

CPE 203 – Midterm Study Guide

**** Disclaimer ****

This list is not necessarily exhaustive. Just because it isn't on the list doesn't mean it won't be on the midterm.

Basic Classes / Terminology

- instance variables
- class vs. object
- constructors
- declaring constants
- static
- final
- public / private / protected
- primitives - boolean, int, double, etc
- arrays (creating, iterating through)
- common classes - String, Math, ArrayList, HashMap, etc.
- mutable/immutable
- Primitive wrapper classes (including auto-boxing and auto-unboxing)

Java Syntax

- if statements
- while loops
- for loops
- for-each loops

Interacting Classes

- calling methods of another class and of your own class
 - Static methods
 - Instance methods
 - Fields / data members (static and instance)
- `main()`
- testing – JUnit, writing test cases

Exceptions

- Throwing
- Catching
- `finally` block
- Checked vs. unchecked (`RuntimeException`)

Interfaces

- creating one
- implementing one
 - With an explicit class
 - With an inner class
 - With a lambda expression
- references of the interface type
- polymorphism / implementing (“overriding”) methods
- determining if code will Always work, might Crash at Runtime, or Won’t Compile
- Using interfaces to reduce coupling between modules, particularly for utilities (sort, etc.)
 - Open/Closed principle
- `Comparator` interface

Inheritance

- Abstract Class
- Abstract Method
- Superclass / Subclass
- `final` to prevent subclassing
- Calling superclass constructor
- Calling superclass method
- Overriding Methods
- Overriding `Object.hashCode()` and `Object.equals(Object)`
 - Using as a key in a hash table
- Type casting / downcasting
- `instanceof`

Design

- Designing a simple class
- When to use instance vs. static method
- Choosing appropriate classes for methods
- Using interfaces to define an API
- Open/Closed Principle

Understand all code (and be able to read/write similar code) that you have written for Labs and Projects.

Understand all code (and be able to read/write similar code) that has been handed out and/or discussed in class, except where specifically noted (for example, you should understand the code in `week_03/3_add_triangle`, with the exception of `MyGUI.java`, because `MyGUI.java` has a comment saying to not worry about the contents of that file).