• Java prerequisites

  Topics:
  – Classes, objects, and references
  – Access modifiers
  – Arguments and parameters
  – Garbage collection

  Study questions: Appendix B

• Designing classes

  Topics:
  – Composition and inheritance
  – Access modifiers
  – static keyword
  – Overriding methods
  – Dynamic binding and method polymorphism
  – Reference type vs. object type
  – Interfaces
  – Typecasting
  – Generic interfaces, classes, and methods
  – Generic type declarations, including bounded types, type wildcards, and bounded wildcards
  – Assertions

  Study questions: Appendix C; Prelude; Java Interludes JI1, JI5

  Exercises:
  – Prelude: 1, 2
• Bag

Note: Since this was our first data structure, we covered several topics in this section even though they weren’t specific only to bags.

Topics:

– ADTs, collections, data structures and their relation to interfaces and classes
– Client vs. implementer
– Considering corner cases
– Test methods
– Bag ADT and interface
– Bag array vs. linked implementations
– Resizing arrays
– Inner classes, static and non-static

Study questions: Chapters 1–3
Exercises:

– Chapter 1: 1, 5
– Chapter 2: 1, 5, 6, 11
– Chapter 3: 1, 5, 12, 14

• Algorithm analysis

Topics:

– Asymptotic analysis
– Big-O notation
– Growth rates
– Amortized analysis
– Sum of the first $n$ integers
– Analysis of bag implementations

Study questions: Chapter 4
Exercises:

– Chapter 4: 1–6, 10–12, 17
• Stack

Topics:

– Stack interface
– Using stacks to match brackets
– Using stacks to evaluate postfix
– Using stacks to convert infix to postfix
– Array vs. linked implementations of stack (runtime, memory usage)
– Program stack / run-time stack

Study questions: Chapters 5, 6

Exercises:

– Chapter 5: 1, 3, 6–8
– Chapter 6: 1, 3, 5, 8, 9

• Recursion

Topics:

– Breaking problems into subproblems
– Requirements for recursion to work
– Activation records
– Divide & conquer vs. general recursion
– Tail recursion
– Easy vs. hard recursive algorithms to make iterative
– Overheads of recursion
– Recursive backtracking
  * General goals
  * The specific structure we used (next, extend, isFullSoln, reject)
– Analyzing recursive methods with recursion trees
– Processing arrays recursively by specifying bounds of subarray

Study questions: Chapters 9, 19

Exercises:

– Chapter 9: 1, 2, 5, 8, 12
– Chapter 19: 4, 6, 7
• Sorting

Topics:

– Simple sorts
  * Selection sort
  * Bubble sort

– Runtime analysis of sort methods

Study questions: Chapters 15

Exercises:

– Chapter 15: 1, 2, 11, 13