

Computer Science 228

Introduction to Data Structures

Spring 2019

Tentative Schedule

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

Updated on 7 January 2019

Week 1: January 14 – 18

- *Jan. 14:* Introduction: course staff; course policies; overview of the course. Algorithms and abstract data types.
- *Jan. 16:* Modularity and abstraction. Objects and classes; overview of encapsulation, inheritance, and polymorphism. Java interfaces.
- *Jan. 18:* Inheritance: interface implementation and class extension. Abstract class. overriding.

Week 2: January 21 – 25

- *Jan. 21:* University holiday; *no class*.
- *Jan. 23:* Polymorphism: compile-time types, runtime types, and dynamic binding.
- *Jan. 25:* Access modifiers. The `Object` class. Primitive types versus object types.

Week 3: January 28 – February 1

- *Jan. 28:* Object equality and the `equals()` method: deep versus shallow comparison. Copying and cloning, copy constructors.
- *Jan. 30:* Shallow copying vs. Deep copying.
- *Feb. 1:* Review of exception handling. Analysis of algorithms.

Week 4: February 4 – 8

- *Feb. 4:* Big-O notation. Array equality.
- *Feb. 6:* Binary search; further examples of algorithm analysis.
- *Feb. 8:* Sorting; selection sort.

Week 5: February 11 – 15

- *Feb. 11:* Insertion sort; merge sort.
- *Feb. 13:* Analysis of partition and quicksort.
- *Feb. 15:* Time complexity of quicksort.

Week 6: February 18 – 22

- *Feb. 18:* Stability of sorting. Generic programming. The `Comparable` interface. The `Comparator` interface.
- *Feb. 20:* Review for exam 1.
 - **Exam 1:** 6:45pm-7:45pm,
- *Feb. 22:* No class.

Week 7: February 25 – March 1

- *Feb. 25:* Wild cards. Sorting with generics. Raw types and erasure.
- *Feb. 27:* The `Java Collections` framework and `Iterator` interface. Array implementation of collections.
- *Mar. 1:* Introduction to linked lists. Singly- and doubly-linked lists.

Week 8: March 4 – March 8

- *Mar. 4:* Linked list implementation of collections. Iterators.
- *Mar. 6:* The `List` interface and the `ListIterator` interface.
- *Mar. 8:* Doubly-linked list implementation of the `List` interface.

Week 9: March 11 – 15

- *Mar. 11:* Stacks and applications.
- *Mar. 13:* Postfix and infix notation.
- *Mar. 15:* Infix-to-postfix conversion.

Week 10: March 18 – 22

Spring break; no class.

Week 11: March 25 – 29

- *Mar. 25:* Convex hulls. Graham's scan.
- *Mar. 27:* Review for exam 2
 - **Exam 2:** 6:45pm-7:45pm
- *Mar. 29:* No class.

Week 12: April 1 – 5

- *Apr. 1:* Queues. Introduction to trees. Child-sibling trees.
- *Apr. 3:* Tree traversals and arity.
- *Apr. 5:* Binary trees. Expression tree.

Week 13: April 8 – 12

- *Apr. 8:* Sets. Binary search trees. Implementation of BSTs
- *Apr. 10:* Continuing BST implementation; successor.
- *Apr. 12* Splay trees.

Week 14: April 15 – 19

- *Apr. 15:* Hash table. Hash functions; `hashCode()`; `equals()` revisited.
- *Apr. 17:* Maps. The `Map` interface.
- *Apr. 19:* Priority queues. Introduction to heaps — pseudocode and big-O analysis.

Week 15: April 22 – 26

- *Apr. 22:* Heap construction and heapsort.
- *Apr. 24:* Graphs. Graph representations: adjacency matrix and adjacency list.
- *Apr. 26:* Graph implementation. Breadth-first search.

Week 16: April 29 – May 3 (Dead week)

- *Apr. 29:* Depth-first search. Topological sort.
- *May. 1:* Shortest path. Dijkstra's algorithm.
- *May 3:* Review for final exam.

Week 17: May 6 – May 9

Final Exam week; no class.