Midterm Exam Study Guide

General Guidelines

The exam is taken from **Non-Programmer's Tutorial for Python 3** , Python worksheets, and Chapters 1-3.3 of *Big Java 4th Edition* by Horstmann. The text, worksheets, quizzes, and programming assignments should be the primary instruments of study to prepare for the exam. The PowerPoint slides in CougarView should be regarded as a companion to rather than substitute for the text. **The learning aids in each chapter and the resources in WileyPLUS are also useful in preparing for the exam.**

Material from Python programming assignments and labs
Material from Transition Guide
Material from Java programming assignments and labs

Test Format (March 2, 2012 in CCT208)
Multiple Choice Questions (20 pts)
Short Answer Questions (40 points) - evaluate, compute, write Python and Java statements, what is the output from Python and Java code segments?
Python and Java Programming Problems (40 pts)

ACADEMIC OBJECTIVES
- The students will demonstrate an understanding of computing and computer science.
- The students will demonstrate knowledge of computer programming concepts.
- The students will demonstrate knowledge of basic syntax of a specific programming language
- The students will demonstrate the ability to read moderately complex programs written in a specific programming language and understand what these programs do
- The students will demonstrate the ability to design algorithms utilizing the principles of object-oriented programming to solve moderately complex problems
- The students will demonstrate the ability to write moderately complex programs in a specific programming language to implement these algorithms

Specific Guidelines

Things you should know:

PYTHON
- ❖ identify the parts of a simple Python program (sections 1-3, pages 1-8)
- ❖ Python terminology, expressions (sections 3, pages 8-11)
- ❖ Python input, variables, and assignment statements (Section 4, pp. 12-17)
- ❖ Python while loops (Section 5, pp. 17-22)

- ❖ Python decision statements (if) (Section 6, pp. 22-28)
- ❖ Debugging Python programs (Section 7, pp. 28-32)
- ❖ Python functions (Section 8, pp. 33-39)
- ❖ Python lists (Section 10, 15, pp. 43-52, 69-73)
- ❖ Python for loops (Section 11, pp. 52-55)
- ❖ Python Boolean expressions (Section 12, pp. 55-61)
- ❖ Python strings (Section 16, pp. 73-78)
- ❖ Python files (Section 17, pp. 79-85)

JAVA

- ❖ identify the parts of a simple Java program (Topic 1.5, pages 11 - 13)
- ❖ Java comments (Topic 1.5, page 12, Special Topic 1.1, page 14)
- ❖ kinds of programming errors (Topic 1.7, pages 18 - 19)
- ❖ Rules for forming Java identifiers, determine if a programmer-defined Java identifier is valid, Java statements that declare and initialize variables (Topic 2.2, pages 36 - 38)
- ❖ What accessor and mutator methods are (Topic 2.7, page 48)
- ❖ What a reference is and what it contains when it does and does not refer to an object, What an alias is and how references can become aliases (Topic 2.10, pages 54 – 56, Topic 5.2.5, page 182)
- ❖ know how applets differ from applications (Special Topic 2.2, pages 63 - 65)
- ❖ know how RGB colors work and be able to identify primary colors based on their RGB value (Topic 2.13.4, page 67)white space characters (space, tab, newline)What access specifiers are and what they mean, what an instance variable is, how the declaration of a method is formed, what a return type is (Topics 3.1 and 3.3, pages 82 - 84, 86)
- ❖ What constructors are, what distinguishing features constructors have, how the declaration of a constructor is formed, what a default constructor is, how many constructors a class can have (Topic 3.3, page 87 and class discussion)

Plus these items discussed in class:
- ❖ how many bits are in a byte (8) and the number of values that can be represented in a byte (2^8=256)
- ❖ letter abbreviations for certain powers of 2 (K, M, G, T)
- ❖ What's special about the package (module) java.lang (it doesn't have to be imported) and example classes in that package (Math, Object, String, System, for example)
- ❖ How to recognize a method (function) call and identify the calling method (function), called method (function), and what type of value is returned by the called method (class discussion)
- ❖ Terms: formal parameter (parameter coded in a method (function) declaration (also known as a method header)), actual parameter also known as argument (coded when a method (function) is called), calling method (method containing the method call), called method (method named in the method call)