

Computer Science 228

Introduction to Data Structures

Spring 2018

Tentative Schedule

Note: Dates are subject to change. Remember to refresh your browser to see updates to this document.

Updated on 2 January 2018

Week 1: January 8 – 12

- *Jan. 8:* Introduction: course staff; course policies; overview of the course. Algorithms and abstract data types.
- *Jan. 10:* Modularity and abstraction. Objects and classes; overview of encapsulation, inheritance, and polymorphism. Java interfaces.
- *Jan. 12:* Inheritance: interface implementation and class extension; overriding. Polymorphism: compile-time types, runtime types, and dynamic binding.

Week 2: January 15 – 19

- *Jan. 15:* University holiday; *no class*.
- *Jan. 17:* More on polymorphism and dynamic binding. Abstract classes.
- *Jan. 19:* Access modifiers. The `Object` class. Primitive types versus object types. Object equality and the `equals()` method: deep versus shallow comparison.

Week 3: January 22 – 26

- *Jan. 22:* Copying and cloning, copy constructors. Shallow copying vs. Deep copying.
- *Jan. 24:* Review of exception handling. Analysis of algorithms and big-O notation.
- *Jan. 26:* More on big-O. Array equality.

Week 4: January 29 – February 2

- *Jan. 29:* Binary search; further examples of algorithm analysis.
- *Jan. 31:* Sorting; selection sort and insertion sort; merge sort.
- *Feb. 2:* Analysis of partition and quicksort. Stability of sorting.

Week 5: February 5 – 9

- *Feb. 5:* Generic programming. The `Comparable` interface.
- *Feb. 7:* The `Comparator` interface. Wild cards.
- *Feb. 9:* Sorting with generics.

Week 6: February 12 – 16

- *Feb. 12:* More on generics: raw types and erasure. The `Java Collections` framework and `Iterator` interface.
- *Feb. 14:* Review for exam 1.
 - **Exam 1:** 6:45pm-7:45pm,
- *Feb. 16:* No class.

Week 7: February 19 – 23

- *Feb. 19:* Array implementation of collections.
- *Feb. 21:* Introduction to linked lists. Singly- and doubly-linked lists.
- *Feb. 23:* Linked list implementation of collections.

Week 8: February 26 – March 2

- *Feb. 26:* The `List` interface and the `ListIterator` interface.
- *Feb. 28:* Doubly-linked list implementation of the `List` interface.
- *Mar. 2:* Implementation of the `List` interface, iterators. Array-based implementation.

Week 9: March 5 – 9

- *Mar. 5:* Stacks and applications.
- *Mar. 7:* Postfix and infix notation. Infix-to-postfix conversion.
- *Mar. 9:* Convex hulls. Graham's scan.

Week 10: March 12 – 16

Spring break; no class.

Week 11: March 19 – 23

- *Mar. 19:* Queues. Introduction to trees. Child-sibling trees.
- *Mar. 21:* Review for exam 2
- *Mar. 22:*
 - **Exam 2:** 6:45pm-7:45pm
- *Mar. 23:* No class.

Week 12: March 26 – 30

- *Mar. 26:* Tree traversals and arity.
- *Mar. 28:* Binary trees. Expression tree.
- *Mar. 30:* Sets. Binary search trees. Implementation of BSTs

Week 13: April 2 – 6

- *Apr. 2:* Continuing BST implementation; successor.
- *Apr. 4:* Splay trees.
- *Apr. 6:* The Map interface.

Week 14: April 9 – 13

- *Apr. 9:* Hash table. Hash functions; `hashCode()`; `equals()` revisited.
- *Apr. 11:* Priority queues. Introduction to heaps — pseudocode and big-O analysis.
- *Apr. 13:* Heap implementation.

Week 15: April 16 – 20

- *Apr. 16:* Graphs.
- *Apr. 18:* Graph representations: adjacency matrix and adjacency list. Breadth-first search.
- *Apr. 20:* Depth-first search.

Week 16: April 23 – 27 (Dead week)

- *Apr. 23:* Topological sort.
- *Apr. 25:* Shortest path. Dijkstra's algorithm.
- *Apr. 27:* Review for final exam.

Week 17: April 30 – May 4

Final Exam week; no class.