CS 1C Course Syllabus

General Information

This is an online course, Advanced Data Structure and Algorithmic Analysis in Java, to be offered at Foothill College for the Spring quarter of 2019.

Course Description and Prerequisite

This online course, **CS F001C 03W,** is based on the teaching materials originally created by Professor Michael Loceff at Foothill College. I, Jim Lai, the instructor for the Spring quarter of 2019, will follow these amazing materials with you and help you get through this exciting journey of the advanced data structure and algorithmic analysis in Java. During the course 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 **java.util** collection classes (like ***ArrayLists***, ***HashSets*** and ***LinkedLists***) to implement higher level ***abstract data types*** (like ***trees, hash tables*** and ***graphs***).

By successfully completing this course, you will 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.

Prerequisite: **CS 1B** (or equivalent).

Strong advisories: A working facility with simple algebra as well as good written English comprehension skills.

Instructor and Contact Information

My name is Jim Lai, and I have been teaching Computer Science courses at Foothill College for about 3 years. Before that I had worked in several high-tech companies as a principal software engineer for many years. I have my Ph.D. degree in Electrical and Computer Engineering. You can email me at laijim@fhda.edu only if necessary such as you have trouble logging in to Canvas. Typically, you would ask questions through Canvas discussions tool or Canvas private message tool.
My office hour is on every Wednesday from 4pm to 5 pm online through Canvas Chat tool. I am very interested to meet you there online.

Text and References

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

You may order this book through the Foothill Bookstore or search online and find an electronic version.

Compilers

You will need a (free) software package called a compiler. In this class we will be using a product that runs on both **Windows PCs** and **Macs** called **Eclipse**.

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

Communication

Public Forums

Questions and comments should be posted to the **Discussions** forums which you can reach by clicking on **Discussions** on the left menu. I will usually reply within 24 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 public 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.

STEM Success Center

If the online forums here are not enough, please visit the STEM center at Foothill College and experts there 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.

Course Content

The course will provide full theoretical explanation, code examples and student programming assignments for each of the following topics:***java.util STRUCTURES and INTERFACES***: ArrayLists, LinkedLists, Maps, Sets, Stacks, HashSets, TreeSets, Deques, 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 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 ***ArrayLists*** and ***Java generics***. Introduction to time complexity. Comparative analysis of ***ArrayLists*** and ***arrays***. The "subset sum problem." The **iTunesEntry data set**.

- **Week 2 -** In-depth analysis and implementation of ***ArrayLists***, ***LinkedLists*** and ***Iterators***. ***Stacks*** and ***sparse 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 **java.util Maps, Sets, HashSets, TreeSets, PriorityQueues Deques 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.

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

Course Calendar

Course Due Dates

Date:	Day	Read Module	Lab Assignment Due 2PM	Take Quiz/Test
Apr 8	Monday	Syllabus & Resource 1R		
Apr 9	Tuesday	Week 1A		
Apr 12	Friday	Week 1B		
Apr 15	Monday	Resource 2R		
Apr 16	Tuesday	Week 2A		
Apr 17	Wednesday		Assignment 1	
Apr 19	Friday	Week 2B		
Apr 22	Monday	Resource 3R		
Apr 23	Tuesday	Week 3A		
Apr 24	Wednesday		Assignment 2	
Apr 26	Friday	Week 3B		
Apr 30	Tuesday	Week 4A		
May 1	Wednesday		Assignment 3	
May 3	Friday	Week 4B		

May 7	Tuesday	Week 5A		
May 8	Wednesday		Assignment 4	
May 10	Friday	Week 5B		
May 14	Tuesday	Week 6A		
May 15	Wednesday		Assignment 5	
May 17	Friday	Week 6B		Midterm Exam
May 21	Tuesday	Week 7A		
May 22	Wednesday		Assignment 6	
May 24	Friday	Week 7B		
May 28	Tuesday	Week 8A		
May 29	Wednesday		(lab due Friday)	
May 31	Friday	Week 8B	Assignment 7	
Jun 4	Tuesday	Week 9A		
Jun 5	Wednesday		(lab due next Monday)	
Jun 7	Friday	Week 9B		
Jun 10	**Monday**		Assignment 8	

Jun 11	Tuesday	Week 10A		
Jun 13	Thursday	Week 10B		
Jun 14	**Friday**	**Skip Ahead to 11B.1**		
Jun 18	Tuesday	Week 11A		
Jun 19	Wednesday		(lab due Friday)	
Jun 21	**Friday**	Week 11B	Assignment 9	
Jun 25	Tuesday		(No assignments accepted after noon on this date.)	Final Exam

Repeat

No late assignments accepted after **Jun 25, NOON.** Also, the Final Exam is not accepted late. It is due by midnight, Tuesday, **Jun 25**. You have three months to prepare for these deadlines.

Grade Information

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

Class Policies - 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/spring2019.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."

Class Policies - 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 **Discussions**.

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 forums.

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.

Class Policies - *Discussions*, *Announcements*, *Tests*

Discussions

You can ask me or other students questions in the **Discussions** 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

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 you 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 **Announcements** 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.

Accommodations for Students with Disabilities

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

Student Learning Outcomes (SLO):

- 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.