CS 210 – Algorithms and Data Structures

College of Information Technology and Engineering
Weisberg Division of Engineering and Computer Science

Semester and Year:
Spring 2009

Classroom Section, Meeting Times, and Location:
Section: 201 Days: MWF Time: 10:00-10:50 Location: GH211

Instructor:
Professor Joe Fuller
Office: 205B Gullickson Hall
Phone: 304-696-6204
Email: fullerj@marshall.edu

Class Webpage:
http://users.marshall.edu/~fullerj

Textbook:
Koffman and Wolfgang, *Objects, Abstraction, Data Structures, and Design Using Java, Version 5.0,* John Wiley & Sons, Inc., 2005 ISBN 0-471-69264-5

Approximate Schedule:

Topic	Class meetings
Program Correctness	3-5 sessions
Efficiency of Algorithms	2 sessions
Inheritance	3-5 sessions
Exam I	~ 5th week
Lists and Collections	3-6 sessions
Stacks and Queues	3-3 sessions
Recursion	2-3 sessions
Exam II	~ 10th week
Trees, Depth First search	6 sessions
Hash Tables	2-3 sessions
Sorting	3 sessions
Self-Balancing Trees	6 sessions
Final Exam	During final exam perios

Material for Midterm Exam 1

I. Program Correctness and Efficiency

Reading: The student should read the following sections of the text: 2.1-2.7

Objectives: In this section, the student will learn to answer the following questions:

a. What are bugs and defects?
b. What is the difference between a run time error and a syntax error?
c. Describe the Java Exception class.
d. What is meant by throw, catch, try, and finally?
e. Why is testing a program so important?
f. What are some of the commonly used methods of testing a program?
g. What are stubs? What are drivers?
h. Who should test a program? When should a program be tested? How should a program be tested?

Evaluation: In order to evaluate the student's success in mastering the topics listed above:

1. Any or all of the following will be part of the first midterm exam
   a. Define bugs and defects.
   b. Define run-time errors and syntax errors and explain the difference.
   c. Describe the use of try-catch-finally blocks.
   d. Describe stubs and drivers. Describe how a software system should be tested.

- The student will write Java programs that use try-catch-finally blocks.
- The student will be asked to demonstrate that he/she can use debuggers in his/her Java IDE.

II. Efficiency of Algorithms

Reading: The student should read section 2.8 of the text

Objectives: In this section, the student will learn to:

a. Define "Big-O" notation.
b. The student will be able to define O(1), O(n), O(n^2), O(log n), O(n log n), and O(n!).
c. The student will be able to determine the Big-O for selected algorithms.
d. The student will be able to explain why Big-O notation is important.

Evaluation: In order to evaluate the student's success in mastering the material listed above:

1. The student will demonstrate his/her mastery of the terms listed above on the first midterm exam.
2. The student will do lab exercises that demonstrate his/her understanding of the importance of Big O to software engineering.

III. Inheritance

Reading: The student should read sections 3.1-3.4, 3.6-3.8

Objectives: In this section the student will review the concepts of inheritance, derived classes, abstract classes, the cosmic class Object, and packages.

Evaluation: In order to evaluate the student's success in mastering the material in the above list

1. On midterm one, the student will provide definitions and discussion of the following terms:
   a. The "is a" relationship
   b. The "has" a relationship"
   c. The Object class
   d. Overriding and overloading methods
   e. Casting and cloning

2. The student will write Java program(s) that demonstrate proficiency in using inheritance, overriding, overloading, casting and cloning.

Midterm One

Midterm one will be given on the topics I, II, and III. It will be scaled to be worth 100 points. It will count 1/3 of the final test average.

Programming and Lab assignments in this section will count toward the final programming average.

Material for Midterm Exam 2

I. Lists and Collections

Reading: The student should read sections 4.1, 4.3-4.8

Objectives: Students will learn

a. The abstract list and abstract collection
b. The `List` and `Collection` interfaces
c. How to define and implement a linked list
d. How to define and implement a doubly linked list
e. The `ArrayList` class
f. How to use the Java `LinkedList` class

Evaluation: In order to evaluate the student's success in mastering the material in the above list

1. On midterm two, students will be asked questions that demonstrate their knowledge of
   a. Lists (ordered and unordered)
   b. Collections
   c. List and Collection interfaces
   d. Linked and doubly linked lists

2. Students will write a Java program that implements generic data structure.
3. Students will be asked to demonstrate their proficiency with the Java `LinkedList` class.

II. Stacks and Queues

Reading: The student should read sections 5.1, 5.2, 6.1, and 6.2

Objectives: Students will learn

a. The specifications for a queue and for a stack
b. The students will learn how to use Java Library classes for queue and stack applications
c. How to implement stacks and queues with a linked list class

Evaluation: In order to evaluate the student's success in mastering the concepts listed above

1. On midterm two, students will be asked to demonstrate their knowledge of
   a. Queues – definition and specification
   b. Stacks – definition and specification
   c. Applications appropriate for queues
   d. Applications appropriate for stacks

2. Students will write at least one Java program to implement a stack or a queue
3. Students will be asked to demonstrate proficiency with the Java Library classes that implement stacks and queues

III. Recursion

Reading: The student should read sections 7.1-7.3 carefully. Students should scan sections 7.4-7.6

Objectives: Students will learn the following:

a. What is a recursive function in mathematics? What is a recursive function in Java (C, C++, etc)
b. How to write recursive functions in Java
c. When it is it appropriate to use recursion? When is it not appropriate?
d. How to compare and contrast recursive methods with iterative methods
e. Students will see examples of recursion including calculation of n!, array searching, and the Towers of Hanoi.

Evaluation: In order to evaluate the student's success in mastering the concepts listed above

1. On midterm 2, students will be asked to demonstrate their knowledge of
   - Recursive functions and recursive function definitions
   - Advantages and disadvantages of recursion

2. Students will be asked to write at least one Java program that uses recursion to solve a problem

Midterm Two

Midterm two will be given on the topics I, II, and III above. It will be scaled to be worth 100 points. It will count 1/3 of the final test average.

Programming and Lab assignments in this section will count toward the final programming average.

Material for Final Exam

I. Trees

Reading: The student should read sections 8.1-8.4 carefully. Students should scan sections 8.5 and 8.6

Objectives: Students will learn the following:

a. Basic definitions of trees and binary trees
b. Tree traversal – in order, post order, and pre order
c. How to specify a binary tree
d. How to specify a binary search tree

Evaluation: In order to evaluate the student's success in mastering the concepts listed above

1. On the final exam, students will be asked to demonstrate their knowledge of
   a. Trees and associated definitions such as root, node, height, path, sub-tree, balanced tree, complete tree, etc.
   b. Methods of tree traversal
   c. How to insert, delete, and retrieve nodes in a binary (search) tree

2. Students will be asked to write a Java program that implements a binary search tree. The program will be written so that the tree can store data from any class.

II. Hash Tables

Reading: Students should read sections 9.1-9.3

Objectives: Students will learn the following

a. The Java Set Interface and the Java Map Interface
b. Definition of a Hash Table
c. Issues associated with a hash table including collisions, chaining, probing, and hashing functions
d. Students will be able to compare and contrast hash tables with linked lists, arrays, and binary search trees

Evaluation: In order to evaluate the student's success in mastering the concepts listed above

1. On the final exam, students will be asked to demonstrate
   a. Their knowledge of the definitions of Maps, Sets, and Hash tables
   b. How to implement a hash table
   c. Their knowledge of the characteristics of a good hash table
   d. Their knowledge of collisions in a hash table and how to deal with them

2. Students will be given several lab and homework exercises to demonstrate their grasp of the concepts from this section.

III. Sorting

Reading: Students should read sections 10.1-10.9.

Objectives: Students will learn to

a. Describe and implement selection sort, bubble sort, insertion sort, Shell sort, Merge sort, and Quick Sort
b. Students will learn the Big Oh associated with each sort
c. Students will use the sort method of the Java Arrays class
d. Students will learn to compare and contrast any two of the sorts listed in item a

Evaluation: In order to evaluate the student's success in mastering the concepts listed above

1. On the final exam, students will be asked to demonstrate
   a. Their knowledge of the each of the sorts covered in item a in the objectives. Students may be expected to use a "by hand" walk through for this purpose
   b. Their ability to compare and contrast the sorts
   c. Their knowledge of the efficiency of each sorting method

2. Students will be given several lab and homework exercises to demonstrate their grasp of the concepts from this section.

IV. Self-Balancing Trees

Reading: The student should read sections 11.1, 11.2, and 11.3

Objectives: In this section the student will learn

a. Why a balanced tree is desirable
b. What it means to do rotations about a node in a tree
c. What an AVL tree is
d. How to specify an AVL tree
e. What a Red-Black tree is
f. How to specify a Red-Black tree
g. What is a 2-3 tree and how to specify it
h. What is a B-Tree and how to specify it

Evaluation: In order to evaluate the student's success in mastering the concepts listed above

1. On midterm 3, students will be asked to demonstrate
   a. Their knowledge of why balanced trees are important
   b. Their knowledge of an AVL tree
   c. Their knowledge of algorithms to balance a tree using AVL algorithms

2. Students will be given several lab and homework exercises to demonstrate their grasp of the concepts from this section.

Final Exam

The final exam will cover topics I-IV above and topics from Exams I and II. Fifty percent of the exam will be devoted to the material covered since Exam II. The other fifty percent will be cumulative and will cover Exams I and II. It will be scaled to be worth 150 points. It will count 1/3 of the final test average. Programming and Lab assignments in this section will count toward the final programming average.

Attendance, participation, and Decorum

Students are expected to behave as if the class is a business meeting. This means

- Attend and participate in every class. After 3 unexcused absences, your grade will be decreased by one letter
- Arrive on time to class.
- Do not sleep during class. Students who sleep during class will be asked to leave the classroom.
- Leave class only when the class has been dismissed by the instructor.
- Turn cell phone and pagers off before class starts.
- Keep laptops turned off during discussions and lectures.
- Refrain from eating during class. Students may bring drinks that have proper lids.

Final Grades

Final grades will be determined by weighting the test average by 60%. Homework, quizzes, and labs will be weighted by 30%. The final 10 percent is reserved for attendance and decorum.