

CS136: Data Structures & Advanced Programming

Spring 2012 Schedule

Revised November 18, 2012 by Prof. McGuire.

Lecture: MWF 9 – 9:50 am TPL 205
Lab: Wed. 1 – 4 pm TCL 216 + 217

Exam dates, no-class days, and labs are guaranteed. Lectures and reading will adjust by a few days throughout the semester. Hyperlinks are underlined.

Chapter reading assignments are in Bailey’s free online textbook Bailey, *Java Structures*. Labs are available Tuesday afternoon and are due 10 pm the following Monday. All exams are 90-minute take-home. I do not grant extensions or accept late work.

MONDAY	WEDNESDAY	FRIDAY
<div style="border: 1px solid black; padding: 2px; display: inline-block;">Sep 3rd</div>	5th	7th 1 INTRODUCTION <ul style="list-style-type: none"> • Data Structures • Interface vs. Implementation • Programming • Honor Code
10th 2 JAVA SYNTAX <ul style="list-style-type: none"> • Declarations • Statements (control flow) • Expressions • Some keywords <u>Sum.java, CardInterface.java, Card.java</u> Reading: Ch 1	12th 3 Lab: Silver Dollar FIXED-SIZE ARRAYS <ul style="list-style-type: none"> • Pointer aliasing • Array syntax • Iteration strategies • 2D Arrays • Shuffling example <u>Card2.java, PokerHand.java</u>	14th 4 CONTRACTS <ul style="list-style-type: none"> • Interfaces • Preconditions • Postconditions • Assertions • A dynamic array interface Reading: Ch 2
17th 5 DYNAMIC ARRAYS <ul style="list-style-type: none"> • Growing an Array • Shrinking an Array • Asymptotic analysis • Amortized analysis Reading: Ch 3	19th 6 Lab: Mazes RECURSION <ul style="list-style-type: none"> • L-Systems • Iterating without loops • The Towers of Hanoi • Mathematical Induction <u>Recursion.java</u> Reading: Ch 4	21st 7 LINKED LISTS <ul style="list-style-type: none"> • Node interface • Iterative implementation • Recursive implementation • Redundant state for performance <u>List.java, ListRecursive.java, FastList.java</u> Reading: Ch 9
24th 8 RECURSION PRACTICE	26th 9 Lab: Recursion Reading: Ch 5 SORTING I <ul style="list-style-type: none"> • Comparable interface • Insertion sort Reading: Ch 6	28th 10 SORTING II <ul style="list-style-type: none"> • Stable sorting • Merge sort • Quick sort • Conclusions <u>List.java</u> (now with merge sort!) Reading: Ch 7
<div style="border: 1px solid black; padding: 2px; display: inline-block;">Oct 1st</div> 11 ORDERED STRUCTURES <ul style="list-style-type: none"> • Natural comparators • Ordered list <u>OrderedList.java</u> Reading: Ch 11	3rd 12 Lab: Sorting SEARCHING <ul style="list-style-type: none"> • Ordered vector • Linear Search • Binary Search • Stacks • Rewriting Recursion <u>OrderedVector.java, StackExample.java</u>	5th <i>Mountain Day (tentative)</i>

MONDAY		WEDNESDAY		FRIDAY	
8th		10th	13	12th	14
<i>Fall Reading Period</i>		Lab: <u>P.S. It's Just a Stack</u> LINEAR STRUCTURES I <ul style="list-style-type: none"> • Array stack • List stack • PostScript language • <i>Continued in Lab...</i> Reading: Ch 10		LINEAR STRUCTURES II <ul style="list-style-type: none"> • Infix to postfix • Queues • Circular buffers 	
15th	15	17th	16	19th	17
BACKTRACKING <ul style="list-style-type: none"> • Review Session (<u>Sample Exam</u>) • Mazes and labyrinths • Backtracking 		<i>No lab (exam)</i> SOLVING THE MAZE <u>maze.zip</u>		ITERATORS, TREES <ul style="list-style-type: none"> • Iterator • Reverse Iterator • Efficient search on a list? • Binary search tree • Other hierarchies: File system, genealogy, expressions, decision trees <u>Vector.java</u> , <u>ReverseVectorIterator.java</u> , <u>AutoReverseIterator.java</u> , <u>OrderedList.java</u> , <u>Tree.java</u> Reading: Ch 8	
22nd	18	24th	19	26th	20
BINARY TREES <ul style="list-style-type: none"> • Tree terminology • Maintaining consistency • Compute depth • Compute height • Displaying <u>BinaryTree.java</u> Reading: Ch 12		Lab: <u>The Two Towers</u> TREE TRAVERSAL <ul style="list-style-type: none"> • in-order iterator • pre-order iterator • post-order iterator • level-order iterator <u>BinaryTree.java</u> , <u>Inorder.java</u> , <u>Preorder.java</u> , <u>Postorder.java</u> , <u>Levelorder.java</u>		TREE SEARCH <ul style="list-style-type: none"> • <code>getDepth</code>, <code>getHeight</code> • <code>getDegree</code> • <code>isFull</code> • <code>contains</code>: Depth-first • Breadth-first • Naïve pathfinding: revisiting the labyrinth 	
29th	21	31st	22	Nov 2nd	23
ARTIFICIAL INTELLIGENCE <ul style="list-style-type: none"> • Tree quiz • Adversarial mazes • Payoff example: <ul style="list-style-type: none"> move backward vs. turn around • Learning 		Lab: <u>Darwin's Maze</u> SHARING INFORMATION <ul style="list-style-type: none"> • Tree quiz • <code>static</code> members • Complex static initializers • Counting instances • Communal memory 		PRIORITY QUEUES <ul style="list-style-type: none"> • Tree quiz • Motivation: <code>removeMin</code> • An inefficient priority queue • Binary heap • Vector heap • <code>percolateUp</code> and <code>add</code> Reading: Ch 13	
5th	24	7th	25	9th	26
HEAPS <ul style="list-style-type: none"> • <code>removeMin</code> • <code>pushDownRoot</code> • Skew heap • Heap sort <u>Heap.java</u>		Lab: <u>Python</u> PYTHON & BINARY SEARCH TREES <ul style="list-style-type: none"> • <code>BST.contains()</code> • <code>BST.insert()</code> • Python syntax (cont. in lab) • Operator overloading • Procedures are values <u>BST.java</u> <u>sample.py</u> Reading: Ch 14		BINARY SEARCH TREES: REMOVE <ul style="list-style-type: none"> • Python review • <code>BST.remove()</code> <u>BST.java</u> (revised)	

MONDAY	WEDNESDAY	FRIDAY
12th 27 GRAPH CONCEPTS <ul style="list-style-type: none"> • Vertices • Edges • Paths • Reachability • Flow • Cycles <i>No lab (exam)</i>	14th 28 GRAPH IMPLEMENTATIONS <ul style="list-style-type: none"> • Matrix • List • Implicit Reading: Ch 16	16th 29 STATE MACHINES
19th 30 PATH-FINDING <ul style="list-style-type: none"> • DFS • Dijkstra's Algorithm • Fibonacci Heap • A* 	21st <i>Thanksgiving Recess</i>	23rd <i>Thanksgiving Recess</i>
26th 31 <i>No class--Jury Duty</i>	28th 32 Lab: <u>Natural Selection</u> <i>(with partners!)</i> ARRAY MAP <ul style="list-style-type: none"> • Association • Array Map • Ordered Array Map <u>ArrayMap.java</u>	30th 33 HASH MAP I <ul style="list-style-type: none"> • Speculative allocation • Hash functions • Modular arithmetic • Collisions <u>SimpleHashMap.java</u>
Dec 3rd 34 HASH MAP II <ul style="list-style-type: none"> • Open addressing • Probing • Double hashing • Chaining • <i>Good</i> hash functions <u>HashMap.java</u> Reading: Ch 15	5th 35 BEYOND "ADVANCED" PROGRAMMING I <ul style="list-style-type: none"> • Review of data structures • Machine organization • Concurrent processing • Distributed computing <i>Tournament at 3pm in Lab</i>	7th 36 <i>Last Day of Classes</i> BEYOND "ADVANCED" PROGRAMMING II <ul style="list-style-type: none"> • Review of Java language features • Type inference • Dynamic types • Macros • Structures for many dimensions