number of quests at any time. The speed is only limited by your ability and availability to solve them. However, you must participate in the forum in step with others who may be on track and offer helpful advice (without revealing answers) through the entire quarter or your participation score will be affected.

Here are some exciting features:

1. The first student to dawg ALL 9 GREEN quests (>= 241 green trophies) gets 2 extra-credit points to top up their final exam score.
2. The second student to dawg ALL 9 GREEN quests gets 1 extra credit point to top up their final exam score.

Both the above awards are contingent on consistent and regular forum participation over the whole quarter.

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 is fully online you will be required to attend one or more weekly catchup group meetings over zoom to discuss where you are at and perhaps help others who may be stuck.

These meetings will take the place of classroom attendance for grade purposes. Therefore it is important to attend and contribute at these meetings if you seek a final score > 90% (required for A and above).

Times for these meetings will be arranged during the first few days of class. If you can't make the time agreed upon by the majority of your classmates you must organize another time with at least two other students who have a similar conflict (I'll set up additional recurring zooms)

Check our youtube channel for examples from the past (both lectures and catchup meetings). They are recorded and posted there.

I am not planning to attend all of the catchup meetings, but I will review the meeting recordings (and follow forum discussions closely) to get a better sense of where each of you is at. I will attend the first meeting.

The meetings are TREMENDOUS collaborative opportunities and have provided endless hours of support, both technical and emotional. However if you attend these meetings only to get unblocked when you're stuck, you'll probably not score high on participation (15%). Use the meetings to generate interesting ideas and insights which you can then post about in the forum (these kinds of posts get most points).

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 - The Genius Bootcamp

The Genius Bootcamp is a supportive and fun community of CS and C++ learners from all over the world. You are auto-enrolled in this bootcamp and you can proceed to complete it for free after you finish CS2A. In fact, you can even get credit for the GREEN quests if you decide to enroll officially in my CS2B next quarter.

At this bootcamp, you are bound to have company from Genius questers who are new to the language like you. You may find that your posts in the subreddit elicit help or responses from bootcampers not in your class.

Please introduce yourself in the Genius Flair of the red subreddit. After that, your interactions will mainly take place in the blue subreddit, r/cs2b.

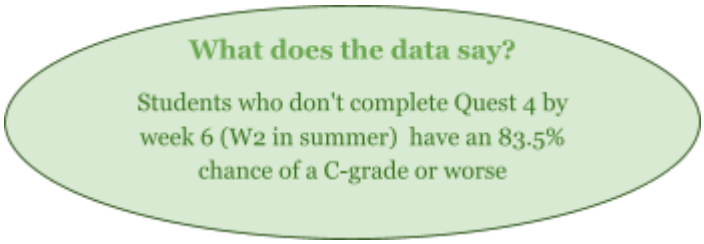
Mystery Quests (capped at **241** for CS2B this quarter)

You will solve these at the public questing site (https://quests.nonlinearmedia.org). If you're new to questing and haven't completed the BLUE level yet, type in the password "*A Tiger named Fangs*". You should hear a whooshing sound. Hit return to enter the first BLUE quest. The spec of each quest (detailed description of what you need to do) will be shown when you click on the name of the quest at the top.

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 GREEN 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. Resist your temptation to hack more, and keep moving when you get a password. P*upping*[5] a quest is sufficient to move to the next quest. You can always come back to polish your previous quests when you have free time before the final week. You get about one week on average before each quest freezes (only 3 days in Summer)

At the end of the quarter, your GREEN trophy count will be capped at **241** 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.



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.

[5] Progress Until Password

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. If you Dawg all the quests you should get at least 241 trophies (the cap).

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 (not even if you dawg it later).

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***.

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 some 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?

No. 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).

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 before you post your insight post.

Final report

A good final report has links and comments for the following types of content. See examples on our sub.

- Posts that sparked or contributed to 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 over the entire duration of the quarter, rather than in spotty bursts once in a while
- Posts that leave sound advice and encouragement for future students

The final report accounts for 20% of your grade. It must occur throughout the quarter, and not be in bursts or bunched up over a short period.

Avoid posting a list or table of links, or listing trivial posts, comments and answers. Ideally, all your selected links had something valuable to ask or say and you have something to say about them now.

Look at good examples in the sub from previous students.

Appendix C - Resources

Text recommendation

My first resource suggestion is the book: Absolute C++ by Walter Savitch. This is the same book that was recommended for CS2A.

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)*

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 CS2B, you must expect to code at least 2 hours EACH and EVERY single day for the next 6 weeks.

Make sure your schedule allows it before you start. If you apply yourself sincerely, you will learn a REALLY USEFUL skill for a happy life and FEEL the difference at the end of this quarter.

Happy Hacking!

&