CS 120 – Data Structures and Algorithms

Spring 2007 Syllabus

Content:	This course cover advanced data structures and algorithm analysis. Topics include trees (e.g., AVL trees, splay trees, b-trees, red-black trees), hashing (separate chaining, open addressing, rehashing), priority queues (e.g., d-heaps, leftist heaps, skew heaps, binomial queues), sorting (e.g., indirect sorting, bucket sort, external sorting), the disjoint set ADT, graph algorithms, algorithm design techniques, and amortized analysis. Advanced C++ language constructs are presented and object-oriented programming is emphasized. Online course materials may be found at https://eureka.westmont.edu/.
Goals and Learning Standards:	(1) Learn about methods of organizing and accessing large amounts of data (*research and technology standard*) (2) Learn about analysis of algorithms and development of complex programs with maximum efficiency (*critical-interdisciplinary thinking standard*) (3) Learn to communicate technical information effectively through oral presentations and written assignments and papers (*written and oral communication standard*)
Professor:	Dr. Kim Kihlstrom *kimkihls@westmont.edu* http://homepage.westmont.edu/kimkihls/ Office: Math/CS Building (near post office) 565-6864 Home: 684 Circle Drive 969-6265
Office Hours:	M 1:30-3:30 pm, TW 3:15-4:45 pm
Required Text:	Weiss, *Data Structures & Algorithm Analysis in C++*, 3rd ed., Addison Wesley, 2006
Assignments:	Reading assignments are to be done **before** the class period for which they are assigned. At the beginning of most class sessions, students will be asked to write a segment of code or a response to a question based on the reading. These free-writes will form the basis for further class discussion. Homework assignments will be due about once a week. All assignments are to be submitted online through Eureka. Late assignments will be subject to a 50% penalty. While some problems will involve mathematical analysis, many will be programming problems. You are expected to submit **working code** in a source file for all programming problems. **Your code must compile to receive credit.** Grading will take into account correctness, programming style, and documentation. A larger programming project that includes a final paper will be due during the last week of class. Additionally, each student will make one oral presentation on course material during the last three weeks of class.

Exams:	There will be a midterm and a cumulative final, which will be in-class open book exams. The exams will stress both qualitative understanding of the concepts and the ability to write code.
Class Sessions:	Regular class attendance is essential for success in the class, and class participation is included in the grading. Class sessions will include free-writes, discussion of principles and ideas, abstraction, algorithm development, group problem solving, and programming labs. We will use laptop computers in class; thus, you need to bring your laptop to class each day. If you do not own a laptop, you may check one out for the semester. Additionally, each student must attend at least two natural and behavioral science seminars during the semester, typically held on Fridays at 3:30.
Grades:	The percentages of grading will be as follows: Homework assignments: 20% Class participation and seminar attendance: 15% Project: 15% Oral presentation: 15% Midterm: 15% Final exam: 20%
Honesty:	Working together on assignments is encouraged. However, **copying** another student's assignment (or portion of an assignment) is **not** allowed and will result in an F for the assignment. If you work together, you must include a **comment** indicating with whom you worked and on which part of your assignment. **Your code should never be the same or essentially the same as someone else's!** Repeated or major violations will result in an F for the course. "To plagiarize is to present someone else's work—his or her words, line of thought, or organizational structure—as our own. This occurs when sources are not cited properly, or when permission is not obtained from the original author to use his or her work. By not acknowledging the sources that are used in our work, we are wrongfully taking material that is not our own. Plagiarism is thus an insidious and disruptive form of dishonesty. It violates relationships with known classmates and professors, and it violates the legal rights of people we may never meet. Another person's 'work' can take many forms: printed or electronic copies of computer programs, musical compositions, drawings, paintings, oral presentations, papers, essays, articles or chapters, statistical data, tables or figures, etc. (The Learning Skills Centre, 1999). In short, if any information that can be considered the intellectual property of another is used without acknowledging the original source properly, this is plagiarism." From Westmont College Plagiarism Policy, http://www.westmont.edu/_academics/pages/provost/curriculum/plagiarism/

COURSE OUTLINE

Weeks 1-8

Text: Weiss, *Data Structures & Algorithm Analysis in C++*, 3rd ed.

Week/Date		Topic	Chapter/Pages
#1	T 1/9	Discrete math, recursion, C++ classes	Ch 1: pp. 1-19
	Th 1/11	C++ details, templates, matrices	Ch 1: pp. 19-41
#2	T 1/16	**NO CLASS – MONDAY CLASSES**	
	Th 1/21	Algorithm analysis, running time calculations	Ch 2: pp. 43-69
#3	T 1/23	List ADT, vector and list in STL	Ch 3: pp. 71-94
	Th 1/25	Stack ADT, queue ADT	Ch 3: pp. 94-108
#4	T 1/30	**NO CLASS**	
	Th 2/1	Binary search trees, AVL trees	Ch 4: pp. 113-149
#5	T 2/6	Splay trees, b-trees	Ch 4: pp. 149-181
	Th 2/8	Hashing	Ch 5: pp. 185-211
#6	T 2/13	Binary heap, leftist heaps	Ch 6: pp. 213-235
	Th 2/15	Skew heaps, binomial queues	Ch 6: pp. 235-257
#7	T 2/20	**NO CLASS – PRESIDENTS' HOLIDAY**	
	Th 2/22	Lower bound, shellsort, heapsort, mergesort	Ch 7: pp. 261-279
#8	T 2/27	Quicksort, indirect sorting	Ch 7: pp. 279-297
	Th 3/1	**MIDTERM EXAM**	Chapters 1-7

COURSE OUTLINE

Weeks 9-15

Text: Weiss, *Data Structures & Algorithm Analysis in C++*, 3rd ed.

Week/Date		Topic	Chapter/Pages
#9	T 3/6	Bucket sort, external sorting	Ch 7: pp. 297-311
	Th 3/8	Disjoint set class	Ch 8: pp. 315-336
	3/12 – 3/16	**SPRING VACATION**	
#10	T 3/20	Graphs, shortest-path algorithms	Ch 9: pp. 339-356
	Th 3/22	Network flow problems	Ch 9: pp. 356-372
#11	T 3/27	Minimum spanning tree	Ch 9: pp. 372-388
	Th 3/29	Depth-first search, NP-completeness	Ch 9: pp. 388-404
#12	T 4/3	Greedy algorithms	Ch 10: pp. 409-427
	Th 4/5	Divide and conquer	Ch 10: pp. 427-442
#13	T 4/10	Dynamic programming, randomized algorithms	Ch 10: pp. 442-464
	Th 4/12	Backtracking algorithms	Ch 10: pp. 464-485
#14	T 4/17	Binomial queues, skew heaps	Ch 11: pp. 491-502
	Th 4/19	Fibonacci heaps, splay trees	Ch 11: pp. 502-515
#15	T 4/24	Top-down splay trees, red-black trees, skip lists	Ch 12: pp. 517-540
	Th 4/26	AA-trees, treaps, k-d trees, pairing heaps	Ch 12: pp. 522-563

FINAL EXAM:
Tuesday, May 1st, 8:00-10:00 am.