MASTER SYLLABI

7/1/03

MINNESOTA SCHOOL OF BUSINESS
GLOBE COLLEGE
TECHNICAL COURSE SYLLABUS

COURSE NUMBER:	**SD251**	COURSE TITLE:	C++ PROGRAMMING
COURSE LENGTH:	12 WEEKS	CREDIT HOURS:	4
PREREQUISITES:	SD230	CONTACT HOURS:	60 (LECTURE 20 / LAB 40)

TEXT: C++: HOW TO PROGRAM, 3RD ***Edition***, H. M. Deitel & P.J. Deitel, Prentice Hall
IBSN: 0-13-089571-7

SUPPLEMENTAL TEXTS: AN INTRODUCTION TO PROGRAMMING WITH C++, Diane Zak
OBJECT-ORIENTED PROGRAMMING USING C++, Joyce Farrell

COURSE DESCRIPTION: This course will introduce the student to the C++ programming language. This course will build on what has been learned in C Programming, show how C++ expands on C, demonstrate the object-oriented features of C++, and cover many of the new features that C++ offers.

OBJECTIVES: Upon completion of this course, the student will be able to:

1. Identify the improvements on the C language.
2. Describe the additional data types.
3. Apply new operators and operator overloading.
4. Explain the additional capabilities of functions, such as overloading and inlines.
5. Describe the elements of object-oriented programming.
6. Declare, define and instantiate a class.
7. Create class constructors and destructors.
8. Explain public and private inheritance.
9. Determine when to use virtual functions.
10. Use the iostream library.
11. Recognize and use functions and class templates.

COURSE OUTLINE:

	Topics & Class Activities	**Required Reading**
Week 1	Introduction to computers and Control Structures	Chapter 1,2
Week 2	Functions and Arrays	Chapters 3, 4
Week 3	Pointers and Strings	Chapter 5
Week 4:	Classes an Data Abstraction	Chapter 6

	Topics & Class Activities	**Required Reading**
Week 5	Classes: Part II	Chapter 7
Week 6	Review / Midterm	
Week 7	Operator Overloading	Chapter 8 **Project #1 Due**
Week 8	Inheritance	Chapter 9 **Project #2 Assigned**
Week 9	Virtual Functions & Polymorphism	Chapter 10
Week 10	Stream I/O	Chapter 11
Week 11	Templates	Chapter 12
Week 12	**Final Exam**	**Project #2 Due**

INSTRUCTIONAL METHODS: The student will need access to a C++ compiler toolset. This course is intended to be platform-neutral; either Windows or Linux may be used to create the programs required for this course. It is recommended that the GNU gcc compiler be used. This is the native compiler for Linux and is part of the Cygwin tools provided by Cygnus Solutions, a Red Hat company. It is recommended that students be provided a CD-ROM available with the Cygwin package, as the current download is about 13 MB. Visual C++ or Borland C++ may be alternatives. Understand that C++ is a newer language than C and the various C++ compilers may not yet support all that is specified in the ANSI/ISO Draft Standard. Some of these lessons could take more than a week. Some lessons are shorter than others to provide extra time for review and discussion. In addition to performing a few exercises found within each chapter, two programming projects will be assigned.

EVALUATION: Student grades will be based on the following assignments and points:

Exercises:	200
Projects (2):	300
Midterm:	200
Final:	200
Participation:	100
	1,000 points total

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 1000 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