Document: Data Structures Course Administration
Revised: August 21, 2006
Course Title: Data Structures
Course Number: CMSC 3613
Section: CRN 10870 Monday, Wednesday, and Friday 11:00 – 11:50 a.m. MCS 113
Instructor: Dr. Thomas R. Turner; Office: MCS 134; Work Phone: 974-5383 e-mail: trturner@ucok.edu
Office Hours: Monday, Wednesday, and Friday: 8:00 – 9:00 a.m.; 10:00 – 11:00 a.m.
Tuesday and Thursday 8:30 – 9:30 a.m.
and by appointment.
Text: Kruse, R. L. and Ryba, A. J. Data Structures and Program Design in C++, Prentice-Hall, Inc. 1999 ISBN 0-13-768995-0
References:
1. Weiss, Mark Allen. Data structures and algorithm analysis in C++, 2nd Edition Addison Wesley Longman, Inc. 1999 ISBN 0-201-36122-1
2. Stroustrup, B. *The C++ Programming Language* 3rd Ed.; Addison-Wesley 1997 ISBN 0-201-88954-4

Prerequisites:
1. CMSC 2613, Programming II
2. CMSC 2813, Assembly Language Programming
3. CMSC 1613, Programming I
4. CMSC 1513, Beginning Programming (Pascal)
5. MATH 2313, Calculus 1
6. Statistics, MATH 3103
7. MATH 2123, Discrete Mathematics

Programming Projects: ***Seven*** (7) projects and ***three*** (3) reports are assigned. Projects and reports are due at the beginning of the class session. Source for all projects must reside in the root directory of your student account on the department computer. No credit is given to projects or reports that are turned in late. No excuses will be accepted for late projects or reports. *If you cannot turn in an assignment in class, put the assignment in my mailbox in the Computing Science Office, MCS 117.* ***One-quarter of the value of the project will be deducted if the project is slipped under my office door.***

Course Scoring:

Task	**Date**	**Value**
Test 1	9-25	150
Test 2	10-30	150
Final Test	12-11	300
Programming Projects	Table 2	210
Reports	Table 3	90
Total		**900**

Grading: **A**: 90% (810) and above; **B**: 80-89% (720-809); **C**: 70-79% (630-719); **D**: 60-69% (540-629); **F**: 59% (0-540) and below.
Notice: Beepers and cellular phones are prohibited in class.
Absences: You ***must*** secure written permission from your instructor to submit a test after it is due. No makeup tests are offered. Lecture notes are distributed only on the day of the lecture. Your instructor does not make notes available to you if you are unable to attend class.
Tardiness: Students who arrive after roll is called may be prohibited from entering the classroom at the instructor's discretion.
Caveat: This lecture schedule, programming projects and due dates, number and dates of tests are all subject to change. Changes are presented in class ***You*** are responsible for the material presented in class.
Class Web Page: The course administration and assignments can be found on URL **http://www.comsc.ucok.edu/~trt/cs3613.html**
Course Directory The course directory is on the department computer (cs.ucok.edu). You can find test data files in the course directory**. ~tt/cs3613/**
Student Disabilities: Students with disabilities who require accommodations may contact Disability Support Services. **http://bronze.ucok.edu/disability_support/**

Table 1. Lecture Schedule

Lecture	Date	Topic	Text
1	8-21	Course Administration	
2	8-23	Submitting Projects	Lecture 1
3	8-25	Command Line Arguments	Lecture 2
4	8-28	Radix Sort	Lecture 3
5	8-30	List Overview	Lecture 4
6	9-1	List Implementation	Lecture 5
7	9-6	Program structure and makefiles	Lecture 6
8	9-8	Margin	
9	9-11	**Turn in p01: Radix sort**	
		Algorithm Analysis	
		$O(f(n))$, $\Omega(g(n))$, $\Theta(h(n))$, Step Counting	Lecture 7
10	9-13	Radix Sort Time Complexity	Lecture 8
11	9-15	Recursion, Recurrence Relations, and Induction	Lecture 9
12	9-18	Trees	Lecture 10
		Turn in p02: Time Complexity Analyses	
13	9-20	Binary Tree Node Removal	Lecture 11
14	9-22	AVL Trees, single and double rotation	Lecture 12
		Distribute Test 1	
15	9-25	***Turn in Test 1***	
		Test 1 Reprise	
16	9-27	AVL Tree Implementation	Lecture 13
17	9-29	AVL Tree Algorithm Analysis	Lecture 14
18	10-2	**Turn in p03: AVL Trees**	
19	10-4	Library Research Report	r01: Library Research Report
20	10-6	B-Trees	Lecture 15
21	10-9	B-Trees implementation	Lecture 16
		Turn in r01: Library Research	
22	10-11	B-Tree implementation	Lecture 16
23	10-13	B-Tree Algorithm Analysis	Lecture 17
24	10-16	**Turn in p04: B+Trees**	
25	10-18	Report format and example	r02: AVL Tree Report
26	10-23	Hashing	Lecture 18
		Turn in: r02: AVL Tree Report	
27	10-25	Open Addressing Implementation	Lecture 19
28	10-27	***Test 2 Distributed***	
29	10-30	***Turn in Test 2***	
		Test 2 Reprise	
30	11-1	Separate Chaining Implementation	Lecture 20
31	11-3	Hashing Algorithm Analysis	Lecture 21
32	11-6	Graphs	Lecture 22
		Turn in p05: Hashing	
33	11-8	Unweighted Shortest Path	Lecture 23
34	11-10	Overview of Lex	Lecture 24
		Regular Expressions	Lecture 25
35	11-13	Nondeterministic Finite Automata	Lecture 26
		Deterministic Finite Automata	Lecture 27
		Turn in p06: Unweighted Shortest Path	

Table 1. Lecture Schedule (continued)

Lecture	Date	Topic	Text
36	11-15	Regular Expressions to NFA	Lecture 28
37	11-17	Conversion of an NFA into a DFA	Lecture 29
38	11-20	**Turn in p07: Pascal Scanner**	
39	11-27	Making and evaluating ethical arguments	Lecture 50
40	11-29	Identifying and evaluating ethical choices	Lecture 51
41	12-1	Understanding the social context of design	Lecture 52
42	12-4	Identifying assumptions and values	Lecture 53
		Turn in r03: Ethics	
43	12-6	Margin	
44	12-8	***Distribute Final Exam***	
45	***12-11***	***Turn in your Final Exam, Monday, December 11, 2006 at 11:00 a.m.***	***Comprehensive***

Table 2. Programming Projects

Project	Due	Value	Description
1	9-11	30	**p01**: Radix Sort
2	9-18	30	**p02**: Time Complexity Analysis
3	10-2	30	**p03**: AVL Trees
4	10-16	30	**p04**: B+Trees
5	11-6	30	**p05**: Hashing
6	11-13	30	**p06**: Unweighted Shortest Path
7	11-20	30	**p07**: Pascal Scanner
Total		**210**	

Table 3. Reports

Report	Due	Value	Description
1	10-9	30	**r01**: Library research
2	10-23	30	**r02**: AVL Tree Report
3	12-4	30	**r03**: Ethics
Total		**90**	