CSc 127B — Introduction to Computer Science II
Fall 2015 (McCann)

http://www.cs.arizona.edu/classes/cs127b/fall15/

# Exam #1 Topic List

**Purpose:** Students often appreciate receiving a list of topics that will be covered on upcoming exams. My usual answer to the question "Which topics should we study for the exam?" is "All of 'em!" While that's true, it's also not detailed.

Please note that this is not meant to be an exhaustive list of exam topics; rather, it's meant to hit the highlights and ensure that you don't overlook a critical topic.

1. Classes and Objects
   - Constructors, class and instance variables, constants, and methods
2. Arrays
   - Declaration, allocation and usage of 2-D arrays
3. Exceptions
   (a) Exception class hierarchy (esp. checked vs. unchecked exceptions)
   (b) Managing exceptions (propogation, `try{}` and `catch(){}`)
   (c) Throwing exceptions
4. Files
   (a) Text vs. binary files
   (b) `System.in`, `System.out`, `System.err`
   (c) The File class (is a file existing, readable, etc.)
   (d) Text file I/O: `Scanner`, `PrintWriter`, and and `BufferedReader`
   (e) Binary file I/O:
      i. Low-level: `FileInputStream`, `FileOutputStream`
      ii. Mid-Level: `DataInputStream`, `DataOutputStream`
      iii. High-Level: `ObjectInputStream`, `ObjectOutputStream`, `Serializable`
5. Class Reuse
   (a) Class composition and adaptation ("has a")
   (b) Inheritance ("is a"); Overriding vs. Overloading
   (c) UML representations of composition and inheritance
   (d) Abstract classes
   (e) Generics (parameterized types, e.g. "`List<E>`")
   (f) Interfaces:
      i. Encapsulation
      ii. Java's interfaces (e.g., `Serializable`, `Comparable`)
      iii. Creating our own

(continued . . . )

6. Lists

   (a) Abstract Data Types (ADTs) and their relationship to Java classes

   (b) List operations: create, destroy, size, capacity, isEmpty, isFull, insert, append, prepend, delete

   (c) Java's `List` interface

7. Java classes and methods to know how to use (beyond those from 127A):

| | | |
|---|---|---|
| (a) | `Exception:` | `getclass().getName()`, `getMessage()`, `printStackTrace()` |
| (b) | `File:` | `exists()`, `canRead()`, `canWrite()`, `isDirectory()` |
| (c) | `Scanner:` | `Scanner(File)` constructor |
| (d) | `PrintWriter:` | `println()` |
| (e) | `BufferedReader:` | `readLine()`, `read()` |
| (f) | `FileInputStream:` | `read()` |
| (g) | `FileOutputStream:` | `write()` |
| (h) | `DataInputStream:` | `read___()` (that is, `readInt()`, `readDouble()`, etc.) |
| (i) | `DataOutputStream:` | `write___()` (that is, `writeInt()`, `writeDouble()`, etc.) |
| (j) | `ObjectInputStream:` | `readObject()` |
| (k) | `ObjectOutputStream:` | `writeObject()` |

   We also expect that you will be able to correctly and completely construct objects from each of these classes (e.g., `new DataInputStream(new FileInputStream(new File("filename")))`)

8. Other Material

   - Don't forget to review what you learned from the sample programs, assignments, sections, and ICAs!