Review for Final Exam

- Final Exam is comprehensive
  - 20-40% Chapters 1-4
  - 20-40% Chapters 5-8
  - 30-40% Chapters 9-11
- 30% of course grade
- 10 bonus points
  (can add up to 3 pts to course grade average)
- Study Strategy
  - Exams 1 and 2 (including "Old Exams"), lab activities, quizzes, and projects, class notes, and examples
  - Textbook: "Key Concepts", examples in each chapter

Course Overview

1. Introduction
2. Data and Expressions
3. Using Classes and Objects
4. Writing Classes
5. Conditionals and Loops
6. More on Conditional and Loops
7. Arrays (Ch8)
8. Object-Oriented Design (Ch7)
9. Inheritance
10. Polymorphism
11. Exceptions

1. Introduction

- Objectives - when we have completed this introduction to computing, you should be able to:
  - Understand the basics of software and its relationship to hardware
  - Write simple Java programs
  - Edit, compile, and run Java programs using jGRASP
  - Set a breakpoint and step through your program in debug mode
  - Use Javadoc comments in your programs
  - Run Checkstyle to verify your comments and format
  - Generate documentation for your programs

2. Data and Expressions

- Objectives - when we have completed this set of notes, you should be familiar with:
  - character strings & escape sequences
  - variables and assignment
  - primitive data
  - if and if-else statements
  - expressions and operator precedence
  - Accepting standard input from the user
  - data conversions

3. Using Classes and Objects

- Objectives - when we have completed this set of notes, you should be familiar with:
  - object creation and reference types
  - the String class
  - packages and the import declaration
  - the Random class
  - the Math class
  - formatting output: NumberFormat and DecimalFormat
  - wrapper classes

4. Writing Classes

- Objectives - when we have completed this set of notes, you should be familiar with:
  - Anatomy of a class: state and behaviors
  - Constructors
  - UML class diagrams
  - Encapsulation
  - Anatomy of a method: Parameters, Local data
  - Constant fields (public and private)
  - Invoking methods in the same class
  - Building a class incrementally
  - Testing a class
  - Writing a driver program

5. Conditionals and Loops

- Objectives - when we have completed this set of notes, you should be familiar with:
  - flow of control: sequence, selection, iteration
  - boolean expressions
  - selection: if and if-else statements
  - iteration: while loops
  - equality, relational, and logical operators
  - block statements
  - comparing objects
  - nested while loops
  - the ArrayList class

6. More Conditionals and Loops

- Objectives - when we have completed this set of notes, you should be familiar with:
  - switch statements
  - the conditional (ternary) operator
  - do statement
  - for loop
  - for-each loop

7. Arrays (Ch8)

- Objectives - when we have completed this set of notes, you should be familiar with:
  - array declaration and use
  - bounds checking and capacity
  - arrays that store object references
  - variable length parameter lists
  - multidimensional arrays

8A. Object-Oriented Design I

- Objectives - when we have completed this set of lecture notes, you should be familiar with:
  - Software development activities
  - determining the classes and objects that are needed for a program
  - the relationships that can exist among classes
  - the static modifier

8B. Object-Oriented Design II

- Objectives - when we have completed this set of notes, you should be familiar with:
  - writing interfaces
  - using interfaces in the Java API including Comparable
  - method and constructor overloading
  - method design
  - types of testing

9. Inheritance

- Objectives - when we have completed this set of notes, you should be familiar with:
  - deriving new classes from existing classes
  - the `protected` modifier
  - creating class hierarchies
  - abstract classes
  - indirect visibility of inherited members
  - designing for inheritance

10A. Polymorphism

- Objectives - when we have completed this set of notes, you should be familiar with:
  - defining polymorphism and its benefits
  - using inheritance to create polymorphic references
  - using interfaces to create polymorphic references

10B. Comparing Objects

- Objectives - when we have completed this set of notes, you should be familiar with:
  - Defining the natural ordering of objects
  - Implementing the Comparator interface
  - The Collections class
  - Sorting a list of objects
  - Sorting an array of objects

11. Exceptions

- Objectives - when we have completed this set of notes, you should be familiar with:
  - the purpose of exceptions
  - exception messages
  - the try-catch statement
  - propagating exceptions
  - checked and unchecked exceptions
  - reading and writing text files
  - try-catch and exceptions for files
  - exception messages
  - opening files in the default web browser

11. Searching and Sorting(Ch10)

- Objectives - when we have completed this set of notes, you should be familiar with:
  - Linear Search
  - Binary Search
  - Selection Sort
  - Insertion Sort