LSC Use Only

UWUCC USE Only
95-42

Senate App 2/6/96

CURRICULUM PROPOSAL COVER SHEET

College Curri

Part II   Description of Curriculum Cha[illegible]

[illegible]

The principal revision is to change the programming la[illegible] [illegible] the same, only the language change[illegible]

There have been [illegible] in the syllabus terminology to reflect the new [illegible] age processin[illegible] is to [illegible]

[illegible]

FORTRAN is a programmin[illegible] language that is not being used as w[illegible] [illegible] mathematical applic[illegible] [illegible] ading, the use of C (and C++ a superset of C) [illegible]

[illegible] field of computer science, we must find methods [illegible]

[illegible]

By changing the language used in CO 110 [illegible] can catch up with a major programming trend in the co[illegible] [illegible] work for incorporating the object oriented approach into our curriculum. This approach will [illegible] in CO 310 [illegible] CO 310.

4. O[illegible] Syllabus of Re[illegible]

Part II. Description of Curriculum Change

2. Summary of the proposed revision

The principal revision is to change the programming language used in the course from FORTRAN to C++. 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 and adjustments in language processing to reflect the fact that C++ is to be taught on microcomputers rather than on the mainframe.

3. Justification for the revision

FORTRAN is a programming language that is not being used as widely as

NEW CO 110 Syllabus of Record Attachment A

I. Catalog Description

CO 110 Problem Solving and Structured Programming 3c-0l-3sh

(For science, mathematics, and computer science majors, and for others who have

problem analysis and computer solution using flowcharting and the C++ language.

III. Course Outline

A. Introduction 2 hrs

1. History of computers
2. Components of a computer
3. Programming languages
4. Compilation vs interpretation

B. The Programming Environment 4 hrs

1. Editing
2. Compiling

5. Debugging
6. Redirection of input and output

C. Algorithm development using flowcharts/psuedocode 4 hrs

1. Software engineering method
2. Classic problems - maximum, minimum, sum, average

D. Basic input and output 1 hr

E. Data types 4 hrs

1. Constants
2. Variables
3. Expressions
4. Library files and functions

F. Simple Data Structures 7 hrs

1. One dimensional arrays
2. Strings as arrays
3. Multi-dimensional arrays
4. Records

1. Single alternative
2. Double alternative
3. Multiple alternative
4. Nested structures

H. Loops 2 hrs

1. While loop

2. For loop

I. Programming language form 2 hrs

1. Syntax
2. Structured code
3. Documentation
4. Case sensitivity

[illegible]

primarily of multiple choice, true-false, and short answer questions.

30-35%	Programming assignments. There are approximately six programming assignments worth varying numbers of points that collectively count this portion of grade. Suggested tasks for the assignments include: linear search, sorting, 2-D array processing, interactive programming, simulation, sequential file processing, modularization.
10-15%	Class participation and quizzes. This may be based on written questions, verbal discussions, computer lab sessions, or other form of interaction.

Suggested Grading Scale:

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

V. Required textbook, supplemental books and readings

Adams, Joel, Leestma, Sanford, and Nyhoff, Larry. C++: An Introduction to Computing. Prentice Hall. 1995.

VI. Special resource requirements

None.

VII. Bibliography

Staugaard, Andrew C., Jr. Structuring Techniques: An Introduction Using Turbo C++.

Perry, Greg. Turbo C++ Programming 101. Sams Publishing. 1993.

OLD CO 110 Syllabus of Record Attachment B

I. Catalog Description

CO 110 Problem Solving and Structured Programming 3c-01-3sh

(For science, mathematics, and computer science majors, and for others who have a sufficiently quantitative orientation.) Basic structure of modern digital computers; batch processing vs. interactive time-shared online computing; problem analysis and computer solution using flowcharting and the FORTRAN language. Exemption or credit by examination possible.

II. Course Objectives

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

A. Use the computing environment at IUP, including appropriate command language and editor(s).

B. Develop algorithms from user problem statements.

C. Express the solutions to computer-oriented problems in flowchart

and/or pseudocode.

D. Give commands to compile, link, and run their own programs, including using common options.

E. Proficiently transform designs of problem solutions into programming language.

F. Apply debugging and testing techniques to locate errors and determine the effectiveness of a program.

III. Course Outline

A. Introduction 2 hrs

1. History of computers
2. Components of a computer
3. Programming languages
4. Compilation vs interpretation

B. The Computing Environment 6 hrs

1. Logging in, facilities
2. Editing
3. Command Language
4. Compilation and linking, with options
5. Command file and batch printouts
6. Introduction to the debugger

C. Algorithm development using flowcharts/psuedocode 4 hrs

1. Software engineering method
2. Classic problems - maximum, minimum, sum, average

D. Basic input and output 1 hr

E. Data types 3 hrs

1. Constants
2. Variables
3. Expressions

F. Arrays 6 hrs

1. One-dimensional
2. Array I/O
3. Multi-dimensional
4. Classic problems - searching, sorting

3. Multiple alternative

H. Loops 3 hrs

1. While loop
2. Repeat-until loop
3. Counting loop
4. Priming read loop

I. Programming language form 2 hrs

1. Syntax
2. Structured code
3. Documentation

J. Advanced formatted I/O including file access 5 hrs

3. Sequential file processing

K. Testing and Debugging 1 hr

L. Subprograms 3 hrs

1. Functions
2. Subroutines
3. Argument passing

assignments include: linear search, sorting, 2-D array processing, interactive programming, simulation, sequential file processing, modularization. (180 points)

10-15% Class participation and quizzes. This may be based on written

Letters of Support

Attachment C

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

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

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

Date: Thursday, September 7, 1995

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

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