# Course Syllabus 1 - Information and Policies

This is the first part of your two-part syllabus. The second part has all the due dates for the course.

## S.1 Course Description and Prerequisite

**CS 2C** is the study of advanced data structures and algorithmic analysis required of all C++-based CS degree programs and vocations. You will get extensive training and practice in the theory, language and coding techniques known to all professional computer scientists. You will learn how to use the many C++ STL class templates (like ***vectors***, ***sets*** and ***lists***) to implement higher level ***abstract data types*** (like ***trees, hash tables*** and ***graphs***). You will also become versed in the implementation of various sorting and searching algorithms, and be able to analyze a problem from many new points of view so you can choose the best solution for the problem at hand.

**CS 2B** (or equivalent) is a prerequisite.

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

## S.2 Instructor

I am Jim Lai, and you can email me at laijim@fhda.edu. Typically you would ask questions through Canvas Discussions tool or Canvas private message tool, and ONLY use this email if you have trouble logging in.

## S.3 Text and References

The text for the course is ***Data Structures and Algorithm Analysis in C++, any Edition*** (2nd or later), by Mark Allen Weiss, Pearson.

You can order this through the Foothill Bookstore at http://books.foothill.edu/, phone: (650) 949-7305.

## S.4 Compilers

In this class, typical compilers will be **Microsoft Visual Studio/C++** for **Windows** users, **Xcode** for **Mac** users, or **Eclipse for C/C++ Developers**.

If you are facile on another Integrated Development Environment (IDE), you are welcome to use that, instead.

## S.5 Communication

### Public Forums

Questions and comments should be posted to the **Discussions Tool (DT)** which you can reach by clicking on **Discussions** on the left menu.   I will usually reply within a few hours. Unless a question is of a private nature (i.e. grades, registration issues), please use the public **Discussions**.   Also, 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.

Steps needed to post your public questions and comments for this course can be found on the Canvas Discussion Instructions Page.

**First Week Required, Afterwards Recommended and Strongly Encouraged**

No points are awarded for contributions, and there are ***no weekly requirements***, but it's good collegial form to participate, inquire and assist. ***Also, you must post an introduction*** in the first week of class or you will be ***dropped as a "no show"*** according to the college requirements.

**Do Not Post Homework Code**

Whether you have a question or suggested answer, ***never post your entire exact homework code*** to forums. Create a separate small program to display your issue or illustration.

## Private Messages

Please use *public* **DT** for any question or comment that involves understanding the modules, tests or assignments.  If you have a confidential question (grades or registration) use the **Message Tool (MT)** by first clicking on **Inbox** at the far left, then selecting this course and your intended recipient (usually me.)

Steps needed to post your private questions and comments for this course can be found on the Canvas Inbox Instructions Page.

## Posting Program Code

You can post code to the public discussions that is not directly from your assignment. If you have an assignment question, translate that into a piece of code that does not reveal your answer or submission, exactly.

When posting code fragments (i.e., portions of your program) into questions, make sure these code fragments are perfectly indented and that they are properly formatted.  For details see the required resource module Pasting Code into Questions.

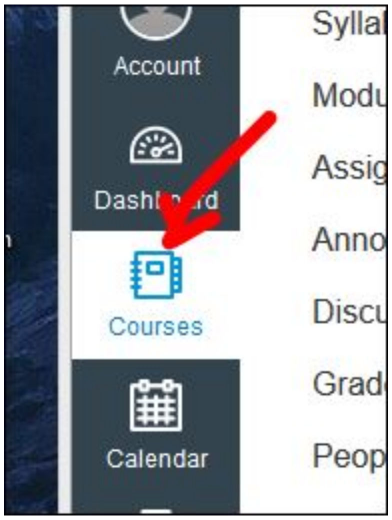
Do not post *entire programs* and ask "what's wrong?" or "is this good?" That's frivolous and indicates you have not tried to narrow down the problem. Find exactly what you want to know about and post only that part of the code.

### STEM Success Center

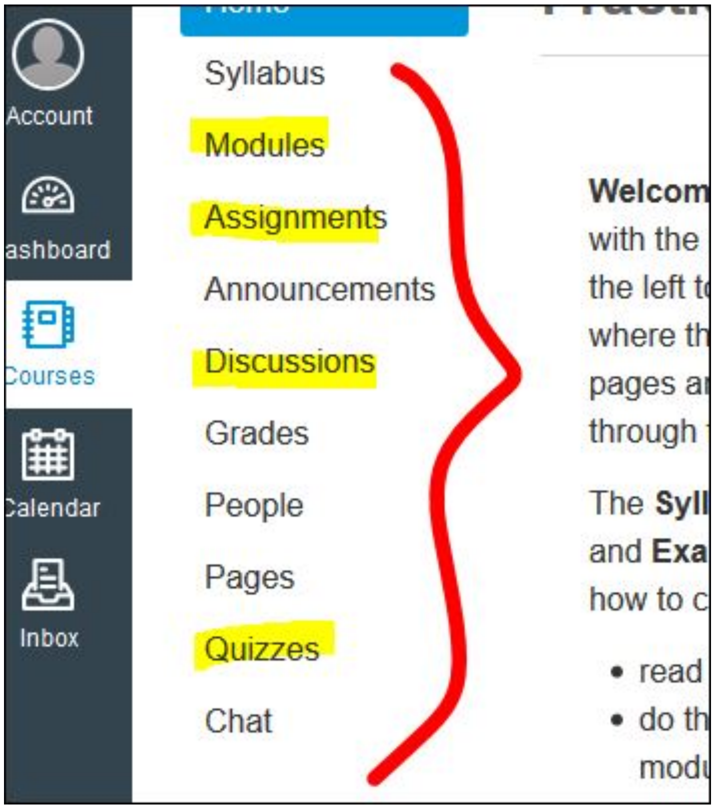
If the online forums here are not enough, please visit the STEM Success Center page and click **Schedule and Available Instructors**. These experts are qualified to help you with assignments or modules without giving you an answer that will short-circuit your discovery process. Let them know that you are not to receive actual assignment solution code or even fragments. They probably know this already, but it's your responsibility to avoid submitting something that was written by a tutor or another person.

## S.6 Where Everything Happens

Access the various areas of your course by first selecting this course through the ***Canvas*** **Courses** choice in the ***far left*** ...



... then examining our ***course choices menu***, also on the left side of the screen, but slightly to the right of **Courses**:



- ***Assignments:*** submitted through the **Assignments Tool (AST)**.
- ***Tests:*** taken through the **Quizzes Tool (QT)**.
- ***Questions or comments:*** posted using the **Discussions Tool (DT)**.
- ***Other areas***: You'll find the names self-explanatory, and you can investigate them on your own.

## S.7 Grades

Your grades are based on programming **lab assignments** (180 points = 75%) and **exams** (20 + 40 = 60 points = 25%).

**Absolute Grading Scale**

| % needed for | this grade |
|---|---|
| 97 | A+ |
| 91 | A |
| 88 | A- |
| 86 | B+ |
| 80 | B |
| 78 | B- |
| 75 | C+ |
| 67 | C |
| 60 | D |
| < 60 | F |

## S.8 Drops and Withdrawal

For a complete reference of all withdrawal dates and deadlines refer to the Foothill College registration page at the college web site here:

https://foothill.edu/calendar/winter2019.html

To stay enrolled in this class, you must participate regularly in your lab assignments and exams. This is part of the class participation that online classes must possess in order to maintain their transferability and accreditation.

### You will be dropped by me for any of the following:

- Missing a scheduled test without prior notice will result in an automatic drop.
- If you do not login for **nine (9)** consecutive days, I will drop you. (See exception below.)

- If you receive a zero on any two lab assignments, I will drop you. (See exception below.)
- If you do not post an introduction in the first week, you will be dropped for non-participation.

### Exception to Above Policies:

If the non-participation that has just been described occurs partially beyond the last date to drop, I may not be able to drop you, and you may receive whatever grade that your points dictate. Therefore, don't assume that you can simply stop participating late in the quarter and you will be dropped. If you intend to drop please do so yourself, so you don't accidentally end up with an unintended "F."

If you decide to drop the class, please let me know. I cannot allow anyone who has dropped to continue to have access to the material.

## S.9 Collaboration

Any variation of collaborating or copying programming lab assignments is prohibited. The assignment must be 100% your own work. Changing a few variables around to make them look different won't fool me. And if it does fool me, you probably had to change so many things that you knew enough to do it yourself in the first place.

You can talk about the modules all day long off-line if you wish. This rule only applies to lab assignments. There is a place to ask for help with homework: the **Public Discussions** labeled for that purpose or the ***STEM Success Center***. I will spend hours helping you each week, both individually, and in groups. You can even answer each other's questions in the **Public Discussions**. If I think you are giving too much information away, I'll edit your post. So there is no reason to ask your fiancée or your cousin's neighbor's lead guitarist.

If you accept help from someone who is not trained to teach without giving away the answer, it will short-circuit your learning process -- you will actually become weaker. Now, you don't have to agree with me - but you do have to follow the rule. If you stay in *this* class, you are agreeing to do the lab assignments on your own or with help from us, here, in this course's public forum.

For those of you wishing to give help, please do not give away the answer directly.  Either tell the person where they can look to find the solution, give them a general idea or ask them to ask me.  Don't post actual assignment code.

## S.10 How to Ask a Question

It is easy to make sure your question is a good one:  Make it specific.  An example of a bad question is, *"My program doesn't work.  Here it is.  Would you please see if you can tell me what I am doing wrong?  Gretel"*  Gretel is *lazy*.  An example of a good question is, *"My program doesn't work.  Through trial and error, I have determined that the problem lies in the following five lines, but I can't seem to narrow it down any further.  Can you help?  Hansel."*  Hansel made an attempt to organize and isolate the problem prior to asking for help.  When he gets my answer, he is sure to remember it because he is prepared to hear exactly what he needs to know.

Another example:  BAD: *"I don't understand the assignment.  I'm lost.  Please help.  Jack."*  The reason this is a bad question is that there are a million things I might say to get Jack on the right track, but I can't know which ones to focus on because I don't know where Jack's misunderstanding lies.  Jack hasn't given me any help to help him.   GOOD: I *understand the homework description up until you say 'XYZ'.  But I'm not sure what you mean by 'XYZ'.   In the lectures 'XYZ' seems to be ... but here it seems to mean something different.  From that point on, things get hazy because of this mismatch.  Would you resolve this apparent difference for me?  Jill."*  Here, Jill has told me exactly the first point at which she is confused so I know what to tell her to set her straight.

I am not discouraging questions:  I want you to ask.  Through them, I get a chance to communicate with you.  But narrow down the question.  Show me you have tried to answer it and have made some progress.  Show me exactly where you seem to be faltering so I can know how to help you.  The same holds true if you are posing your question to a fellow student or to the whole class.

## S.11 To Obtain Disability-Related Accommodations ...

... please contact ***Disability Resource Center*** **(DRC)** at the start of the quarter.   To contact **DRC**, you may:

- Visit **DRC** in Room 5400
- Email **DRC** at adaptivelearningdrc@foothill.edu
- Call **DRC** at 650-949-7017 to make an appointment

## S.12 Expanded Content

The course will provide full theoretical explanation, code examples and student programming assignments for each of the following topics: ***STL STRUCTURES***: vectors, lists, maps, sets, stacks, queues, and user-designed alternatives to all of these templates. ***TIME COMPLEXITY***: big-oh and theta algorithm times. ***TREES***: User-defined trees, binary search trees, AVL trees, rotation and top-down splaying. ***HASHING***: hash tables, hash functions, linear and quadratic probing. ***PRIORITY QUEUES***: binary heap implementation of priority queues. ***SORTING***: insertion sort, Shellsort, in-place heapsort, mergesort, quicksort, indirect sort and critical analysis of each sorting technique. ***GRAPH THEORY***: graph data structures, shortest path algorithms (dijkstra), minimum spanning trees (kruskal) and maximum flow problems.

- **Week 1** - Introduction to the world of advanced computer science. Review of ***vectors*** and ***templates***. Introduction to time complexity.  Comparative analysis of ***vectors*** and ***arrays***. The "subset sum problem." The **iTunesEntry data set**.
- **Week 2** - In-depth analysis and implementation of ***vectors***, ***lists*** and ***iterators***.  ***Stacks*** and s***parse matrices***.
- **Week 3** - Time complexity, big-oh, little-oh, omega, and theta analysis of algorithms.  Linear, logarithmic, quadratic and other growth rates.  ***Binary searching***, proper and improper uses of ***recursion*** and the **RECONS data set** of stars near Earth.

- **Week 4** - General trees and binary search trees. Full and lazy deletion. Analysis of tree algorithms. The **Project Gutenberg data set**.
- **Week 5** - Tree balancing, ***AVL trees***, rotation and ***top-down splaying***. Inheritance applied to class templates.
- **Week 6** - ***Hash tables***, hashing functions, open addressing, ***linear*** and ***quadratic probing***, the use of prime numbers in hashing functions.
- **Week 7** - ***Priority queues*** and ***binary heaps***. Complete tree implementation of binary heaps. Percolate up and percolate down. ***Heap sort***.
- **Week 8** - ***Insertion sort***, ***Shellsort***, ***in-place heap sort***, ***merge sort*** and time analyses of these sorting algorithms.
- **Week 9** - ***Quick sort*** and ***indirect sorting***. Review of **STL maps, sets, queues and stacks**.
- **Week 10** - Introduction to ***graph theory***. Vertices, edges, adjacency lists, paths and graph data structures. ***Shortest path*** algorithms and the ***dijkstra*** technique.
- **Week 11** - ***Minimum spanning trees*** and the ***kruskal*** technique. The ***maximum flow problem***.
- **Week 12** - Final exam and closing remarks.

### Student learning outcomes:

- The successful student will be able to write and incorporate balanced trees, hash tables, directed graphs and priority queues in his or her software.
- The successful student will be able to analyze the time complexity of a variety of algorithms and data structure access techniques and choose the best algorithm and/or data structure for the project at hand.

## S.13 Official Course Calendar and Due Dates

Next, proceed to the second syllabus part:

# Course Syllabus 2 - Activity and Due Dates

This is the second of your two-page syllabus. The first page has the general policies and rules for the course.

## S.14 Weekly Activities

Every week you have two lessons, or **Modules**, to study and one **Lab Assignment** to turn in. There are exceptions (see calendar, below), but this is the basic drill. This course is a lot of fun, and a lot of work. To pass it you have to make time to do both of these activities.

### Weekly Time Estimate

- **Module Reading - about six hours**. This includes pasting code into your compiler and trying it out.
- **Lab Assignment - about 10 hours**. This varies greatly with individuals. Some students take five hours, some take 20 hours.

### Typical Week

Here is the day-by-day breakdown of a typical week. Some weeks differ, but this will help you understand approximately what you are facing on a daily basis.

| **Typical Week** | |
|---|---|
| Monday (first 2 or 3 weeks only) | Read resource module R |

| | |
|---|---|
| Tuesday | Read module A |
| Wednesday | Assignment due (2 PM) |
| Friday | Read module B |

## S.15 Other Activities: *Discussions*, *Announcements*, *Tests*

### Discussions

You can ask me or other students questions in the **Discussion** area.  I hope you will be active in this area.  Read through the recent **Discussions** posts every time you log in to make sure you gain the benefit of other students' questions.

**Weekly Posts Recommended (Not Required)**

Other than the ***first week's introduction***, you are not required to post every week. However, if you are having difficulty, you should reach out and ask questions.

**No Exact Homework Code Allowed**

Please phrase questions in plain English or use non-homework code examples to demonstrate your question or suggested answer when posting.

**Follow Module 3R When Posting**

***Code fragments*** must be formatted according **Module 3R** to receive an answer. Otherwise, we'll ask you to fix the formatting and we'll check back to answer the question once the formatting is achieved.

You must also ***post an introduction*** in the first week to avoid being dropped as a no-show.

## Announcements

You will see an **Announcement** area in the ***Canvas*** course tools menu on the left. Check that area every time you login for late-breaking news.

## Tests

There is a midterm exam on *Friday* of the sixth week, and there is a Final Exam on *Tuesday* of the 12th week.  These tests will be available for exactly 18 hours starting 6 AM on the due date and be due by midnight.  You must take the tests in that 18 hour period.  I will not accept late midterms or final exams. You are to take the midterm in a single one-hour sitting and the final in a single two-hour sitting. Details about whether or not the test will automatically submit and lock-you-out an hour (or two) after you begin it will be disclosed in the announcement area prior to the exam date.

## S.16 Official Calendar

**Official Due Dates for Course**

| Date: | Day | Read Module | Lab Assignment Due 2PM | Take Quiz/Test |
|---|---|---|---|---|
| Jan 7 | Monday | Syllabus & Resource 1R | | |
| Jan 8 | Tuesday | Week 1A | | |
| Jan 11 | Friday | Week 1B | | |

| | | | |
|---|---|---|---|
| Jan 14 | Monday | Resource 2R | |
| Jan 15 | Tuesday | Week 2A | |
| Jan 16 | Wednesday | | Assignment 1 |
| Jan 18 | Friday | Week 2B | |
| Jan 21 | Monday | Resource 3R | |
| Jan 22 | Tuesday | Week 3A | |
| Jan 23 | Wednesday | | Assignment 2 |
| Jan 25 | Friday | Week 3B | |
| Jan 29 | Tuesday | Week 4A | |
| Jan 30 | Wednesday | | Assignment 3 |
| Feb 1 | Friday | Week 4B | |
| Feb 5 | Tuesday | Week 5A | |
| Feb 6 | Wednesday | | Assignment 4 |
| Feb 8 | Friday | Week 5B | |
| Feb 12 | Tuesday | Week 6A | |
| Feb 13 | Wednesday | | Assignment 5 |

|  |  |  |  |  |
|---|---|---|---|---|
| Feb 15 | Friday | Week 6B |  | Midterm Exam |
| Feb 19 | Tuesday | Week 7A |  |  |
| Feb 20 | Wednesday |  | Assignment 6 |  |
| Feb 22 | Friday | Week 7B |  |  |
| Feb 26 | Tuesday | Week 8A |  |  |
| Feb 27 | Wednesday |  | (lab due Friday) |  |
| Mar 1 | Friday | Week 8B | Assignment 7 |  |
| Mar 5 | Tuesday | Week 9A |  |  |
| Mar 6 | Wednesday |  | (lab due next Monday) |  |
| Mar 8 | Friday | Week 9B |  |  |
| Mar 11 | **Monday** |  | Assignment 8 |  |
| Mar 12 | Tuesday | Week 10A |  |  |
| Mar 14 | Thursday | Week 10B |  |  |
| Mar 15 | **Friday** | **Skip Ahead to 11B.1** |  |  |
| Mar 19 | Tuesday | Week 11A |  |  |

| | | | | |
|---|---|---|---|---|
| Mar 20 | Wednesday | | (lab due Friday) | |
| Mar 22 | **Friday** | Week 11B | Assignment 9 | |
| Mar 26 | Tuesday | | (No assignments accepted after noon on this date.) | Final Exam |

**Repeat**

No late assignments accepted after **Mar 26, NOON**. Also, the Final Exam is not accepted late. It is due by midnight, Tuesday, **Mar 26**. You have three months to prepare for these deadlines.

## S.17 Next Steps

Now that you have the idea, you can look up and see that in the first week you are supposed to read:

- Monday - This syllabus and resource R1
- Tuesday - Week 1A
- Friday - Week 1B