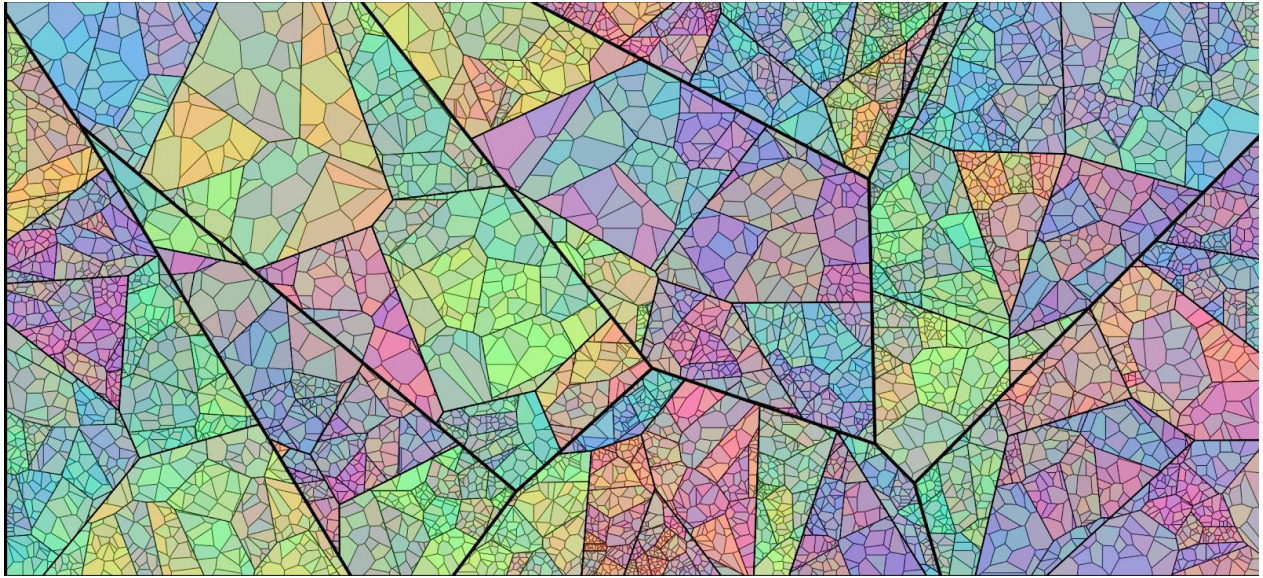


# CS 2C Syllabus - Spring 2019

## *Intermediate Object Oriented Programming in C++*

Hello. My name is Anand Venkataraman. I will be your instructor for this course. In this document I have laid out the various policies for our class. Please read it carefully and let me know if you have any questions.



## Contents

[Course Description](#)

[Weekly Time Estimate](#)

[Preparatory Tasks](#)

[Text and References](#)

[Assessment](#)

[Communication](#)

[STEM Success Center](#)

[Other Resources](#)

[Course outline and SLOs](#)

[Disability Resource Center](#)

[Important Dates](#)

## Course Description

CS 2C continues where CS 2B left off. Students already comfortable with the implementation and use of simple linked data structures in C++ will have an opportunity to master essential advanced data structures and algorithms using C++.

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.

Successful completion of CS 2B is a prerequisite for CS 2C. A working facility with simple algebra as well as good written English comprehension skills are both strong advisories.

## Weekly Time Estimate

Many students find the material challenging if they don't allocate enough time for classwork and homework. Even if you are already facile with C++, you may find yourself needing significant time to work through assignments and implement them according to spec. Many past students who underestimated the amount of work this involved ended up with grades that they were not satisfied with.

Make sure you're not one of those students by making the right choice up front. Are you able to allocate enough time for this course or would you rather take it in a later quarter when it better fits your schedule?

Every week, you'll have one or more Zybook chapters and/or Canvas Modules to study and one or more lab assignments to complete.



If you have some programming experience already, expect to spend about 8-12 hours per week reading and/or attending lectures. You will need to spend an additional 10-15 hours working on programming labs. To be on the safe side, budget about 25 hours per week (initially) for this course.

## Preparatory Tasks

You must complete the first required task for this course no later than 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.

## Text and References

All the required reference material will be provided via Canvas. In addition, for those who like hands-on interactive participation activities, I have adopted a Zybook (Algorithms and Data Structures).

Note that the Zybook is completely optional. You don't need it to complete this course. It is only for those who prefer a different style of reading. Unfortunately, the Zybook is not free. If you're

interested in subscribing to its content, see below:



The Zybook we'll use is CS 02C: *Advanced Data Structures & Algorithms in C++*.

### To get started with your Zybook:

1. Sign in or create an account at [learn.zybooks.com](http://learn.zybooks.com)
2. Enter the following zyBook code:  
**FOOTHILLCS02CVenkataramanSpring2019**
3. Subscribe

A subscription costs \$77. Students may begin subscribing the week before the quarter starts and the cutoff to subscribe is the day before the quarter ends. Subscriptions will last until the end

of the quarter. I'm told they can be extended for a nominal fee.

In addition to provided Canvas Modules and the Zybook, there is a **highly** recommended text for the course. It is *Data Structures and Algorithm Analysis in C++, any Edition  $\geq$  2nd*, by Mark Allen Weiss, Pearson. I like this book so much that I highly recommend you buy a copy for your personal library. It is bound to be useful to you long after you complete this course. You can order it through our bookstore at <http://books.foothill.edu/>, phone: (650) 949-7305.

Another recommended reference that may help with style is *The Elements of C++ Style*, by Misfeldt, et al, Cambridge University Press.

### Typical Routine

One assignment is due each week, most weeks. We'll try and stick as closely as possible to the activity schedule in the green table below. But minor deviations are possible to adapt for specific requirements of your section of the course.

Week	Canvas Module	Zybook Chapters	Topic	Lab Due	What
1	1	1	Algorithms and ADTs review	Sun, 14 Apr	Subset sum
2	2	2	Algorithm analysis	Sun, 21 Apr	Sparse matrices
3	3	2	Time complexity and Big-O	Sun, 28 Apr	Matrix products
4	4	6	General trees (and BSTs)	Sun, 5 May	Lazy deletion (BST)
5	5	7.1 - 7.4	AVL Balancing and Splaying	Sun, 12 May	Top-down splaying
6	6	All above	Review/Midterm		
7	7	5	Hash tables Quadratic probing	Sun, 26 May	Quadratic probing
8	8	3	Sorting	Sun, 2 Jun	Shellsort gaps
9	9	8	Priority Queues, Heaps, Heapsort	Sun, 9 Jun	Quicksort limits
10	A	9	Dijkstra's and Kruskal's algorithms		
11	B	9	A maxflow algorithm	Sun, 23 Jun	Maxflow
12	-	All above	Final Exam	Thu, 27 Jun	All extra credit due

Assignments are worth varying amounts of points depending on their difficulty. At the end, the grand total for all labs will be scaled to 70%.

Special access codes for each assignment will be made available via Canvas.



## Assessment

Your final grade will be based on programming lab assignments (Total scaled from max to 70%), participation (scaled to 5%) and exams (scaled to 25%).

The assessment has been designed to test both conceptual understanding and knowledge of language features. The labs emphasize the latter and the exams emphasize the former. The idea is that you should be able to get a passing grade by doing well in the labs, but in order to get into A-grade territory, you have to demonstrate a solid grasp of the concepts. An A+ is possible if you truly enjoy programming, program in your

spare time for fun, and make the effort to independently look up, discuss (in the forums) and learn topics I will announce from time to time in announcements.

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, although you may get a decent passing grade. Keep this in mind as you decide whether this quarter is the right time for you to be doing CS 2C.

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

## Tests (25%)

There is a midterm exam during the Thursday of the middle week (**Thu May 16**), and there is a final exam during the Thursday of the last week (**Thu Jun 27**). Both tests will be administered via Canvas and can be taken remotely. Further details about the tests will be given at the appropriate time.

Attempting the tests implies an acceptance of the [Foothill Honor Code](#) which means you agree not to cheat on it.

Labs, exams and quizzes submitted after their deadlines without prior arrangements are for personal edification only. Their scores will NOT contribute to your final grade.

## Participation (5% of final grade)

F2F students: You should be actively engaged in the classroom in order to receive participation

points. I will calculate it based on your attendance and class activities.

Online students: Participation comes from an activity measurement formula used by Canvas. Meaningful contribution in Canvas is at least one non-frivolous post in the discussion forum, or a serious answer to a question posted by a fellow student or me.

## Programming Labs (70%)

Submissions must be turned in via <http://cs.psme.foothill.edu/c> using a special code that is given out for each lab. Note that the specs for these labs may be unavailable until the week before the lab is due. You may submit your labs any number of times, but I suggest you don't use the site as your debugger. Rather, make sure you have a clean and debugged version of your code (to the best of your ability) ready before you submit into the site for my testing. I don't keep records or copies of your code unless you specifically request it by including your Student ID in comments at the top of your source file.

Even in that case, I will only keep your most recent submission.

I encourage you to additionally visit me with your code to get my feedback. There is an objective rubric in determining your lab score. But I can offer subjective feedback on your programming style, which, of course, you are free to ignore with no impact to your score for the lab.

If you don't know how to use a debugger, I'd say that's the first skill you should master. When you ask me or a tutor for help with your code, we'll ask you what you've already done yourself using the debugger. You should be prepared to answer questions that demonstrate a good grasp of what you're supposed to do.

If you have a bug in your code, I will NOT DEBUG IT FOR YOU, even if that means you won't ace the lab. Instead I may give you some hints. Debugging own code is an essential skill that aspiring programmers must learn and enjoy - Yes, enjoy!

Now would be a good time to read and understand [Foothill's Academic Integrity Policy](#). I strongly recommend you do so now. We take plagiarism very seriously and you don't want a permanent record on your transcript.



## Extra Credit

From time to time, I'll present challenge coding problems in class and online which you can answer to get extra credit. I'll announce the precise way in which each such challenge should be attempted and submitted at the

appropriate times. I'll make a note of your extra-credit work and apply EC points at the end of the term after the final before the grades are finalized.

## 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

You are free to use the IDE of your choice to implement your programs required for this course. However, note that I may not be able to provide support for any IDE other than Xcode, Visual Studio and Linux/g++. Some tutors in our STEM center may be able to help. But again, there is no guarantee that they'd be able to support any environment other than the three mentioned above.

I recommend using Microsoft Visual Studio/C++ (Community Edition) for Windows users or Xcode for Mac users. Our F2F students will find that our classroom is equipped with one computer per student and these computers (PCs) have an IDE installed on them. If you're in my F2F section, you're welcome to bring your personal laptops to class and use them. I can help you with any of the above IDEs. In addition, we may also have access to a Unix/Linux command line environment where you can compile and run your programs using g++. Again, this is an environment where I will be able to help you.

## Communication

Please use public discussion forums in Canvas for any question or comment that relates to the class (except questions of a private nature). If you have a confidential question (grades or registration) use the Email feature of Canvas by first clicking on Inbox (in your left-nav), and then

selecting me as the recipient of your message. You will be able to attach files to your message.

You'll find it rewarding to engage in dialogue with your classmates in the Discussion Forums, which you can reach by clicking on the Discussions link in your Canvas page for this course. I'll check the discussion forums often and answer any important open questions that no one has yet answered.

I also encourage you to meet with each other after class, 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.



Feel free to answer your fellow student questions, even if you only have a guess as to what the answer is. It's great to engage in conversation with each other in this manner.

## Contacting me

You can email me at [anand@fhda.edu](mailto:anand@fhda.edu). If you are on campus and would like to see me personally during my office hour, my room number is 0x1021 (in hexadecimal). If you're enrolled in 2C, you probably already know how to decode that into decimal.

My office hours are:

- Mon/Wed: 10-11am (online)
- Tue/Thu: 10-11am (in my office)
- Other times by appointment

This information is also posted on my departmental page and you are welcome to stop

by to ask me questions during that time. My preferred way of being contacted at other times is via the private messaging feature in Canvas.

## STEM Success Center

If the online forums here are not enough, please visit the STEM Center page, and try to make time to meet with a tutor. These people are qualified to help you with assignments or modules without giving you an answer that will short-circuit your learning process.

The STEM Center (Room 4213) is also a place on main campus where students without their own computers can do their lab work. The schedule for the STEM Center and its tutors is posted on the main STEM Center web page. Please enquire about online computer science tutors.

## Other Resources

Professor Elaine Haight maintains [a blog called Opportunities for CS students](#). It contains announcements of internships, scholarships, free software offers, pertinent public lectures, etc. Bookmark it and visit it frequently.

## 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.

## 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 meet appointment booking deadlines at the Testing Center:

- Exams must be booked at least three (3) business days/weekdays in advance of the instructor approved exam date/time.
- Finals exams must be scheduled seven (7) business days/weekdays in advance

of the instructor approved exam date and time.

Failure to meet appointment booking deadlines will result in the forfeiture of testing accommodations and you will be required to take your exam in class.

Contact the DRC if you 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 Disability Resource Center (DRC) as early as possible in the quarter. To contact DRC, you may:

Visit DRC in Building 5400, Student Resource Center:

- On the web: <http://www.foothill.edu/drc/>
- Email DRC at [drc@foothill.edu](mailto:drc@foothill.edu)
- Call DRC at 650-949-7017 to make an appointment

If you already have an accommodation notification from DRC, please contact me privately to discuss your needs.

## Important Dates

For a list of important dates for the winter quarter, see [the official college page here](#).

Refer to the green table in page 3 of this syllabus for dates regarding lab submissions, the midterm and final exam.

Happy Hacking!

& (anand)