CS 2B Syllabus - Winter 2021

Intermediate Object Oriented Programming in C++

Hey there. My name is Anand. I will be your instructor for this course. Please read this syllabus carefully. You should especially read it if you have never taken a class from me before.

Jump To

Course Description

What you're signing up for

Operational details

Assessment

Weekly Time Estimate

Learning Resources

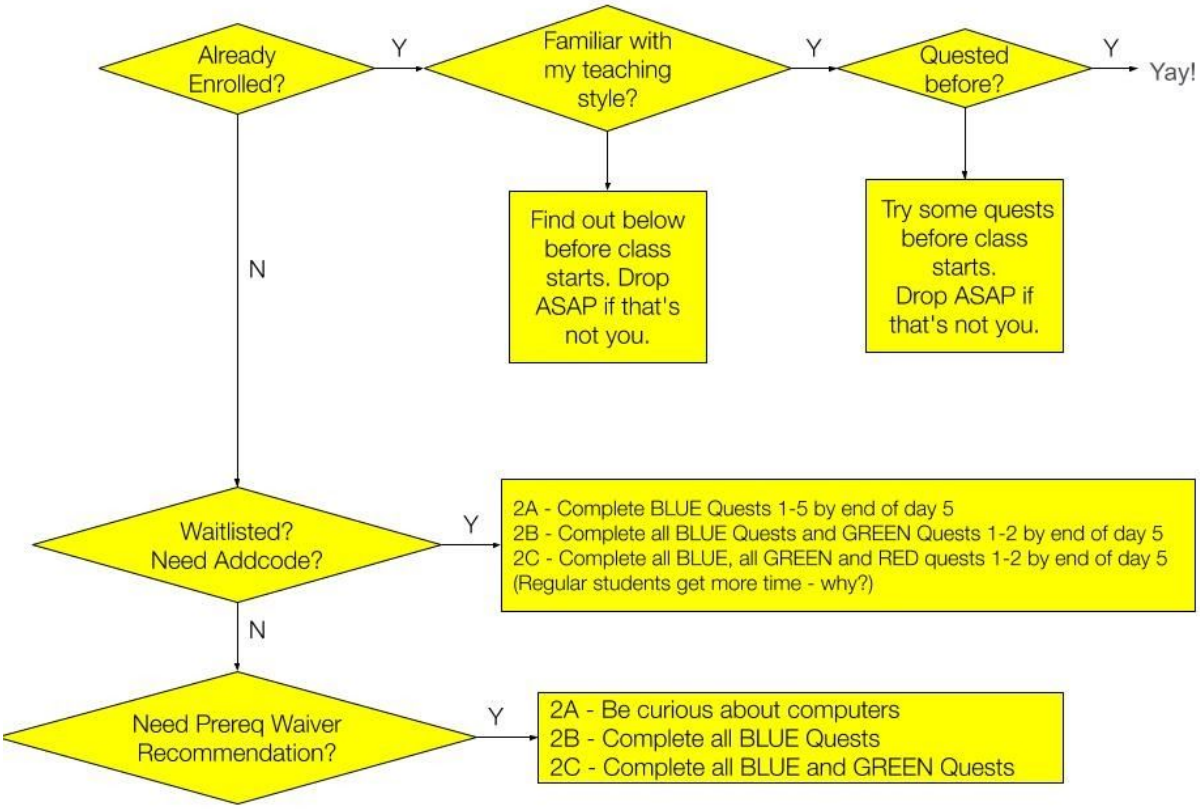
Communication

STEM Center

Disability Resource Center

Important Dates

First things first



Already Enrolled?

Y

Familiar with my teaching style?

Y

Quested before?

Y

Yay!

Find out below before class starts. Drop ASAP if that's not you.

Try some quests before class starts. Drop ASAP if that's not you.

N

Waitlisted? Need Addcode?

Y

2A - Complete BLUE Quests 1-5 by end of day 5
2B - Complete all BLUE Quests and GREEN Quests 1-2 by end of day 5
2C - Complete all BLUE, all GREEN and RED quests 1-2 by end of day 5
(Regular students get more time - why?)

N

Need Prereq Waiver Recommendation?

Y

2A - Be curious about computers
2B - Complete all BLUE Quests
2C - Complete all BLUE and GREEN Quests

My teaching style: (1) Highly hands-off (2) Will NOT debug your code for you (3) Nobody else is allowed to debug your code either (4) You will largely help each other in class (5) I will simply be an active observer in the forums to make sure you don't say anything silly or bad and find out how much value you have contributed to OTHERS at the end of the quarter. (6) Very occasionally, I may comment in the forums to answer unanswered questions or correct wrong suggestions (7) Your participation scores are worth 15%, and they are SUBJECTIVELY determined by me and you can't find out how much you will get. (8) 1-1 Office hours are ONLY for discussing personal/confidential issues. (9) Asking a private question of general value to all students will get an answer along with compensatory negative points in my private spreadsheet.

Course Description

CS 2B continues where CS 2A 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.

A working facility with simple algebra as well as good written English comprehension skills are both strong advisories.

Important note

By enrolling in this course in Winter 2021, you are implicitly agreeing that this syllabus provides a bare minimum of what you may experience during a one-quarter run of this class. However, experimental variations (e.g. zooming breakouts) may gradually be introduced on a per section basis. You agree to be part of these too and to meet reasonable (as determined by me) additional flexible learning requirements that may be incorporated into this class before the finals (totals may be scaled appropriately if you're doing it for a grade).

What you're signing up for

Here is the way I approach teaching: I don't think I have ever been fond of stuffing knowledge down the throats of people who don't want it. So you shouldn't be in my CS **2B** class if you don't want CS at this time in your life. But how do you know if you want CS or not?

Try the BLUE level quests starting from a tiger named Fangs. If you manage to make it through to the end, you'll likely survive CS2B. Else drop the course, do the BLUE quests in your own time without stress, and come back when you're ready.

If you're just doing the quests for your own fun at your own pace - Awesome. That's the way to be. No fun being stressed-ful when you're questful.

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)

If you don't have these skills yet, take some time to learn them first (usually, quite easy) and then enroll.

Operational details

Canvas is our hub to coordinate some activities and take online exams. Most of the rest of our work will happen at other online locations, including youtube, zoom, quests, reddit, etc. You will start your adventure in Foothill's class by posting an introductory note (required) about yourself in Canvas. You can simply reply to my own introductory post if you prefer.

Other than that, we will use publicly available resources with discretion, courtesy and efficiency to share subject-related information and help each other.

Don't say anything that you'll end up regretting. OTOH do try to let your natural genuine curiosity shine through. 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 tips, tricks and observations. They usually have more lasting value.

- Participating in the course discussion forums earns participation points (max 15).
- Not participating does not earn any points.
- Participating negatively by souring up someone else's experience earns negative participation points (no min).

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 (scaled to 60%), participation (scaled to 15%) and exams (scaled to 25%). I will 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

The assessment has been designed to test both conceptual understanding and knowledge of practical issues. The quests emphasize the latter and the exams/quizzes emphasize the former. The idea is that you should be able to get a passing grade by doing well in the quests and moderately well in everything else, but in order to get into A-grade territory, you have to demonstrate a solid grasp of the concepts and good class citizenship[1]. An A+ is possible if you truly enjoy programming, program in your spare time for fun, and take the trouble to independently look up, discuss (in the forums) and learn topics I will announce from time to time in announcements.

[1] "How does good class citizenship contribute to learning?" you ask. Good question. I'm using it as a suitably weighted proxy for confidence in a person's conceptual knowledge. In the past, I noticed a good correlation between a person's understanding of a concept and their willingness to explain it to someone else.

With that said, if you're focused solely on your grade and do everything flawlessly by the book, but fail to demonstrate good conceptual understanding, you will likely not get an A in this course.

In this course there will be:

- 9 Mystery Quests you will solve at the average rate of about ONE per week (your own pace). These quests need to be solved using C++ (worth 60%)
- 1 midterm and 1 final exam (worth 25% together)
- Online participation (worth 15%)

MHM Contest - New This Quarter[2]

I've decided to make this class more *worth-your-while*. You can actually make money in this course.

Yes! This quarter, I will award $250 of my own moolah to the most helpful member (mhm). How can you get it?

- In Week 11, you can nominate ONE fellow subredditor (may even be a past student) who is not already an mhm winner (email me their name and confirm you got their permission).
- In Week 12, as you prepare for your finals, you get to cast your vote on the nominations, which I'll assemble into a poll on the sub (or Canvas).
- At the end of week 12, the polls will close and the MHM wins $250 USD from me (electronic transfer)

Obviously, you **can** nominate yourself (and should consider it). You can also nominate someone else or nobody. But you need to get that person's permission first and find out if they have already won.

Also, it should be possible for me to paypal the money or something easier. I'm not gonna bank-transfer or anything (zelle is ok). And def not to places that don't accept US currency.

One more thing - Altho everybody know, like, I be the most helpful dude on the planet, you canna nominate me.

Tip - Don't nominate someone immediately after getting past a Ganon-level bug because their reddit post saved your ass. Wait until Week 11 when you can reflect on the quarter more peacefully. You can only nominate someone once. But you can thank them lots of times.

Mystery Quests

You will solve these at our public questing site (https://quests.nonlinearmedia.org). Each quest will give you a certain number of trophies. You can check your total trophy count at any time by visiting our /q site.

The password for the first quest can be found by solving all the BLUE quests (doesn't count towards your grade). The BLUE quests button down your CS2A knowledge to better prepare you for CS2B. Start at the tiger. If you haven't discovered the password by the official start date, send me an email with ONLY the following words:

"Hey &, Could I please have the password for the first GREEN quest?"

[2] Very important note: Foothill college is not liable for anything I promise you.

DON'T try and explain. If you do offer an explanation, I will have to understand it first, and then ask a number of personal questions to get to know your life situation in detail so I can put your explanation in context, which may delay your password unnecessarily.

Such students have tended to ask slower-lane questions in the middle lane where many questers are zipping by at high speed. To avoid this and keep your learning experience fun, I'd like to know who didn't complete the BLUE quests so I can keep a closer eye on their activities and progress.

👉 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, in a past quarter, 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.

At the end of the quarter, your total trophy count will be capped at 210 and scaled from 210 to 60%. You can win AT LEAST that many trophies if you make it through to the last one. 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.

Exams

You will have one midterm exam on the Thu of Week 6 (Feb 11) and one final exam on the Thu of Week 6 (Mar 25). The midterm is worth 20 points and the final 40. Together, their combined score will be scaled from 60 to 20%.

These 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.

All exams are open-book and can be taken anywhere you get a decent Internet connection. I don't recommend taking it on mobile devices.

I'm not going to be able to prevent determined cheaters from cheating. But keep in mind that cheaters only cheat themselves. Copying is a waste of your time. Few good software companies employ programmers based upon their qualifications if their demonstrated competence doesn't measure up to their stated expertise.

Besides, you'll find that copying robs you of a great opportunity to really learn the language and having a load of fun.

Participation

This is worth a whopping 15%. To put it in perspective, imagine that you were taking this course for a grade.

If you do everything else flawlessly, except participating in the online class, then you can get a maximum of 85 points. It translates to a grade of B+.

To make it into A territory, you not only have to be good at what you do, but must be able to explain concepts to others in your own words. The participation score is a confidential number I keep in my own spreadsheet by continuously monitoring the discussion forums and estimating how helpful, informative and/or encouraging each participant is. If you don't show up here, you are not deemed a participant.

You can use a thumb rule and give yourself 1 participation point for every helpful post you make in our sub. Add 1 more if the person for whom your post is meant follows up thanking you for a good tip. Subtract 1 for each unhelpful or mean post and a further 0.5 for each post that got deleted by me. If you're over 16 (I mean participation points), you will likely try to avoid the first negative one - it will reset your total to 15. This means everybody who earns a reputation as a helpful dude gets exactly one freebie mild invective, which they'd be wise not to use.

College Recommendations?

Many students who complete my CS2B or CS2C successfully ask me to write college recommendations for them. I don't write or make comparative recommendations for students, nor provide my opinion or evaluation of your current or future abilities. I do not share the grade you earned in my classes.

However, I can help you help yourself by giving you the chance to point the admissions officers at your work (e.g. your reddit posts).

Weekly Time Estimate

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. Here is one suggestion:

Week	Topic	Complete	Notes
1	Pointers and Memory	Mystery Quest 1	Review from 2a
	Recursion	Mystery Quest 2	
3	Bitwise Operators	Mystery Quest 3	
	Trees, Deep copies	Mystery Quest 4	
5	Exceptions & Operator overloading		Quests 1-4 Freeze
	Review/Midterm	Midterm Exam	
7	Inheritance and Chaining	Mystery Quest 5	
	Polymorphism & Virtual Fx	Mystery Quest 6	
9	Streams and Templates	Mystery Quest 7	
	The STL	Mystery Quest 8	
11	Sorting	Mystery Quest 9	Quests 5-9 Freeze
	Review/Final Exam	Final Exam	

Every week, give yourself 2-4 topics to study and one or more programming quests to complete. If you have some programming experience already, expect to spend about 8-12 hours per week reading and/or attending lectures or watching videos. Budget an additional 20-30 hours for working on programming quests. To be on the safe side, budget about **25** hours per week (initially) for this course.

Not ready for the commitment? Better drop early than late, while students are still waiting to get in.

See good coders help each other on our sub. This should give you an idea of just how much work is needed.

Preparatory Tasks

You must complete the first required task for this course by midnight of the first day of the quarter. This is just a simple 3-question quiz that **does not require prior knowledge of C++.** If you don't complete this task, you will be dropped and your seat likely given to a student on the waitlist. Consider this the equivalent of showing up to the first lecture. Not doing it will be treated as a no-show to the first lecture.

Also, if you think you may be dropping this course, I urge you to drop ASAP so I can give your seat to someone else on the waiting list.

Learning Resources

Rather than prescribe any particular resource,

- I'll give you a list of topics we'll cover each week.
- I'll suggest two *entirely optional* resources for you to learn about these topics. In the Canvas discussion forums, you can feel free to ask where to find certain topics. You can even share your own pointers to illuminating posts, articles, websites or videos with your classmates.

None of the suggested resources is a required purchase. The first resource is a set of Canvas Modules that ex-prof Michael Loceff created when he taught this course. Thanks to Michael, I'm able to make his entire set of modules available for reference absolutely free.

Many of you who are used to his style of teaching and his labs may recognize the spirit in them living on in our quests.

You can find Michael's materials when you scroll towards the bottom of the list of modules in Canvas. Much of it is still relevant enough to link into this course. This should be your first point of reference for most topics. But be aware of salient differences between the content of his modules 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 right.

When in doubt, try it out.

If you still don't get it. Then hit our subreddit.

My second resource suggestion is the book: *Absolute* ***C++*** (any edition at least as recent as the 2nd), by Walter Savitch, Addison Wesley. You may already have this from your CS2A.

You can order these through our bookstore at http://books.foothill.edu/, phone: (650) 949-7305.

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 contains announcements of internships, scholarships, free software offers, public lectures, etc.

Canvas

This quarter, we will be using Canvas ONLY for the following:

- Reading announcements, reference material (modules) and the syllabus
- Introducing yourself with your reddit handle (your only required post in Canvas)
- Taking quizzes and exams
- Reviewing quest/test scores when they are ready (will be announced)
- Getting the password for your first quest
- Accessing virtual learning resources such as the STEM center, zoom, online tutorial rooms, etc.

Make sure your Canvas config settings are such that you get notified when there is a new announcement.

Discussion Forums

This quarter, we'll continue to use our GREEN subreddit for all quest related discussions. Please note the following important information:

1. DO NOT SHARE personally identifying information of any kind. However,
2. Your avatar name should start with your first name (on Canvas) and an underscore, followed by your initial (or full last name) + some optional digits
3. No matter what your avatar's name, you must sign your posts with your first name (I strongly discourage unsigned posts. A reply should be able to start with something like "Hi John"
4. You should never post your student ID (CWID) online. A lot of personal information about you can be unlocked by someone who has it.
5. If you have something negative to say about someone's post in the forums, you should direct your concern to me, not to the person in the forum.

6. KEEP IN MIND that these discussion posts will persist into the next quarter and later for future students. So everything helpful you say will help far more students than just your current classmates.
7. Use Canvas for anything not quest-related (enrollment, exams, modules, etc.)
8. No posting source code and fishing for answers. Debug your own code.

Keep this in mind: 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.

Getting started on your Mystery Quests

The password to the first quest can be EARNED by solving ALL of the BLUE quests. If you are proficient in CS2A concepts, this should be straightforward. No? Better start already at the tiger, or wait until class begins officially.

Passwords for subsequent quests will be automatically revealed upon *satisfactory progress* (as the machine sees it) in each preceding quest.

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.

Bugs in your code?

Getting your code debugged by someone else is NOT allowed. That includes me, tutors, teachers, friends 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, a friend or relative and ask them what the issue is. What you can do is:

1. Check our subreddit to see if others have had similar issues
2. Explain (in our subreddit) what you're trying to do
3. Describe in English the detailed steps you would need to undertake (pseudocode)
4. Describe the behavior of your program and ask why it diverges from (2) if it does

Sometimes, a tutor, a fellow student or I may get curious about your code and want to see it. Under these exceptional circumstances, you can share your code on request.

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?

Extensions

Extensions don't make sense because the quests are self-paced. You just have to complete each by their "freeze" dates to get credit. After their freeze dates, you can 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, please don't ask for extensions.

Programming style

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.

Compilers

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).

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.

I'm generally online and able to chat in real time M-Th 10am to 11am.[3] At other times, 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, you will learn how to decode that into decimal.

One-on-one meetings are only for discussing confidential stuff. You cannot privately ask me for an explanation that is bound to be generally useful. And you cannot show your quest code to anyone (including me).

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:

[3] Times are in decimal, not binary. I kid you not, but someone did stop by my office at 2am once.

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.

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, 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 it to me and discuss accommodations.

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 summer quarter, see the official college page here.

Happy Hacking!

&