SIENA*college*
Computer Science

Computer Science 225
Advanced Programming
Siena College
Spring 2020

Topic Notes: Java You Know

We begin by reminding you about the Java and general programming concepts and constructs you already know.

Java/Programming Basics

By now, these topics should be second nature.

- sequential execution
- variables and primitive data types
- evaluation of arithmetic and boolean expressions
- basic `String` manipulations (*e.g.*, `length`, concatenation, `substring`, `split`, `charAt`, `toLowerCase`, `toUpperCase`, `startsWith`, `trim`)
- basic type conversions (`DecimalFormat`, `Integer.parseInt()`, `Double.parseDouble()`)
- `equals` *vs.* `==`
- conditional execution (`if`, `if .. else`)
- repetition (`while`, `for`, `do .. while`)
- methods, both writing and calling, including parameter passing and returning values
- keyboard/terminal I/O (`java.util.Scanners` on `System.in`, `System.out`)
- basic file I/O (`java.util.Scanners` on a `java.io.File`, `java.io.PrintWriters`)
- random number generation (`java.util.Random` class)
- arrays
- basics of defining classes (instance variables, constructors, methods)

Intermediate Java/Programming

Things you likedly learned late in your introductory Java course or in data structures. Maybe you missed or have forgotten a few.

- two-dimensional arrays
- Javadoc
- `static` (variables, methods)
- overriding `equals`, `toString` methods
- `switch` statements
- enhanced `for` loop
- interfaces
  - API examples: `java.util.Iterator`, `Iterable`, `Comparable`
- exceptions (`try` .. `catch`, `throw`, `throws`)
- recursion
- complexity (Big O basics)
- timing (`System.currentTimeMillis`, `System.nanoTime`)
- `StringBuffer` and `StringBuilder`
- ternary operator

Fundamental Data Structures and Related Topics

The core data structures material you know and love.

- wrapper classes, autoboxing, autounboxing
- lists
  - array-based (`java.util.ArrayList`)
  - singly-linked, doubly-linked (`java.util.LinkedList`), circular
- generic structures/type parameters
- linear structures: stacks, queues, deques
- restricted data structures and why they are beneficial
- ordered structures
  - linear vs. binary search
- maps/dictionaries

- binary trees
  - binary search trees
  - balanced BSTs (`java.util.TreeMap`)
- heaps
- priority queues (`java.util.PriorityQueue`)
- hashing

Java API

Some of these you might know, others you might not, but they do things you are well aware of.

- `String` (it probably does more than you think)
- `java.util.Arrays`
- `java.util.HashMap`