

CS136: Data Structures & Advanced Programming

Fall 2011 Schedule

Revised November 28, 2011 by Prof. McGuire.

Lecture:	MWF	9 – 9:50 am	TCL 206
Lab:	Wed.	1 – 4 pm	TCL 217
Office Hrs:	Sun.	3 – 5 pm	TCL 217
	Tues.	4:10 – 5 pm	TCL 308

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 Monday or Tuesday morning 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	
Sep 5th		7th		9th	1
				<i>First day of CS136</i> INTRODUCTION	
				<ul style="list-style-type: none"> • Data Structures • Interface vs. Implementation • Programming 	
12th	2	14th	3	16th	4
JAVA SYNTAX		Lab: <u>Silver Dollar</u>		CONTRACTS	
<ul style="list-style-type: none"> • Declarations • Statements (control flow) • Expressions • Some keywords 		FIXED-SIZE ARRAYS		<ul style="list-style-type: none"> • Interfaces • Preconditions • Postconditions • Assertions • A dynamic array interface 	
<u>Sum.java</u> , <u>CardInterface.java</u> , <u>Card.java</u>		<ul style="list-style-type: none"> • Pointer aliasing • Array syntax • Iteration strategies • 2D Arrays • Shuffling example • Honor Code (in lab) 		Reading: Ch 2	
Reading: Ch 1		<u>Card2.java</u> , <u>PokerHand.java</u>			
19th	5	21st	6	23rd	7
DYNAMIC ARRAYS		Lab: <u>Mazes</u>		LINKED LISTS	
<ul style="list-style-type: none"> • Growing an Array • Shrinking an Array • Asymptotic analysis • Amortized analysis 		RECURSION		<ul style="list-style-type: none"> • Node interface • Iterative implementation 	
		<ul style="list-style-type: none"> • L-Systems • Iterating without loops • The Towers of Hanoi • Mathematical Induction 		<u>List.java</u>	
Reading: Ch 3		<u>Recursion.java</u>		Reading: Ch 9	
26th	8	28th	9	30th	10
LINKED LIST VARIATIONS		Lab: <u>Recursion</u>		SORTING II	
<ul style="list-style-type: none"> • Recursive implementation • Redundant state for performance • Insertion sort 		SORTING I		<ul style="list-style-type: none"> • Merge sort • Implementing merge sort • Quick sort • Radix sort 	
<u>ListRecursive.java</u> , <u>FastList.java</u>		<ul style="list-style-type: none"> • Comparable interface • Comparator interface • NaturalComparator • Implementing insertion sort • Stable sorting 		<u>List.java</u> (now with merge sort!)	
Reading: Ch 6		Reading: Ch 7			
Oct 3rd	11	5th	12	7th	
ORDER & SEARCHING		Lab: <u>Sorting</u>		<i>Mountain Day (tentative)</i>	
<ul style="list-style-type: none"> • Ordered list • Ordered vector • Linear Search • Binary Search 		LINEAR STRUCTURES I			
<u>OrderedList.java</u>		<ul style="list-style-type: none"> • Stacks • Rewriting Recursion • Array stack • List stack 			
<u>OrderedVector.java</u>		<u>StackExample.java</u>			
Reading: Ch 11		Reading: Ch 10			

MONDAY	WEDNESDAY	FRIDAY
10th <i>Fall Reading Period</i>	12th 13 <u>Lab: P.S. It's Just a Stack</u> LINEAR STRUCTURES II <ul style="list-style-type: none"> • Infix to postfix • PostScript language (<i>continued in lab...</i>) • Queues • Circular buffers <u>QueueDemo.java</u>	14th 14 INHERITANCE <ul style="list-style-type: none"> • Review Session (<u>Sample Exam</u>) • Inheritance • Generics
17th 15 SOLVING THE MAZE <ul style="list-style-type: none"> • Backtracking • Recursion Revisited • Enumeration Revisited <u>maze.zip</u> Optional bonus reading: Xu and Kaplan, <u>Image-Guided Maze Construction</u> SIGGRAPH 2007	19th 16 ITERATORS AND 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 <i>No lab: TAKE-HOME</i> MIDTERM	21st 17 BINARY TREES <ul style="list-style-type: none"> • Tree terminology • Maintaining consistency • Compute depth • Compute height • Displaying <u>BinaryTree.java</u> Reading: Ch 12 Midterm due: 9am
24th 18 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>	26th 19 <u>Lab: The Two Towers</u> (<i>two weeks, with partners!</i>) TREE HELPERS <ul style="list-style-type: none"> • isRoot, isLeaf • getDepth, getHeight • getDegree • size 	28th 20 TREE SEARCH <ul style="list-style-type: none"> • isFull • contains: Depth-first (Breadth-first) • isFull • CS major and course requirements <u>Tree.java</u>
31st 21 ARTIFICIAL INTELLIGENCE <ul style="list-style-type: none"> • Naïve pathfinding: revisiting the labyrinth • Adversarial mazes • Payoff example: move backward vs. turn around 	Nov 2nd 22 <u>Lab: Darwin's Maze</u> PRIORITY QUEUES <ul style="list-style-type: none"> • Motivation: removeMin • An inefficient priority queue • Binary heap • Vector heap • percolateUp and add Reading: Ch 13	4th 23 HEAPS <ul style="list-style-type: none"> • removeMin • pushDownRoot • Skew heap <u>Heap.java</u>
7th 24 SHARING INFORMATION <ul style="list-style-type: none"> • (Heapsort) • static members • Complex static initializers • Counting instances • Communal memory 	9th 25 <u>Lab: Natural Selection</u> (<i>with partners!</i>) GRAPH CONCEPTS (AI PT. 2) <ul style="list-style-type: none"> • Vertices, Edges, Paths • Reachability • Cycles • Flow • State Machines • Finite Automata 	11th 26 PATH-FINDING <ul style="list-style-type: none"> • DFS • Dijkstra's Algorithm • Fibonacci Heap • A*

MONDAY	WEDNESDAY	FRIDAY
14th 27 GRAPH IMPLEMENTATIONS <ul style="list-style-type: none"> • Matrix • List • Implicit Reading: Ch 16	16th 28 BINARY SEARCH TREES: INSERT <ul style="list-style-type: none"> • Binary search • Concealing the nodes • BST.contains() • BST.insert() <u>BST.java</u> Reading: Ch 14	18th 29 BINARY SEARCH TREES: REMOVE <ul style="list-style-type: none"> • Remove <u>BST.java</u> (revised)
21st 30 RESEARCH <i>Darwin Tournament 10pm</i>	23rd <i>Thanksgiving Recess</i>	25th <i>Thanksgiving Recess</i>
28th 31 PYTHON BASICS <ul style="list-style-type: none"> • Dynamic types • Functions • Syntax • Collections: list, dict, string, and tuple <u>Sort.java</u> , <u>sort.py</u> <i>Darwin Tournament 10pm (rescheduled)</i>	30th 32 <u>Lab: Maze in Python</u> PYTHON ELEGANCE <ul style="list-style-type: none"> • Literals • Indexing • List comprehensions • Iteration • Implementing Maze 	Dec 2nd 33 ARRAY MAP <ul style="list-style-type: none"> • Association • Array Map • Ordered Array Map <u>ArrayMap.java</u>
5th 34 HASH MAP I <ul style="list-style-type: none"> • Speculative allocation • Hash functions • Modular arithmetic • Collisions <u>SimpleHashMap.java</u>	7th 35 HASH MAP II <ul style="list-style-type: none"> • Open addressing • Probing • Double hashing • Chaining • Good hash functions <u>HashMap.java</u> Reading: Ch 15 <i>No lab</i>	9th 36 <i>Last Day of Classes</i> RESEARCH