CPSC 1302 - Computer Science 2 - Midterm Study Guide

Study Material

Chapters 8-12 of *Big Java* by Cay Horstmann

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

Test Format (March 16, 2011 in CCT406)
Multiple Choice Questions (20 points) - terminology
Short Answer Questions (40 points) - evaluate, compute, write Java code, what is the output?
UML Diagram (10 points)
1 Programming Problem (30 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

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

- o Event Processing (Events, Event Sources, Event Listeners)
  - o Inner Classes
  - o Buttons
  - o Mouse Events
  - o Using Timer Class
- Inheritance
  - o Class Hierarchies
  - o Creating Subclasses
  - o Overriding Methods
  - o Visibility
  - o Designing for Inheritance
  - o Polymorphism via Inheritance
  - o Late Binding
  - o Object Class
- Input/Output and Exception Handling
  - o Reading and Writing Text Files
  - o Reading Text Input
  - o Throwing Exceptions
  - o Checked / Unchecked Exceptions
  - o Catching Exceptions
  - o The *finally* Clause
  - o Designing Your Own Exception Types
- Object-Oriented Design
  - o Software Life Cycle
  - o Discovering Classes
  - o Relationships Between Classes
  - o Case Study: Printing an Invoice
  - o Case Study: ATM