# Overview

*We conclude the course by considering hash tables, a data structure that achieves constant-time performance for core symbol table operations, provided that search keys are standard data types or simply defined. Then we consider several fundamental (and useful) examples of symbol-table clients.*

**Lecture: Hash Tables.** We begin by describing the desirable properties of hash function and how to implement them in Java, including a fundamental tenet known as the *uniform hashing assumption* that underlies the potential success of a hashing application. Then, we consider two strategies for implementing hash tables—*separate chaining* and *linear probing*. Both strategies yield constant-time performance for search and insert under the uniform hashing assumption. We conclude with applications of symbol tables including sets, dictionary clients, indexing clients, and sparse vectors.

**Exercises.** (*Sorry, we are still waiting for Coursera to migrate the exercises from the old platform.*) Drill exercises on the lecture material.

**Final Exam.** (*Sorry, we are still waiting for Coursera to migrate the final exam from the old platform.*) The final exam is cumulative and designed to make sure you understand how each algorithm works and when it is effective. The final will not involve Java programming.

**Job Interview Questions.** Algorithmic interview questions based on the lecture material.

**Suggested Readings.** Section 3.4 in *Algorithms, 4th edition*.

✓ Complete