CPSC 1302 - Computer Science 2 – Final Exam Study Guide

Study Material

Chapters 8-15, 18 of *Big Java* by Lewis & Loftus

Material from programming assignments, Wiley Plus, class discussions, and notes
CHAPTER QUIZZES & EXERCISES!!!

Test Format (MONDAY, May 9, 2011 in CCT406)
Multiple Choice Questions (40 points) - terminology
Short Answer Questions (110 points) - evaluate, compute, write Java code, what is the output?
UML Diagram (10 points)
1 Programming Problem (40 points) that has at least one user-defined class and a main class

ACADEMIC OBJECTIVES

- Students will demonstrate the ability to read moderately complex programs written in a specific programming language and understand what these programs do
- Students will demonstrate the ability to design algorithms utilizing the principles of object-oriented programming (classes, encapsulation, inheritance mechanisms, polymorphism) to solve moderately complex problems
- Students will demonstrate the ability to write moderately complex programs in a specific programming language to implement these algorithms
- Students will demonstrate the ability to follow specified style guidelines in writing programs, and understand how the guidelines enhance readability and promote correctness in programs
- The students will demonstrate the ability to design algorithms utilizing some principles of programming (exception handling, recursive programming, and basic data structures) to solve moderately complex problems

Specifically Study

- Designing Classes
    - Discovering Classes
    - Cohesion and Coupling
    - Immutable Classes
    - Side Effects
    - Preconditions and Postconditions
    - Static Methods and Variables
    - Scope
- Interfaces and Polymorphism
    - Using Interfaces for Algorithm Reuse

    - Converting Between Class and Interface Types
    - Polymorphism via Interfaces
    - Designing for Polymorphism
    - Event Processing (Events, Event Sources, Event Listeners)
    - Inner Classes
    - Buttons
    - Mouse Events
    - Using Timer Class
- Inheritance
    - Class Hierarchies
    - Creating Subclasses
    - Overriding Methods
    - Visibility
    - Designing for Inheritance
    - Polymorphism via Inheritance
    - Late Binding
    - Object Class
- Input/Output and Exception Handling
    - Reading and Writing Text Files
    - Reading Text Input
    - Throwing Exceptions
    - Checked / Unchecked Exceptions
    - Catching Exceptions
    - The *finally* Clause
    - Designing Your Own Exception Types
- Object-Oriented Design
    - Software Life Cycle
    - Discovering Classes
    - Relationships Between Classes
    - Case Study: Printing an Invoice
    - Case Study: ATM
- Recursion
    - Examples of Recursion
    - Recursive Helper Methods
    - Efficiency of Recursion
- Sorting & Searching
    - $O(n^2)$ sorts - Bubble Sort, Selection Sort, Insertion Sort
    - O(n log n) sorts – Merge Sort
    - Analyzing Algorithms – O(log n), O(n), O(n log n), $O(n^2)$
    - Sequential search, Binary Search
    - Comparable Class (Equals, CompareTo methods)
- An Introduction to Data Structures
    - Using Linked Lists
    - Implementing Linked Lists
    - Abstract Data Types (ADTs)
    - Stacks

    - Queues
- Graphical User Interfaces
    - Text Input
    - Text Areas
    - Layout Management
    - Choices
    - Menus
    - Swing Documentation