MASTER SYLLABI

7/1/03

MINNESOTA SCHOOL OF BUSINESS
GLOBE COLLEGE OF BUSINESS
TECHNICAL COURSE SYLLABUS

COURSE NUMBER:	**SD290**	COURSE TITLE:	ALGORITHMS & DATA STRUCTURES IN C++
COURSE LENGTH:	12 WEEKS	CREDIT HOURS:	3
PREREQUISITES:	SD235	CONTACT HOURS:	50 (LECTURE 10/ LAB 40)

TEXT: DATA STRUCTURES: A PSEUDOCODE APPROACH WITH C++, Gilberg & Farouzan, Brooks/Cole, ISBN: 053495216X

COURSE DESCRIPTION: The course will give students an overview of the use of Advanced Data Type Structures (ADTs) in programming. The language used will be C++, but it will be applicable to many other programming languages.

OBJECTIVES: Upon completion of this course students will be able to:

1. Demonstrate the use of data structures in a programming language.
2. To be able to utilize the best data structure for a particular program.
3. Learn when to optimize a program and when to leave it sub optimal.
4. Be able to apply ADT's to real problems.
5. Apply the ideas of linked lists and arrays to programs.
6. Apply the concept of graph algorithms to solve problems.

COURSE OUTLINE:

Topic/s & Class Activities **Required Reading**

Unit 1. Introduction to Advanced Data Types and class templates, algorithmic efficiency and Searching algorithms

Unit 2. Singly linked lists

Unit 3. Stack ADT – Linked List implementations

Unit 4. Stack ADT – Array implementations

Unit 5. Queue ADT – Linked List implementations

Unit 6. Queue ADT – Array implementations

Unit 7. AVL Tree ADT

Unit 8. B-Tree ADT

Unit 9. Graph ADT's

Unit 10. String ADT's

Week 1. Introduction to ADT's, Algorithm analysis, Algorithm efficiency –
Searching –
Search Algorithms, list searches & collision resolution –

	Topic/s & Class Activities	**Required Reading**
Week 2.	Linked lists – Linked list Algorithms – Processing & Building a linked list – Basic Stack operations and implementation – 8 queens problem – Stack ADT array implementation –	.
Week 3.	Queues – C++ Data Categorization – Queue ADT's Linked list & Array implementation –	
Week 4.	Recursion, factorial case study, how it works – Designing Recursive algorithms – Fibonacci and Tower of Hanoi –	
Week 5.	Introduction to Binary, Expression and General Trees – Search Trees, Binary and AVL - AVL tree implementation and ADT's –	
Week 6.	Heap definition, structure and basic algorithms – Heap applications and programming –	
Week 7.	Multiway Trees, B-Trees, B-Tree ADT's –	
Week 8.	Advanced sorting Concepts – Insertion, Selection and Exchange Sorts – External Sorts –	
Week 9.	Graph terminology, Structures and operations – Storage structures and algorithms – Networks and ADT's –	
Week 10.	**Projects**	
Week 11.	**Projects**	
Week 12.	**Presentation of Projects**	

INSTRUCTIONAL METHODS: The instruction will be delivered primarily by lecture, discussion, practice programming problems and projects. Programming using a computer in class is an essential part of the instruction.

EVALUATION METHODS:

Grades are an indicator of overall performance, achievement and participation. Students are responsible for completing all course requirements on time to receive credit. There will be no written tests; all evaluation will be on written and oral projects. The time for a final exam will be used for formal project presentations.

Written projects / reports	30%
Classroom exercises	20%
Final Project / Exam	30%
Participation	20%

The final grade for the course is based on an accumulation of points in each of the above areas and weighted accordingly. A total of 200 points are possible. These points are based on the following percentages:

100-90%	A
89-80%	B
79-70%	C
69-60%	D
59% and lower	N/C