

# COS 325

## Computer Programming: C++

Date: April 17, 2003

Division: Natural and Behavioral Sciences

Number of Credits: 3

Location: Nadeau 109

Meeting Times: 9:30 – 10:50 PM MTh

### Course Description

Prerequisite: Cos 260. Pursues a more in-depth view of a common high level programming language around which course is centered. Covers advanced problem analysis, algorithm development, and data structure selection and usage techniques as they apply to the language. Offered in alternate years. 3 credit hours.

### Instructor

<b>Instructor</b>	Tony Gauvin	<b>E-mail</b>	TonyG@maine.edu
<b>Phone</b>	(207) 834-7519	<b>Office Hours</b>	1:00–1:50 PM MW Or by appointment
<b>Office</b>	236 Nadeau Hall		

### Texts

*How to Program C++*

Author: Deitel & Deitel

Publisher: Prentice Hall

ISBN: 0-13-089571-7

*Getting Started with Microsoft Visual C++ 6 with an Introduction to MFC*

Author: Deitel & Deitel

Publisher: Prentice Hall

ISBN: 0-13-016147-0

## **Required Materials**

**Two 3.5" high density (HD) floppy diskettes**, one to be used as a backup. **Diskette holder** to keep your disks in. These are available at the campus book store.

**OR**

Two 100MB Zip Disc (available at Campus Book Store)

## **Goals and Objectives**

- Provide an introduction to the Unified Modeling Language (UML), a graphical language that allows system developers to represent their object-oriented designs using a common notation
- Understand the concepts of object-oriented programming
- Understand elementary object-oriented analysis and design
- Understand encapsulation
- Understand extensibility
- Understand software reusability
- Understand how to use class libraries including the Microsoft Foundation Libraries (MFC)
- Learn the use the Microsoft Visual C++ 6 integrated development environment (IDE)
- Perform C++-style input/output
- Perform dynamic memory allocation with *new* and *delete*
- Use default function arguments
- Build abstract data types using classes
- Understand and use constructor and destructor functions to initialize and destroy class objects
- Overload functions
- Create function templates
- Overload operators to work with user-defined classes
- Create container classes
- Create class templates
- Use inheritance to build class hierarchies
- Perform inheritance with class templates
- Understand and use polymorphism and virtual functions
- Understand exception handling
- Understand multiple inheritance

## **Requirements**

Students will be required to demonstrate their knowledge of C++ programming through examinations, assigned projects and the Capstone project. Individualized Capstone projects will allow each student to develop an in-depth understanding of a particular aspect of C++ programming and demonstrate his/her ability to apply newly learned skills and concepts to the instructor and his classmates.

## **Other Resources**

WebCT

<http://webct.umfk.maine.edu:8900/>

Author Website

<http://www.prenhall.com/deitel/>

<http://www.deitel.com/>

Microsoft C++ Site

<http://msdn.microsoft.com/visualc/default.asp>

## **Method of Instruction**

Lectures and demonstrations covering the above listed material will be further supplemented with **hands-on** C++ programming lessons. Lectures will provide general conceptual overviews and demonstrations of each concept and technique. Projects and project assignments will be (to the greatest extent possible) tailored to the participants' needs. Materials from outside sources will be used for added emphasis. All information and material presented in class and through assigned readings are to be considered **fair game** in any exam.

## **Method of evaluation**

The examinations will be comprised of questions that test the student's knowledge of the concepts along with their ability to apply those concepts to real-world programming issues. The Capstone project will be evaluated on content, form, presentation and the ability of the student to conduct an independent project within the C++ programming area. Assignments will be required to demonstrate understanding of the concepts being discussed. Students will be expected to participate and collaborate with their classmates and the course instructor. Attendance at all classes without participation does not warrant a full 10% component in the over-all course grade.

## Course Grade Calculation

<b>Exams (5 @ 7% each)</b>	<b>35%</b>
<b>Assignments (10 @ 3.5 % each)</b>	<b>35%</b>
<b>Capstone Project</b>	<b>20%</b>
<b>Attendance/Participation</b>	<b><u>10%</u></b>
	<b>100%</b>

## Grading Scale

A	90 - 100 %
B	80 - 89 %
C	70 - 79 %
D	60 - 69 %
F	0 - 59 %

## Absence Policy

As participation is considered essential to a successful educational experience, students may register at most two unexcused class absences before being considered dropped from the course. For each unexcused class absence, two and a half percentage points of the total 100 will be deducted from your final score. Excused absences are those for which prior arrangements have been made with the instructor or extenuating circumstances can be clearly documented. Please check with the instructor and make arrangements prior to any planned absences. There will be no makeup exams, students must instead make arrangements with the instructor to take the exam prior to the planned absence or forfeit the opportunity to complete it. Students failing to maintain a passing grade at any point during the semester may be involuntary dropped from the course through an instructor-initiated withdrawal process. Overall, it is essential to keep the instructor informed of your progress as well as any difficulties you may be experiencing.

## Course Schedule

Week	Subject	Required Reading	Notes
1	Introduction	C++ Chap 1 MFC Chap 1	
2	Structures	C++ Chap 2	HMWK 1
3	Functions	C++ Chap 3	HMWK 2
4	Arrays, Pointers and Strings	C++ Chap 4 & 5	HMWK 3

5	Classes	C++ Chap 6 & 7	HMWK 4
6	Operator Overloading and Inheritance	C++ Chap 8 & 9	HMWK 5
7	Virtual Functions, Polymorphism, and Input/output Streams	C++ Chap 10 & 11	HMWK 6
8	Using templates and exception handling	C++ Chap 12 & 13	HMWK 7
9	File processing and Data Structure	C++ Chap 14 & 15	HMWK 8
<b>Spring Break</b>			
10	Bits, Characters, strings and structures; The Preprocessor; C Legacy Code Topics	C++ Chap 16, 17 & 18	
11	String processing and the Standard Template Library	C++ Chap 19 & 20	HMWK 9
12	MFC Programming	MFC Chap 2 & 3	
13	MFC GUI's and Graphics	MFC Chap 4 & 5	HMWK 10
14	Review		
Finals	Capstone Presentations		

## **Examination Dates**

(Subject to change)

<b>Date</b>	<b>Subject</b>	<b>Type</b>
February 11	C++ Chap 1, 2 & 3, MFC Chap 1	Comprehensive
February 28	C++ Chap 4, 5, 6, 7, 8 & 9	Comprehensive
March 14	C++ Chap 10, 11, 12, 13, 14, & 15	Comprehensive
April 8	C++ Chap 16, 17, 18, 19 & 20	Comprehensive
April 29	MFC Chap 1, 2, 3, & 4	Comprehensive
May 6	Capstone Presentations	Individualized

## **Special Notes**

Students **must** record on **all** submitted materials their **name, instructor name(s), course and assignment number, and date** of submission. Students are encouraged to retain for their record a personal copy of **all** submitted materials.

Students are encouraged to regularly submit during non-class hours constructive suggestions regarding delivery of the course.

The instructor(s) reserve(s) the right to make necessary changes to the syllabus in light of any circumstances occurring during the delivery of the course provided such changes are announced in class.

Students are required to observe appropriate error prevention techniques throughout the semester. For example, students must frequently save their work to disk to prevent loss. Loss due to power fluctuations, illegal instructions, viruses, or any other reason beyond the instructors' control will not constitute adequate reason for accommodations.

Assignments are due no later than the beginning of the next class. Following submission, students are advised to not change the files in any way nor attempt to resubmit them unless specifically instructed to do so by the instructor.

Assignments may be submitted during class or placed in the appropriate instructor's mailbox no later than the due date. Assignments submitted beyond the due date will receive a **twenty-point** reduction per day.