UNIVERSITY OF NEBRASKA AT OMAHA
COURSE SYLLABUS/DESCRIPTION

Department and Course Number	CSCI 3320
Course Title	Data Structures
Course Coordinator	Jong-Hoon Youn
Total Credits	3
Date of Last Revision	May 13, 2014

1.0 Course Description Information

1.1 Catalog description:
The purpose of this course is to familiarize students with algorithms and programming techniques for different data structures used in software systems. Students will also learn time analysis of computer programs, various searching and sorting techniques, program solving strategies and storage management algorithms.

1.2 Prerequisites of the course:
CSCI 1620 - Introduction to Computer Science II: Programming Languages: Java or C++
MATH 1310 (or equivalent)

1.3 Overview of content and purpose of the course:
This course discusses data structures, methods of organizing large amounts of data, algorithm analysis, and the estimation of the running time of algorithms. At the end of the course, the students will be able to understand the purposes and methods of the most commonly occurring data structures, analyze the data structure needs of particular problems, compare the efficiency of various implementations and algorithms, and write applications using data structures discussed in the course. The major data structures covered in this course include trees, hash tables, heaps, and graphs.

1.4 Unusual circumstances of the course.
None.

2.0 Course Justification Information

2.1 Anticipated audience / demand:
This is a core undergraduate computer science course intended for computer science major students. It is also offered as a crosslisted graduate computer science course. The crosslisted course is intended as a transition course for graduate students who do not have a pure computer science background

2.2 Indicate how often this course will be offered and the anticipated enrollment:
This course will be offered regularly in the Fall and Spring semesters with an anticipated enrollment of around 60 students per semester.

2.3 If it is a significant change to an existing course, please explain why it is needed:
N/A

3.0 List of performance objectives stated in learning outcomes in a student's perspective:

3.1 Familiarize students with different data structures such as lists, stacks, queues, trees, heaps, hash-tables and file structures which are used to store runtime data of a computer program

3.2 Discuss different data manipulation strategies including different algorithms for sorting and searching items within various data structures

3.3 Focus on different program solving problems including greedy programming, dynamic programming, and backtracking

3.4 Encourage students to develop algorithms and solutions to different data structures related problems

3.5 To stress the importance of efficiency in terms of time and storage space for the computer programs developed by the students

4.0 Content and Organization Information

4.1 List of major topics to be covered in chronological sequence (specify number of weeks on each).

- 4.1.1 Introduction (1.5 hours)
- 4.1.2 Algorithm Analysis and Time complexity (4.5 hours)
- 4.1.3 Lists, Stacks, Queues (2 hours)
- 4.1.4 Trees (9 hours)
  - 4.1.4.1 Binary Tree
  - 4.1.4.2 Binary Search Tree
  - 4.1.4.3 AVL Trees
  - 4.1.4.4 B-Trees
- 4.1.5 Hashing (4 hours)
  - 4.1.5.1 Hash Function
  - 4.1.5.2 Separate Chaining
  - 4.1.5.3 Open Addressing
  - 4.1.5.4 Rehashing
  - 4.1.5.5 Extendible Hashing
- 4.1.6 Priority Queues (Heaps) (4.5 hours)
  - 4.1.6.1 Binary Heap
  - 4.1.6.2 d-heap
  - 4.1.6.3 Leftist Heap
  - 4.1.6.4 Binomial Queue
- 4.1.7 Sorting (6 hours)
  - 4.1.7.1 Insertion Sort
  - 4.1.7.2 Shell Sort
  - 4.1.7.3 Heap Sort
  - 4.1.7.4 Merge Sort
  - 4.1.7.5 Quick Sort
  - 4.1.7.6 Indirect Sorting
  - 4.1.7.7 Bucket Sort
  - 4.1.7.8 External Sorting
- 4.1.8 Graph Algorithms (4.5 hours)

4.1.8.1 Graph Representations
4.1.8.2 Djikstra's Algorithm
4.1.8.3 Minimum Spanning Tree

4.1.9 Algorithm Design Techniques (4.5 hours)
4.1.9.1 Greedy Algorithms
4.1.9.2 Divide and Conquer
4.1.9.3 Dynamic Programming

5.0 Teaching Methodology Information

5.1 Methods:
Teaching methods will include in-class lectures, hands-on lab exercises, in-class quizzes, homework assignments, case studies, and demonstrations of the code written by the student

5.2 Student role:
Students are expected to attend all lectures and labs, participate in class discussions, and complete assigned homework and examinations.

6.0 Evaluation Information

6.1 Describe the typical types of student projects that will be the basis for evaluating student performance:
Homework assignments will typically be small problems requiring programming solutions as well as short writing assignments. The following problems can be assigned as a short written assignment as well as a programming assignment:

- Implement the solutions to problems involving lists, stacks and queues
- Implement the solutions to problems involving trees
- Implement the solutions to problems involving heaps
- Implement the solutions to problems involving sorting methods
- Implement the solutions to problems involving graphs
- Implement the solutions to problems involving problem solving strategies

6.2 Describe the typical basis for determining the final grade (e.g. weighting of various student projects):.

Homework Assignment: 40%
In-class quizzes: 10%
Midterm exam:30%
Final exam: 20%

6.3 Grading type:

Percent	Grade	Percent	Grade
97 – 100	A+	77 – 79	C+
94 – 96	A	70 – 76	C
90 – 93	A–	70 – 73	C–
87 – 89	B+	67 – 69	D+
84 – 86	B	64 – 66	D
80 – 83	B–	60 – 63	D–
		0—59	F

7.0 Resource Material Information

7.1 Textbooks and/or other required readings used in course.
"Data Structures and Algorithms Analysis in Jave" by Mark A. Weiss, Addison Wesley, 3rd edition, 2012.

7.2 Other suggested reading materials:.

7.2.1 "A Practical Introduction to Data Structures and Algorithm Analysis," by Clifford Shaffer, Prentice Hall, 2nd edition, 2001.

7.2.2 " Algorithms in Java, Parts 1-4 (3rd Edition)," Robert Sedgewick, 3rd edition, Addison Wesley, 2002.

7.2.3 "Introduction to Algorithms," Cormen, Leiserson and Rivest, McGraw Hill, 3rd edition, 2009.

7.2.4 "Data Structures and Problem Solving in C++," Mark A. Weiss, 2nd edition, Addison Wesley, 2002.

7.3 Current bibliography and other resources:
None

8.0 Computer Science Accreditation Board (CSAB) Category Content (class time in hours):

CSAB Category	*Core*	*Advanced*
Data structures	14.5	5
Computer organization and architecture		
Algorithms and software design	19.5	
Concepts of programming languages		

9.0 Oral and Written Communications

Every student is required to submit at least ___0__ written reports (not including exams, tests, quizzes, or commented programs) to typically _0____ pages and to make __0___ oral presentations of typically __0___ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

10.0 Social and Ethical Issues

None

11.0 Theoretical content

Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

1. Time and Space complexity of computer programs (3 hours)
2. Usage and analysis of different data structures used in computer science problems including lists, trees, hashtables, heaps, and graphs (20 hours)
3. Sorting techniques used in computer science problems and their analysis (6 hours)
4. Program Solving Techniques in computer science such as greedy approach, heuristic-based approach, dynamic programming, backtracking, and mathematical analysis of each of these techniques (10 hours)

12.0 Problem analysis

Please describe the analysis experiences common to all course sections.
Students will be required to analyze the problems discussed in class along with the instructor. Problems given in homework assignments should be analyzed by the student. Problem analysis comprise the following steps:

1. Determine the most appropriate data structure for the problem
2. Determine the time complexity in solving the problem with the data structure used
3. Determine the space complexity in solving the problem with the data structure used
4. Compare the time and space complexities in solving the problem with other data structures

13.0 Solution design

Please describe the design experiences common to all course sections.
Students taking the course are expected to develop the software design for all the problems given as part of the homework assignments, quizzes and examinations. The design involves the implementation of software objects to encapsulate the data structures the students use to solve the problems, and the interaction method between those objects.

CHANGE HISTORY

Date	***Change***	***By whom***	***Comments***
10/11/2202	Initial ABET version	Dasgupta	
06/13/2003	Cleanup	Wileman	
10/27/2008	Checked syllabus currency	Dasgupta	1. Added programming language “Java” in section 1.4.1 2. Added mapping to CS outcomes below
05/13/2014	Checked syllabus currency, and reformatted to a new template	Youn	Removed some topics that are not discussed in depth from the *Content and Organization Information* section, and updated

			the number of hours on each topic Updated *Resource Material Information*.