

# Computer Science 228

## Introduction to Data Structures

### Spring 2015

### Tentative Schedule

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

*Updated on 9 January 2015*

#### **Week 1: January 12–16**

- *Jan. 12:* Introduction: course staff; course policies; overview of the course. Algorithms and abstract data types.
- *Jan. 14:* Modularity and abstraction. Objects and classes; overview of encapsulation, inheritance, and polymorphism. Java interfaces.
- *Jan. 16:* Inheritance: interface implementation and class extension; overriding. Polymorphism: compile-time types, runtime types, and dynamic binding.

#### **Week 2: January 19–23**

- *Jan. 19:* University holiday; no class.
- *Jan. 21:* More on polymorphism and dynamic binding. Abstract classes.
- *Jan. 23:* Access modifiers. The `Object` class. Primitive types versus object types. Object equality and the `equals()` method.

#### **Week 3: January 26–30**

- *Jan. 26:* More on `equals()`: deep versus shallow comparison. Copying and cloning.
- *Jan. 28:* Shallow copying versus deep copying; copy constructors; cloning. Static variables and static methods.
- *Jan. 30:* Lifetime of variables. Review of exception handling.

## Week 4: February 2–6

- *Feb. 2:* Analysis of algorithms and big-O notation.
- *Feb. 4:* Binary search; further examples of algorithm analysis.
- *Feb. 6:* Sorting; selection sort and insertion sort; big-O analysis.

## Week 5: February 9–13

- *Feb. 9:* Mergesort. Partitioning and quicksort.
- *Feb. 11:* Implementing and analyzing quicksort.
- *Feb. 13:* The `Comparable` interface and the `Comparator` interface.

## Week 6: February 16–20

- *Feb. 16:* Sorting with generics.
- *Feb. 18:* Review for exam.
- *Feb. 19:*
  - **Exam 1:** 6:45pm-7:45pm,
- *Feb. 20:* No lecture.

## Week 7: February 23–27

- *Feb. 23:* More on generics: raw types and erasure. The `Java Collections` framework; iterators and the `Java Iterator` interface.
- *Feb. 25:* Array implementation of collections.
- *Feb. 27:* Introduction to linked lists. Linked list implementation of collections.

## Week 8: March 2–6

- *Mar. 2:* Linked list implementation of collections, continued.
- *Mar. 4:* The `List` interface and the `ListIterator` interface.
- *Mar. 6:* Linked list implementations of the `List` interface.

## **Week 9: March 9–13**

- *Mar. 9:* Stacks and queues; applications.
- *Mar. 11:* Introduction to trees.
- *Mar. 13:* Implementing binary trees.

## **Week 10: March 16–20**

*Spring break; no class.*

## **Week 11: March 23–27**

- *Mar. 23:* Introduction to binary search trees.
- *Mar. 25:* Review for exam.
- *Mar. 26:*
  - **Exam 2:** 6:45pm-7:45pm
- *Mar. 27:* No lecture.

## **Week 12: March 30–April 3**

- *Mar. 30:* Implementing binary search trees.
- *Apr. 1:* Introduction to heaps — pseudocode and big-O analysis.
- *Apr. 3:* Implementing heaps.

## **Week 13: April 6–10**

- *Apr. 6:* The Map interface.
- *Apr. 8:* Graphs: definitions; directed graphs; undirected graphs.
- *Apr. 10:* Depth-first search; breadth-first search.

## **Week 14: April 13–17**

- *Apr. 13:* Minimum spanning trees; Prim's algorithm.
- *Apr. 15:* Single-source shortest paths; Dijkstra's algorithm.
- *Apr. 17:* Examples.

## **Week 15: April 20–24**

- *Apr. 20:* Hashing: basic concepts.
- *Apr. 22:* Hash functions; `hashCode()`; `equals()` revisited.
- *Apr. 24:* Implementing chained hashing.

## **Week 16: April 27–May 1 (Dead week)**

- *Apr. 27:* Finish implementation of chained hashing.
- *Apr. 29:* Review for final exam.
- *May. 1:* Review of practice problems.

## **Week 17: May 4–8**

*Final Exam week; no class.*