

LSC Use Only
Number: _____
Submission Date: _____
Action-Date: _____



UWUCC USE Only
Number: _____
Submission Date: _____
Action-Date: _____

01-12a
00-55a

App 4/10/01
Senate App 12/4/01

CURRICULUM PROPOSAL COVER SHEET
University-Wide Undergraduate Curriculum Committee

I. CONTACT

Contact Person Tia Watts / Charles Shubra Phone 724-357-4492

Department Computer Science

II. PROPOSAL TYPE (Check All Appropriate Lines)

COSC210 **COURSE** OO and GUI Programming

Suggested 20 character title

X New Course* COSC210 Object Oriented and GUI Programming

Course Number and Full Title

 Course Revision _____

Course Number and Full Title

 Liberal Studies Approval+
for new or existing course _____

Course Number and Full Title

 Course Deletion _____

Course Number and Full Title

 Number and/or Title Change _____

Old Number and/or Full Old Title

New Number and/or Full New Title

 Course or Catalog Description Change _____

Course Number and Full Title

 PROGRAM: Major Minor Track

 New Program* _____

Program Name

 Program Revision* _____

Program Name

 Program Deletion* _____

Program Name

 Title Change _____

Old Program Name

New Program Name

III. Approvals (signatures and date)

Tia Watts
Department Curriculum Committee

[Signature] 01/31/01
College Curriculum Committee

+Director of Liberal Studies (where applicable)

[Signature]
Department Chair

[Signature]
College Dean

*Provost (where applicable)

COSC 210
Object Oriented and GUI Programming
Syllabus of Record

I. Catalogue Description

COSC 210 Object Oriented and GUI Programming

(3c-0l-3sh)
3 class hours
0 lab hours
3 semester hours

Prerequisite: COSC110

An in-depth introduction to the Object Oriented Programming (OOP) paradigm. The focus will be on designing, implementing, and using objects. We will cover function and operator overloading, templates, inheritance and polymorphism. This course will also include an introduction to Graphical User Interface (GUI) design and programming.

II. Course Objectives

The student will:

1. Learn the fundamental concepts of the OOP paradigm.
2. Implement object definitions.
3. Incorporate objects and arrays of objects in application programming.
4. Use overloading, templates, and inheritance when designing, implementing, and using objects.
5. Design Object Oriented applications.
6. Learn the fundamental concepts of GUI design.
7. Develop and test object-oriented GUI programs.

III. Course Outline

- | | |
|---|---------|
| A. Introduction to the Object Oriented Programming Paradigm | 2 hours |
| 1. Overview of the fundamental concepts of OOP | |
| 2. Object model terminology | |
| 3. Introduction to the language and programming environment | |
| B. (C++) Programming Basics | 4 hours |
| 1. Data representation and standard types | |
| 2. Literals and literal types | |
| 3. Namespaces | |
| 4. Input and output (I/O) | |
| 5. Operators and expressions | |
| 6. Control flow constructs | |
| 7. Functions and return types | |
| 8. User defined data structures | |
| 9. Static and dynamic arrays | |
| 10. Structs and unions | |
| 11. Memory allocation | |
| 12. Debugging techniques | |
| C. Defining Objects | 4 hours |
| 1. Encapsulation | |
| 2. Constructors and destructors | |
| 3. Member variable and functions | |

5. Public, protected, and private declarations	
6. Object initialization	
D. Overloading	3 hours
1. Function overloading	
2. Overloading resolution	
3. Operator overloading	
4. Friend functions	
E. Using Objects	4 hours
1. Scoping	
2. Static and dynamic objects	
3. Arrays of objects	
F. Object Design	3 hours
1. Goals of Object Oriented Software	
G. Templates	4 hours
1. Template functions	
2. Simple sorting techniques (exchange, insertion, selection)	
3. Template object definitions	
H. Inheritance	4 hours
1. Derived types	
2. Virtual functions	
3. Declarations	
4. Single and multiple inheritance	
5. Public vs. private inheritance	
6. Virtual Derivations	
I. Run-Time Type Identification	2 hours
1. Polymorphism	
J. Exception Handling and Debugging	3 hours
1. Exception handling functions	
2. Expected and unexpected exceptions	
3. Using a debugger	
K. Graphical User Interface Design	2 hours
1. Goals of GUI software	
L. Implementing GUIs using Object Oriented Programming	5 hours
1. Concepts and terminology	
2. Dialogue Interface	
3. Single Document Interface	
4. Multiple Document Interface	

IV. Evaluation Methods

Evaluation:

Exams: 3 (50-60%) (including final)

Projects: 5-6 (30-40%)

Quizzes, Homework, and Lab Exercises: (10-20%)

Grading Scale:

The standard grading scale will be used.

90-100% : A; 80-89% : B; 70-79% : C; 60-69% : D; below 60% : F.

Attendance policy:

The attendance policy will conform to the University wide attendance criteria.

Samples of projects assigned to help students fulfill course objectives:

Project 1. Develop a new numerical class such as rational, complex, or Roman. This project will introduce the students to the class syntax, private and public attributes and methods.

Project 2. Expand the class developed for project 1 to incorporate operator overloading. This project will include the syntax for operator overloading and friend functions.

Project 3. Develop a template container class. This project will include template syntax, and the use of dynamic arrays.

Project 4. Develop a hierarchical set of classes such as quadrilaterals. This project will include inheritance, polymorphism, protected attributes and methods and overloading.

Project 5. Develop a Dialog Based Graphical User Interface application. This project will introduce the concepts of GUI design and development. The project will also include the use of a large library of predefined classes.

V. Textbook(s)

Deitel, H.M. and Deitel, P.J.
C++ How to Program. Second Edition.
Prentice Hall PTR, Upper Saddle River, NJ. 1998.

Deitel, H.M., Deitel, P.J., Nieto, T.R., and Strassberger, E.T.
Getting Started with Microsoft Visual C++ 6 with an Introduction to MFC
Prentice Hall PTR, Upper Saddle River, NJ. 1998

VI. Special Resource Requirements

VII. Bibliography

1. Anderson, Paul and Anderson, Gail. Navigating C++ and Object-Oriented Design, Prentice Hall PTR, Upper Saddle River, New Jersey. 1998.
2. Dale, Nell, Weems, Chip, and Headington, Mark, Programming in C++, Jones and Bartlett Publishers, Sudbury, Mass. 2001.
3. Dattatri, Kayshav. C++ Effective Object-Oriented Software Construction, Prentice Hall PTR, Upper Saddle River, NJ. 2000.
4. Eckel, Bruce. Thinking in C++, 2nd Edition. Prentice Hall PTR, Upper Saddle River, New Jersey. 2000.
5. Friedman, Frank L. and Koffman, Elliot B. Problem Solving, Abstraction, and Design Using C++, Addison Wesley Longman, Inc, Reading, Mass. 2000.
6. Gregory, Kate. Using Visual C++ 6, QUE, Indianapolis, Indiana. 1998. (Optional text)
7. Gurewich, Ori and Gurewich, Nathan. Teach Yourself Visual C++ in 21 Days, SAMS Publishing, Indianapolis, Indiana. 1998.
8. Main, Michael and Savitch, Walter. Data Structures and Other Objects Using C++, Addison Wesley Longman Publishing Company, Reading, Massachusetts. 1997.
9. Murray, William H. and Pappas, Chris H.. MFC Programming in C++ with the Standard Template Libraries, Prentice Hall PTR, Upper Saddle River, NJ. 2000.
10. Perry, Jo Ellen and Levin, Harold D. An Introduction to Object-Oriented Design in C++, Addison Wesley Publishing Company, Reading, Massachusetts. 1996.
11. Savitch, Walter. Problem Solving with C++, Addison Wesley Publishing Company, Menlo Park, California. 1996.
12. Weiss, Mark Allen. Algorithms, Data Structures, and Problem Solving with C++, Addison Wesley Publishing Company, Menlo Park, California. 1996.

Course Analysis Questionnaire

A. Details of the Course

- A1 This course will be a required course for all Computer Science majors. Students from other departments may take this course as an elective if they have taken the prerequisite course.
- A2 This course will affect COSC310. Some material currently covered in COSC310 will be moved to this course. This will allow the remaining topics and additional topics to be covered in COSC310 in greater depth.
- A3 This course will be offered as a special topics course during the Spring and Fall 2000 terms.
- A4 This course is not currently intended to be a dual level course.
- A5 This course is not to be taken for variable credit.
- A6 Similar courses are offered at these institutions:
 - Murray College, Illinois
 - Oregon Institute of Technology, Oregon
 - Sonoma State University, California
 - Vrije Universiteit, Amsterdam, The Netherlands
- A7 This is not currently required by any accrediting authority. In recent years we have been incorporating this content in COSC310. It has resulted in an overloaded course.

B. Interdisciplinary Implications

- B1 Each section of this course will be taught by a single instructor.
- B2 This course does not overlap any courses offered in other departments.
- B3 Students from the School of Continuing education may register for this course.

C. Implementation

- C1 Faculty resources for teaching this course are currently adequate. The addition of this course will be accompanied by other changes in our department. These changes will accommodate the inclusion of this course.
- C2 The current classroom and laboratory space available on campus will be sufficient for this course. An object and GUI oriented programming environment (such as Microsoft Visual C++) will be provided in our departmental lab and in the open campus labs.
- C3 No resources for this course are being funded by grants.
- C4 This course will be offered each semester.
- C5 One or two sections of this course will be offered each semester.
- C6 The enrollment in this course will be limited to the number of computers available in our teaching lab.
- C7 No professional society recommends enrollment limits for this course.

D. Miscellaneous