Objectives:
1. To understand Object-Oriented Design (OOD) and its advantages.
2. To revisit the concepts of objects and classes, with emphasis on implementation.
3. To understand the responsibilities of a class.
4. To understand how to use the Classes, Responsibilities and Collaborations (CRC) card to generate a good class design.

Reading Assignment:
1. Nell/Chip/Mark, Chapter 6

Concepts:
1. The OOD Model
2. Implementing OOD
3. Inheritance
4. Data Representation
5. Packages

Announcements:
1. Midterm Test next week
2. Lab- Midterm Review Session
1. OOD Model
   - OOD advantages over traditional functional decomposition
     - OOD focuses on objects instead of tasks
     - Objects are self-contained entities composed of data and operations on that data
     - Classes define the pattern used when instantiating an object
     - OOD is better suited to solve large problems

2. Implementing OOD
   - Identify the initial set of objects in the problem
     - Nouns represent Objects
     - Verbs represent Actions (methods)
   - Filter the list to eliminate all duplication
   - Identify the responsibilities of the objects necessary to solve the problem
     - A responsibility of a class is an action the class must be able to perform
     - Types of responsibilities include:
       - Constructor
       - Copy Constructor
       - Transformer
       - Observer
       - Iterator
   - Identify the collaboration necessary between classes to solve the problem
   - Information collected can then be transferred to a CRC card, which aids in the design process
   - Now teamwork can occur with individual programmer

3. Inheritance
   - Defining a new class based on an existing class
   - Adapting the methods in the new class

4. Data Representation
   - Instance Data- internal representation of a specific object, records object’s state
   - Class Data- available to all objects of a class, defined by using static keyword
   - Local Data is specific to a given call of a method

5. Packages
   - A compilation of classes
   - Any field or method with package access can be accessed by any member of the package
Objectives:
1. To prepare students for Midterm Exam
2. To understand the material in the Midterm Study Guide Handout

Midterm Study Guide Handout

Chapter 1 main topics
- Software Life Cycle Phases
  - Design
  - Implementation
  - Debugging
  - Testing
  - Documentation
  - Maintenance
- Basic Programming Terminology
  - Object Oriented Design – Objects, Classes
  - Algorithms and Psuedocode
  - Programming Languages- Machine, Assembly, High-Level
  - Compilation, Linking, Interpretation- How does Java Work?
  - Basic Control Structures- sequence, selection, loop, subprogram
  - What are Data Types
- Java Specific Terminology
  - Java Virtual Machine
  - Java Compiler
  - Structure of a Java Program
  - Java Identifiers
  - Reserved Words

Chapter 2 main topics
- Primitive Data Types
  - Byte, char, short, int, long – Understand what each is used to store
  - Boolean
  - Floating Point Numbers – float, double
- Methods
  - Why use Methods?
  - Constructors
  - Void methods
  - Helper methods
- Variables
- Java String Class
- Java I/O Device
- Arithmetic Expression
  - Arithmetic Operators
  - Modulus
  - Rules of Precedence
  - Associatively
Chapter 3 main topics

- Java Operators
  - Increment / Decrement
  - Prefix / Postfix
  - Precedence
  - Associativity
  - Logical / Rational Operators
- Type and String Conversion
  - Type casting
  - String conversion
  - Primitive / reference types
- String Operators
  - String Methods
  - Substrings
  - Converting to Numberical Representations
  - Getting Input
  - Copying References

Chapter 4 main topics

- The boolean data type
- Logical expressions
  - Relational operators: < > <= >= == !=
  - Logical operators:
    - && (AND)
    - || (OR)
    - ! (NOT)
    - Short-circuit evaluation
- Comparing String objects
  - s1.equals(s2) — returns true or false
  - s1.compareTo(s2) — returns an int value (negative, 0, positive)
- Relational operators with floating point types
  - “x == y” vs. “Math.abs(x – y) < 0.00001”
  - “x != y” vs. “Math.abs(x – y) >= 0.00001”
- The if statement
  - if-else form
  - blocks (compound statements)
  - if form
  - Nested if statements

Chapter 5 main topics

- Converting String s to int or double:
  - int number = Integer.parseInt(s);
  - double amount = Double.parseDouble(s);
- Additional useful String operations:
  - Trim whitespace from the ends of a String object:
    - s.trim()
  - Find the position of the first occurrence of “x” in String s:
    - int pos = s.indexOf(“x”);
  - Find the position of the next occurrence of “x” in s:
    - int pos = s.indexOf(“x”, pos + 1);
  - Get the substring of s from position p1 to position p2:
    - String s2 = s.substring(p1, p2);
- Get the substring of s from position p to the end of the string:
  String s2 = s.substring(p);
- Get the character at position p in the String s:
  char ch = s.charAt(p);

- The while statement
- Count-controlled loops
- Event-controlled loops
  - Sentinel-controlled loops
  - End-of-file-controlled loops
- Nested loops

Some final suggestions:
Try the Quick Check and Exam Preparation Exercises at the ends of the chapters.

Good luck!