Fall 2010

# CS 400/600: Data Structures and Software Design

Meilin Liu
*Wright State University - Main Campus*, meilin.liu@wright.edu

# *Computer Science (CS) 400/600*
# *Data Structures and Software Design*
**Wright State University**

**Course Description**
This is a fundamental course for students majoring in Computer Science. Students will learn: basic algorithm analysis techniques; asymptotic complexity; big-O and big-Omega notations; efficient algorithms for discrete structures including lists, trees, stacks, and graphs; fundamental computing algorithms including sorting, searching, and hashing techniques.

**Goals**
There are several goals to accomplish in CS 400/600
1. Master algorithm analysis
2. Master elementary data structures: arrays, stacks, queues, linked lists
3. Master advanced data structures: heaps, trees, hash table
4. Master sorting algorithms including: insertion sort, selection sort, merge sort, bubble sort, quick sort.
5. Study searching algorithms: linear and binary search
6. Study tree traversal algorithms.
7. Study elementary graph algorithms: representations, breath-first search, depth-first search, minimum spanning tree, shortest path algorithms

**Lecturer**
Meilin Liu
Office: 353 Russ Engineering Center
Phone: 937-775-5061
Office Hours: 2:00 – 4:00 pm Tuesday /Thursday
Email: meilin.liu@wright.edu
Web: www.wright.edu/~meilin.liu

**Class**
- Tuesday /Thursday 4:10 pm- 5:25 pm 204 MC FH

**Text**
Required: *Data Structures and Algorithms in C++ : Michael T. Goodrich, Roberto Tamassia, and David M. Mount, ISBN: 0-471-20208-8, Publisher: Wiley, 2004*

Reference*: Introduction to Algorithms, by T.H. Cormen, C.E. Leiserson ,R.L. Rivest, and C. Stein, ISBN: 0262032937, Publisher: The MIT Press,2009, Third Edition.*

**Required Work**

| | | |
|---|---|---|
| Program Assignment | 30% | (Possibly three or four programming projects.) |
| Homework | 10% | |
| Quizzes | 10% | Pop up Quizzes to keep everyone up with the class readings! |
| Midterm Exam | 20% | |
| Final Exam | 30% | |

**Grading**
The base scale is: A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59. This is the highest requirement that will be used. The scales may be lowered or revised if necessary.

**Policies and Notes**

- Attendance: Attendance is not required, but may be documented by the pop up quizzes. If you are not a regular attendee, it will be your responsibility to seek out what material was covered in the lecture and learn it. Most of my exam questions will be taken directly from ideas covered during the lecture, so it greatly helps if you attend!

- I will utilize webCT (wisdom.wright.edu) to post updates to the course, assignments, solutions, announcements, and schedule, etc. Get in the habit of checking it regularly.

- Always make back ups of all of you work. Never have just one copy of anything!

- If you are going to miss an exam, for any reason, discuss it with me in advance. If it is an emergency situation, please notify me as soon as possible

- You can reach me a number of ways. Email is the best way to reach me. You can also reach me by phone during the day at 775-5601. If you need human contact either stop in during my office hours, make an appointment by email.

- There are technologies we will use in this class that you may not already know, such as working with tools in lab. We will cover some of these technologies or they will be discussed in lab. If you have trouble, please don't hesitate to come and talk with one of the teaching assistants or me.

- The key to learning in this class will be spending time working through the problems. Don't wait until 2 hours before something is due to try to learn the concept. This normally ends in a disaster! Stay up with the readings and try to work through some of the problems in the book. There will be lots of problems, so try and work through them when you get them and don't wait until the end. This is not a class where 3 hours of "cramming" right before the midterm/final will translate into a good grade!

- See the "Course Policies" handout for important information on the acceptable coding standards for the programming assignments, additional academic integrity policies, and other course policies.

**Academic Misconduct**

In this class, the only way to truly learn the concepts to is do the work yourself. I encourage working with other people on the course concepts. When you begin to write the assignment, complete and submit your own work.

Work that has obviously been copied or in the more extreme case, when the original author's name has not even been changed, both parties will receive a 0 grade for that assignment. Both parties will also be turned over to the Office of Judicial Affairs.

**Tentative Lecture Schedule:**

| Week | Topics | Reading |
|---|---|---|
| 1 | Introduction; Unix review; Array; Lists | Ch. 1, 2 |
| 2 | Algorithm Analysis; Stacks, Queues | Ch. 3,4,5 |
| 3 | Algorithm Analysis; Stacks, Queues | Ch. 3,4,5 |
| 4 | General Trees, Binary Trees, Tree Traversal | Ch. 6 |
| 5 | Search Trees | Ch. 9 |
| – | **Midterm Exam:** Tues. Oct. 12nd, 2010 *(tentative)* | – |
| 6 | Hash Table | Ch. 8 |
| 7 | Graphs: Definitions, Implementation, Traversal | Ch. 12.1-12.5 |
| 8 | Graph Applications: Spanning Tree, Shortest Path | Ch. 12.6-12.7 |
| 9 | Sorting: internal and external | Ch. 10 |
| 10 | Text processing and Tries | Ch. 11.1, 11.3 |
| – | **Final Exam:** Tuesday, November 16th, 2010, 5:45 – 7:45 pm | – |

Always have readings scheduled for that day complete prior to the class meeting