

Advanced Data Structures & Algorithms in Java

Dept. of Computer Science, Foothill College

Course Description

In CS1C you will learn about many core data structures and be able to select which algorithm is best to use given a problem. Data structures are the plumbing and wiring of programs. Besides learning about data structures, this class will explore many algorithms. Some algorithms are used within the data structures and arise as natural support for those structures. Other algorithms are solutions to problems in their own right, and the data structures are used as a means to an end-as an expedient toward implementing the algorithm. We will use various data structures and algorithms in developing applications that will enable you and user's of your application to more easily manage and analyze real world data.

My goal in this course is to make you comfortable and excited about data structures and algorithms required in Java-based CS degree programs and vocations. You will learn the design and implementation of various data structures such as arrays, linked lists, stacks, etc and algorithms for accessing and manipulating data structures. 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.

We will use a test driven development (TDD) approach in implementing our applications. The goal of this approach is to minimize bloated code by first defining how a feature is used before we start development. Otherwise, if we dive into writing code as soon as we talk to the client (or in this class read a programming assignment description), we run the risk of spending lots of time on code and features that the user never uses, or writing buggy code that makes our application fail. Our goal is to write good object oriented application(s) that the user would want to use!

Your job is to continuously improve your implementation. So, if you find a bug in your implementation or determine that your approach resulted in an incomplete feature, then isolate the problem and work on one feature at a time until it meets all the test requirements.

Topics we will cover are:

- Abstract Data Types.
- Java generics bounded and unbounded.

- Implementation and analysis of ArrayLists and LinkedLists.
- Implementation of Iterators.
- Stack and Queue data structures based on LinkedList.
- Time complexity, big-oh, little-oh, omega, theta analysis of algorithms.
- Linear, logarithmic, quadratic and other growth rates.
- Structure of a tree.
- Binary Search Trees. Full and Lazy deletion.
- Analysis of tree algorithms.
- Tree balancing, AVL trees and rotations. Top-down splaying.
- Hash tables, Hashing Functions.
- Hashing and Collision Resolution via Open Addressing.
- Priority queues and binary heaps.
- Percolating in a heap.
- Basic sorting algorithms.
- Divide and conquer sorting algorithms.
- Time analysis of sorting algorithms.
- Introduction to graph theory.

Development tools we discuss:

- Debugging via your integrated development environment (IDE).
- Introduction to version control via git.
- Parsing a Javascript Object Notation (JSON) file.
- Working with JAR files.

Laboratory

Laboratory option of this class is conducted online. The amount of time you spend varies greatly with the individual. Some students take 10 hours, some take 20 hours.

Complete Every Single Project Every Single Week.

I believe that most of your learning will take place while you are working on your laboratory assignments. Therefore, every week you will be working on an assignment and each assignment is a computer program. It is impossible to succeed in this class without studying *every single week* of the quarter.