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

RECEIVED
OCT 2 0 1995
LIBERAL STUDIES

UWUCC USE Only
Number: 95-43
Submission Date: _____
Action-Date: App 12/12/95
Senate App 2/6/96

CURRICULUM PROPOSAL COVER SHEET
University-Wide Undergraduate Curriculum Committee

I. CONTACT

Contact Person James L. Wolfe Phone 6104

Department Computer Science

II. PROPOSAL TYPE (Check All Appropriate Lines)

X **COURSE** Data Structures
Suggested 20 character title

____ New Course* _____
Course Number and Full Title

X Course Revision CO 310 Data Structures
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

X Course or Catalog Description Change CO 310 Data Structures
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)

_____ Department Curriculum Committee

Gary L Bertenthal Department Chair

_____ College Curriculum Committee

John D. Eck 10/20/95 College Dean

_____ +Director of Liberal Studies (where applicable)

_____ *Provost (where applicable)

Part II. Description of Curriculum Change

1. New Syllabus of Record

See Attachment A.

2. Summary of the proposed revision

The principal revision is to change the programming language used in the course from Pascal to C++, as well as to drop CO 220 as a prerequisite. The techniques and concepts taught, as well as the overall approach for the course, remains the same; only the language changes. There have been adjustments in the syllabus terminology to reflect the new language.

3. Justification for the revision

Object-oriented programming has for several years been widely touted as a very important programming and design approach. To keep up with the changing field of computer science, we must find methods of including the object-oriented approach in our degree programs. The programming language C++ was developed for use in object-oriented programming.

Pascal is a programming language that was developed specifically for teaching data types and data structures, but is not widely used in industry. In recent years, many universities, as well as some high schools, have changed to using C or C++ as the programming language they use to cover data structures. C and C++ have also become the worldwide choice for program development of nearly all types.

Currently, Pascal is taught intensively during the first few weeks of CO 310; the remainder of the course then uses Pascal as the vehicle for discussing data structures. By changing the language used in CO 310 to C++, we can catch up with a major programming trend in the computer field and incorporate the object-oriented approach into our curriculum. The object-oriented approach is an extension of the current material taught in CO 310. Time for inclusion of this approach will become available (without any loss of material) because students will not have to learn a new language in the revised CO 310 - they will have learned the basics of C++ in CO 110.

The removal of CO 220 as a prerequisite for CO 310 is designed to make CO 110 - CO 310 a two course sequence for Computer Science majors. This change will align our course sequence more closely with the ACM recommended curriculum. At the same time, removing the CO 220 prerequisite will insure that students coming into CO 310 will have a more consistent background than they do now.

Also, the alternate path of taking only CO 220 prior to CO 310 will no longer be reasonable for students - the revised CO 310 is structured such that it assumes that students already know the basics of C++. This would not be the case for students who have taken only CO 220 prior to CO 310.

4. Old Syllabus of Record

See Attachment B.

5. Letters of Support

See Attachment C.

NEW CO 310 Syllabus of Record Attachment A

I. Catalog Description

CO 310 Data Structures 3c-0l-3sh

Prerequisite: CO110
Basic concepts of data; storage systems and structures; lists, arrays,strings, hashing techniques, searching and sorting techniques; data structures in programming languages; string processing. Programming in an object-oriented language.

II. Course Objectives

Upon successful completion of this course, the students will be able to:

1. Analyze the time and space efficiency of several significant classes of algorithms.

2. Use abstract data structures, including stacks, queues, trees, graphs, hash tables, and linked lists, that are appropriate for a given algorithm.

3. Implement abstract data structures, objects and algorithms using an object-oriented approach and determine the effect of the implementation on the performance of the algorithm.

4. Use the object-oriented approach to program design and recognize its capabilities relative to the procedural paradigm.

III. Course Outline

1. Introduction to abstract data types and objects (6 hours)

   a. abstract data types

   b. encapsulation

   c. objects, classes, and inheritance

   d. polymorphism

   e. public and private attributes

f. passing arguments

2. Linear Data Structures (12 hours)

a. strings

b. array storage mapping functions

c. dynamic storage allocation - pointers

d. linked lists - singly and doubly linked lists

e. stacks and queues - as objects

f. circular lists and multi-lists

g. implementations of linear objects using several actual data structures

3. Sorting (6 hours)

a. elementary sorting techniques (exchange, selection, and insertion)

b. recursion as a programming technique

c. advanced sorting techniques (merge and quick)

d. algorithm efficiency analysis (big O notation)

4. Hierarchical Data Structures (9 hours)

a. general trees

b. binary trees

c. array and pointer implementations of binary trees

d. preorder, postorder, inorder traversals

e. special applications of binary trees (search trees, heaps)

f. heap sort

g. other trees (AVL, b-trees, 2-3-trees)

5. Graphs (3 hours)

6. Hash tables (3 hours)

IV. Evaluation Methods

		Suggested Grading Scale:	
Exam 1	100 points		
Exam 2	100 points	90-100%	A
Final	100 points	80-89%	B
Programming Projects	300 points	70-79%	C
Quizzes	~50 points	60-69%	D
Homework	~50 points	0-59%	F

V. Text

Frank M. Carrano, Data Structures and Problem Solving with C++, Walls and Mirrors, Benjamin Cummings Publishing Co. (1995)

VI. Special Resource Requirements

None - the needed resources are already in place.

VII. Bibliography

William Ford and William Topp, Data Structures with C++, Prentice Hall, 1995.

Joseph Bergin, Data Abstraction, McGraw-Hill, 1994.

James F. Korsh and Leonard J. Garrett, Data Structures, Algorithms, and Programming Style using C, PWS-Kent, 1988.

Johnsonbaugh, Richard, and Kalin, Martin. Object-Oriented Programming in C++. Prentice Hall. 1995.

Sengupta, Saumyendra, and Korobkin, Carl. C++: Object-Oriented Data Structures. Springer-Verlag. 1994.

OLD CO 310 Syllabus of Record Attachment B

I. Catalog Description

CO 310 Data Structures 3c-0l-3sh

Prerequisite: CO110 or CO220
Basic concepts of data; storage systems and structures; lists, arrays,strings, hashing techniques, searching and sorting techniques; data structures in programming languages; string processing. Programming in a block structured language.

II. Course Objectives

Upon successful completion of this course, the students will be able to:

1. Analyze the time and space efficiency of several significant classes of algorithms.

2. Design abstract data structures appropriate for a given algorithm.

3. Implement abstract data structures and algorithms using a block structured language and determine the effect of the implementation on the performance of the algorithm.

III. Course Outline

1. Introduction to a block structured Language (8 hours)

a. elements of the language

b. expressions

c. basic data structures - arrays records, strings

d. control structures

e. block structure and scoping

f. procedures and functions

g. Units and their application to abstraction

2. Linear Data Structures (13 hours)

a. Arrays - single, double, and N-dimensional arrays

b. array storage mapping functions

c. dynamic storage allocation - pointers

d. Linked Lists - singly and doubly linked lists

e. Stacks & Queues - as abstract data types

f. Circular lists and multi-lists

g. Implementations of physical linear data structures using several actual data structures

3. Sorting (9 hours)

a. elementary sorting techniques (exchange, selection, and insertion)

b. Recursion as a programming technique

c. advanced sorting techniques (merge and quick)

d. algorithm efficiency analysis (big O notation)

4. Hierarchical Data Structures (9 hours)

a. General Trees

b. Binary Trees

c. Array and pointer implementations of binary trees

d. Preorder, Postorder, Inorder traversals

e. Special applications of binary trees (search trees, heaps)

f. Heap sort

g. Other trees (AVL, b-trees, 2-3-trees)

4. Graphs (3 hours)

IV. Evaluation Methods

Exam 1	100 points
Exam 2	100 points
Final	100 points
Projects	300 points
Quizzes	~50 points
Homework	~50 points

V. Texts

Data Structures and Problem Solving with Turbo Pascal
Walls and Mirrors
Frank M. Carrano, Paul Helman, and Robert Verhoff
Benjamin Cummings Publishing Co. (1993)

Oh! Pascal! Turbo Pascal 6.0 (optional)
Doug Cooper
Norton Publishing Company

VI. Special Resource Requirements

386 (or faster) Personal Computers
(Borland) Turbo Pascal

VII. Bibliography

Fundementals of Data Structures in Pascal
Ellis Horowitz & Sartaj Sahni
Computer Science Press (1990)

Data Structures & Program Design
Robert L. Kruse
Prentice Hall Publishing (1987)

Data Structures with Abstract Data Types and Pascal
Daniel F. Stubbs & Neil W. Webre
Brooks/Cole Publishing (1989)

To: James Wolfe, Chairman
Computer Science Department Curriculum Committee

From: Gerald Buriok, Chairman
Mathematics Department GMB

Date: September 11, 1995

Subject: Computer Science Curriculum Changes

In your proposal of March 29, 1995, you outlined five changes the Computer Science Department wishes to implement in your curriculum. Since students majoring in our Mathematics and Applied Mathematics programs are required to take CO 110 Problem Solving and Structured Programming, and students in Applied Mathematics are required to take CO 250 Introduction to Numerical Methods, changes in these courses will directly affect our students. Other changes in your proposal may affect students in the Mathematics Department who are seeking a minor in Computer Science.

The faculty of the Mathematics Department support the five changes outlined in your proposal, and in particular, the changes you are recommending for CO 110 and CO 250. Using the computer programming language C++ in CO 110 while maintaining FORTRAN as the language of CO 250 means our students will be exposed to two programming languages instead of one, as is currently the case. This would appear to be an advantage to them. Likewise, since C++ will be used on personal computers while FORTRAN will still be taught on the VAX system, our students will have broader exposure to programming on different platforms. One potential drawback to this, however, is that students who have completed CO 220 prior to enrolling in CO 250 will have VAX experience, while those without CO 220 (such as Applied Mathematics majors) who have completed only CO 110 will have no prior VAX experience. It is my understanding that Computer Science faculty who teach CO 250 will address this concern.

From: GROVE::WHITSON 7-SEP-1995 15:10:45.50
To: JIM WOLFE
CC: GLBUTER
Subj: Dual level courses in Computer s

Jim:
The following is a memo that I am also sending as a hard copy to you:

Date: Thursday, September 7, 1995

To: Jim Wolfe,
Chair Computer Science Dept. Curriculum Committee

From: Dennis Whitson,
Chair, Physics Department

Re: Use of the C language in CO 110 and CO 310

This issue was brought up at a Physics Department meeting on September 5, 1995. The Physics Faculty were in favor of switching to the use of the C language in CO 110 if the non-object approach was used. It was felt that the Physics Student needs to be introduced to the general subject of programming in a way that emphasizes algorithm development and fosters capabilities comparable to the current course which uses FORTRAN. You should leave object oriented programming to CO 310, though introducing it in the last few weeks of CO 110 would not be objectionable.

The use of FORTRAN in CO 250, while using C in CO 110 would inevietably degrade CO 250 to some degree, but I think that the experience of having two computer languages would be of benefit to our students. This issue was not taken up at the above meeting, but I don't believe that there would be any significant objection to using either FORTRAN or C in CO 250.

The other issues in your projected curriculum changes are not of any real

Date: Thursday, September 7, 1995

To: Jim Wolfe,
Chair Computer Science Dept. Curriculum Committee

From: Dennis Whitson,
Chair, Physics Department

Re: Use of the C language in CO 110 and CO 310

This issue was brought up at a Physics Department meeting on September 5, 1995. The Physics Faculty were in favor of switching to the use of the C language in CO 110 if the non-object approach was used. It was felt that the Physics Student needs to be introduced to the general subject of programming in a way that emphasizes algorithm development and fosters capabilities comparable to the current course which uses FORTRAN. You should leave object oriented programming to CO 310, though introducing it in the last few weeks of CO 110 would not be objectionable.

The use of FORTRAN in CO 250, while using C in CO 110 would inevitably degrade CO 250 to some degree, but I think that the experience of having two computer languages would be of benefit to our students. This issue was not taken up at the above meeting, but I don't believe that there would be any significant objection to using either FORTRAN or C in CO 250.

The other issues in your projected curriculum changes are not of any real concern to the Physics Department.