CS 560: Data Structures and Algorithms II
Fall 2005---Course Outlines

Instructor:

Name:	Dr. Sattiraju Prabhakar
Email:	prabhakar@cs.wichita.edu
Office:	Jabara Hall, Room 241
Telephone:	(316)978-3928
Office Hours:	MW: 4:30PM – 5:30PM, Other times by prior appointment

Graduate Teaching Assistant:

Name:	Fakolujo, Ayodele
Email:	aafakolu@cs.wichita.edu
Office:	JB 220
Telephone:	978-5273
Office Hours:	TBA

Course Documents and Submissions:

Teaching Material:	http://www.cs.wichita.edu/~prabhakar/Teaching/DSAII_F2005.htm
Availability:	Usually available a few hours before the class
Course Related Info:	http://www.cs.wichita.edu/~prabhakar/Teaching/DSAII_F2005.htm
Assignments:	http://blackboard.wichita.edu
Messages:	http://blackboard.wichita.edu

Course Instruction:

Lecture Timings:	MW 5:35PM – 6:50PM
Lecture Room:	JB 128
Tutorial Timings:	MW 7:00PM – 7:25PM

Course Material Sources:

Main Text:

Introduction to Algorithms, 2nd Edition
Thomas H. Corman, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein
The MIT Press, 2001

Prerequisites:

CS 300 – Data Structures and Algorithms I
CS 320, Math 344 or 511, and Stat 460

Course Objectives:

This course introduces to the design and analysis of algorithms, by building upon the earlier course of Data Structures and Algorithms I. Specific emphasis will be given to design, analysis, specification and implementation of algorithms, along with proof of correctness for these algorithms. The analysis focuses upon the time and space complexities of the algorithm.

Required Background:

1. Students are required to have experience at least in one of the following programming languages: {C, C++, Java}. They need to choose one of the programming languages for doing their assignments.
2. Given an algorithm, in an abstract form[1], they should be able to develop a bug free program in the programming language of their choice.
3. They should have mathematical background provided in the prerequisite courses.

Course Organization

Syllabus:

Following topics will be covered during the semester.

1. Role of Algorithms in Computing
2. Introduction to Analyzing and Designing Algorithms
3. Notations for Functions
4. Recurrences
5. Heapsort
6. Greedy Algorithms
7. Quicksort
8. Sorting in Linear Time
9. Binary Search Trees
10. Red-Black Trees
11. Elementary Graph Algorithms
12. Minimum Spanning Trees
13. Hash Tables

Class and Lab Organization:

Classes are, in general, inclusive of instruction and tutorial sessions. The lab session will be treated as an extension of the class session. The instruction covers both the theoretical and implementational aspects of the algorithms. In the tutorial sessions, the students are required to participate in solving problems. The instructor's instruction of the implementation will be in pseudo C like language.

[1] The algorithms are in a pseudo C like language.

Evaluation:

Grading:	**Relative**
1. Exams (2):	50%
2. Programming Assignments (2):	20%
3. Problem Solving Assns (4):	30%

Number	Exam	Date	Weight	Topics(Tentative)
1.	Mid Term	Oct 12, 2005	20%	***Comprehensive***
2.	Final	Dec 7, 2005	30%	***Comprehensive***

Alternate Exam Policy: Except on medical grounds with proper medical certificate, no alternate exams will be conducted.
Academic Honesty: This course adapts University policies on academic honesty.
Quizzes: These are surprise quizzes and will not be announced in advance.
Lab Quizzes: There will be surprise lab quizzes

Assignments:

1. These are take home assignments.
2. Each assignment is submitted to the blackboard electronically. At the same, time, each group makes a hard copy submission to the instructor.
3. Delayed submissions will be penalized at the rate of 10% per day.
4. Extension to delayed submissions will not be granted except for medical or emergency situations. The students need to submit the proof for each situation and take a prior permission.
5. The instructor's solutions to the assignments will be made available soon after the students submit their solutions.

Programming Assignments:

1. The purpose of the programming aspect of the assignments is to help the students to develop detailed understanding of algorithms discussed in the class.
2. You can choose one of the programming languages for your assignments: {C, C++, Java}. Please note that there is no instruction of programming language in the class.
3. Each assignment is done by a team of maximum two people. The students need to select their partners for each assignment. No additional marks given for a group with one person. Each group makes a single submission.
4. You and your partner will be using the same programming language for the assignment. No additional marks will be given if you cannot find a partner who can use the same programming language as you do.
5. No support will be provided for debugging your programs. You are expected to know how to develop programs using the programming language you selected.

6. Support will be given in terms of consultations during office hours, for those having difficulty in implementing algorithms.
7. You cannot receive 100% marks for programming assignment if it does not work. The maximum mark you can receive for a program that does not work is 75%.
8. The instructor's solutions to programming assignment will be in C language.
9. Demos: The programming assignments are assessed during their demos. The undergraduate students demo their assignments to the GTA. The Masters students demo their assignments in front of the class.
10. *Nomination of work sharing:*
    a. The students need to nominate the work shared within the group. That is, the group members need to identify the amount of work done by each member. This needs to be done by mutual consent. For example, {John FirstMember: 45%; Mary SecondPerson: 55%} is a valid nomination. The nominations should sum to 100%.
    b. The nominations are submitted along with the assignments.
    c. Each student, in a group, is expected to contribute to the assignment at all levels.
    d. The final mark on an assignment is calculated as follows. The mark received by the submission is multiplied by the contribution of a group member. The contribution of a group member, to an assignment, is computed from the percentage nominations made by the members and the assessment by the instructor during demo.

Solving Problems:

1. The students understand algorithmic techniques by applying them to various problem solving situations.
2. The problems can be on specification, analysis, design and application of algorithms.

Exams:

1. The students will be asked to solve problems. The (problem solving) assignments, and tutorial sessions are designed to help the students to prepare for the exams. In addition, students are required to prepare for the exams by reading the book, interacting in the class, and solving problems. The final exam is comprehensive and includes all the topics covered in the class.
2. There are no make up exams. Only those students with medical reason or in a crisis situation are exempt for this rule. But they need to provide evidence of their situation. If you are planning on an overseas trip, or you have not arranged a leave from your employer, no alternate exam will be provided.
3. Both exams are closed book and closed notes.

Grading:

1. Grading scheme: This is relative.
2. All assignments, and exams will be numerically marked, not graded into A .. F grades.
3. Only after all the marks are available, the final grades are computed.

4. The class average is computed. With respect to the average, different grades are given.

Tentative Schedule of Topics[2]:

Lec#	Date	Topic	Activity
1	Aug 22, 2005	Introduction to Course and Algorithms	
2	Aug 24, 2005	Insertion Sort and Analyzing Algorithms (Chapter 2)	
3	Aug 29, 2005	Insertion Sort and Analyzing Algorithms (Chapter 2)	
4	Aug 31, 2005	Growth of Functions (Chapter 3)	Assignment1 >>
5	Sep 7, 2005	Growth of Functions (Chapter 3)	
6	Sep 12, 2005	Growth of Functions (Chapter 3)	<< Assignment1
7	Sep 14, 2005	Recurrences (Chapter 4)	Assignment2 >>
8	Sep 19, 2005	Recurrences (Chapter 4)	
9	Sep 21, 2005	Heapsort (Chapter 6)	<< Assignment2
10	Sep 26, 2005	Heapsort (Chapter 6)	Prog Ass1 >>
11	Sep 28, 2005	Heapsort (Chapter 6)	
12	Oct 3, 2005	Quicksort (Chapter 7)	
13	Oct 5, 2005	Quicksort (Chapter 7)	<< Prog Ass1
14	Oct 10, 2005	Quicksort (Chapter 7)	
Exam	Oct 12, 2005		*MidTerm Exam*
Hol	Oct 16 - 18, 2005		*Spring Break*
15	Oct 19, 2005	Binary Search Trees (Chapter 12)	
16	Oct 24, 2005	Binary Search Trees (Chapter 12)	
17	Oct 26, 2005	Binary Search Trees (Chapter 12)	Assignment3 >>
18	Oct 31, 2005	Red-black Trees (Chapter 13)	
19	Nov 2, 2005	Red-black Trees (Chapter 13)	<<Assignment3
20	Nov 7, 2005	Red-black Trees (Chapter 13)	Assignment4 >>
21	Nov 14, 2005	Hash Tables (Chapter 11)	
22	Nov 16, 2005	Hash Tables (Chapter 11)	<<Assignment4 Prog Ass2>>
23	Nov 21, 2005	Hash Tables (Chapter 11)	
24	Nov 23, 2005	Elementary Graph Algorithms (Chapter 22)	
25	Nov 28, 2005	Elementary Graph Algorithms (Chapter 22)	<< Prog Ass2
26	Nov 30, 2005	Elementary Graph Algorithms (Chapter 22)	
27	Dec 05, 2005	Minimum Spanning Trees	
	Dec 07, 2005	JB 128, 5:35 – 7:25PM	*Final Exam*

[2] The schedule of classes can change. The instructor makes the decision based on the class requirements.