

# CSE 331: Algorithms and Data Structures Spring 2008

*Version 1: 5 Jan 2008*

## Website

[www.cse.msu.edu/~cse331](http://www.cse.msu.edu/~cse331) will contain general course information and information specific to lecture Section 2.

## Instructors:

The instructors are working together so that students in both sections have a similar course experience. Similar lecture slides and homeworks are planned. However, some variation in the sections is possible due to student questions, instructor examples, etc.

### Section 1:

T R 10:20 - 11:40 EB 1234 George Stockman (stockman@cse.msu.edu)  
office hours: Mon & Thur 2-4 PM in EB 3115E

### Section 2:

T R 3:00 - 4:20 EB 1234 John Weng (weng@cse.msu.edu)

## Teaching Assistants

to be determined

## Course Description

In this course, students will survey fundamental data structures and many associated algorithms. Study of classical abstract data types (ADTs) will be fairly comprehensive. Emphasis will be placed on matching the appropriate data structures and algorithms to application problems. Analysis of algorithms is crucial to making proper selections, so analysis is important in the course. We will study programming techniques including the definition of classes, and use of dynamic memory and linked data structures, including lists and trees. Even though the treatment of algorithms and data structures is mostly conceptual,

students are expected to be able to transform these algorithms and data structures into programs with proper approaches of software module development.

## Textbook

Mark Allen Weiss (2006) Data Structures and Algorithm Analysis in C++, 3rd Edition, Addison Wesley (try hard to get the 3rd edition)

## Prerequisites

CSE 260 and CSE 232 are required

## Graded Work

2 midterm exams,	15% each	14 Feb and 27 March
final exam,	25%	29 April
homework/programs	40%	
class participation	5%	

Final grades will be assigned based on the following scale:

>= 90%	85%	80%	75%	70%	65%	60%
4.0	3.5	3.0	2.5	2.0	1.5	1.0

The instructors reserve the right to make changes that are needed for proper delivery of the course. In particular, the grading scale cut points may be moved downward but not upward (say 85% becomes 83% in the end).

## Homework

Homework will be due on [Thursday night (11:59)], except for the first week and the weeks of midterm exams. **Homeworks require a report of results submitted in .pdf format, .doc, or other approved form.** Homeworks typically will have multiple problems and may require mathematical analysis, analysis of an existing programs or functions, or programming something new.

Programs are important artifacts of our profession. However, learning to program is NOT the main emphasis of this course; rather it is to program more efficiently and to solve problems by using appropriate algorithms. Your analysis of what a program does is much more important than the program code itself. Students completing programs within the hour prior to submission are not taking the responsibility that the level of this course implies.

## Homework deliverables

Students will submit work via Handin. Usually, there will be multiple files. A file, in approved format, outlining results is always required. If programming is required, program source files, and perhaps a Makefile, will also be submitted. Data files may also be required. Students are encouraged to do clean handwork, say for analysis, and then to scan that into image or .pdf format to be included in the report.

## Homework and program format

**Guidance on report and code format will be provided later.**

## Academic Integrity

MSU standards apply. All work submitted for credit is to be performed by the individual student. It is never appropriate to give or receive a computer file containing any part of a homework solution. It is inappropriate for students to enlist tutors to do the work that they should be doing themselves. The best source of help is to use the office hours of the TA and Instructors.

## Calendar

- 1 Jan 8: Ch 1,2: Course intro; intro to algorithm analysis
- 2 Jan 15: Ch 7: Sorting; insertion sort; quicksort; merge sort; external sorting
- 3 Jan 22: Ch 3: Lists, queues, stacks
- 4 Jan 29: Ch 4: Binary Search Trees
- 5 Feb 5: Ch 4.4 AVL trees;
- 6 Feb 12: Ch 4.5 Splay trees; Ch 4.7 B-trees; Exam I, Thursday 14
- 7 Feb 19 Ch 6: Heaps 6.1-6.4
- 8 Feb 26 Ch 5: Hashing
- 9 Mar 4 Spring Break
- 10 Mar 11 Ch 9.1, 9.2 Graphs and graph algorithms; topo sort
- 11 Mar 18 Ch 9.3: shortest path algorithms
- 12 Mar 25 Ch 9.5 spanning trees Exam II, Thur. 29 Mar
- 13 Apr 1 Ch 9.6 DFS algorithms
- 14 Apr 8 Ch 10 algorithm design techniques 10.1-10.4

15 Apr 15 Ch 10 algorithm design techniques 10.1-10.4

16 Apr 22 Ch 9.7 Complexity theory and course summary

17 Apr 29 Finals Week: Final is Tuesday, 29 April, 10:00 - 12:00 noon (The final for Section 2 is Thursday, 1 May, 3-5 PM.)