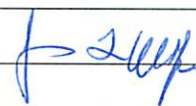
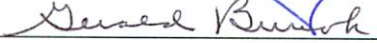
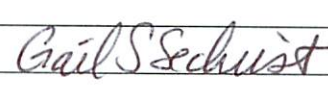


LSC Use Only No: 07-16m	LSC Action-Date:	UWUCC USE Only No.	UWUCC Action-Date: App-10/30/07	Senate Action Date: App-12/4/07
-------------------------	------------------	--------------------	---------------------------------	---------------------------------

Curriculum Proposal Cover Sheet - University-Wide Undergraduate Curriculum Committee

Contact Person SANWAR ALI	Email Address SANWAR
Proposing Department/Unit COMPUTER SCIENCE	Phone 7-7994

Check all appropriate lines and complete information as requested. Use a separate cover sheet for each course proposal and for each program proposal.

1. Course Proposals (check all that apply) ☐ New Course ☐ Course Prefix Change ☐ Course Deletion ☒ Course Revision ☐ Course Number and/or Title Change ☒ Catalog Description Change		
COSC 420 Modern Programming Languages		<u>Proposed</u> course prefix, number and full title, if changing
<u>Current</u> Course prefix, number and full title		
2. Additional Course Designations: check if appropriate ☐ This course is also proposed as a Liberal Studies Course. ☐ Other: (e.g., Women's Studies, Pan-African) ☐ This course is also proposed as an Honors College Course.		
3. Program Proposals ☐ New Degree Program ☐ Program Title Change ☐ Program Revision ☐ New Minor Program ☐ New Track ☐ Other		
<u>Current</u> program name		<u>Proposed</u> program name, if changing
4. Approvals		
Department Curriculum Committee Chair(s)		5 Dec 06
Department Chair(s)		12/7/06
College Curriculum Committee Chair		05/17/07
College Dean		9/24/07
Director of Liberal Studies *		
Director of Honors College *		
Provost *		
Additional signatures as appropriate: (include title)		
UWUCC Co-Chairs		10/30/07

* where applicable

Received

SEP 25 2007

Liberal Studies

Part-II Description of the Curriculum Change

1. New Syllabus of Record

I. Catalog Description

COSC 420 Modern Programming Languages

3c-0l-3cr

Prerequisite: COSC 310

A comparative survey of programming language paradigms. Includes an examination of the properties, applications, syntax and semantics of selected object-oriented, functional and declarative programming languages.

II. Course Outcomes

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

- analyze the syntax, semantics, type handling, coercion rules, scoping rules, naming rules, and exception handling of several programming languages
- interpret BNF, syntax diagrams, parse trees, context-free grammars and other forms to syntax and semantics of a programming language
- distinguish the lexical and syntax analysis of a language
- perceive the differences between imperative, object-oriented, functional, aspect-oriented, and declarative languages
- recognize the types of applications where imperative, object-oriented, functional, aspect-oriented, and declarative languages will be best suited
- demonstrate the strengths and weaknesses of different programming languages

III. Detailed Course Outline

- | | |
|--|------------------|
| 1. Programming Languages- preliminaries, history, and evolution | 0.5 week |
| 2. Describing Syntax and Semantics | 1.5 weeks |
| <ul style="list-style-type: none">- Lexemes, tokens- Language recognizers- Language generators- Context-free grammars- Backus-Naur Form (BNF)- Extended BNF- Syntax diagrams- Parse trees- Ambiguity- Unambiguous grammar- Attribute grammar | |
| 3. Lexical and Syntax Analysis | 1.0 week |
| <ul style="list-style-type: none">- Lexical analysis- Parsing problem- Top-down parsing- Bottom-up parsing- Recursive-descent parsing- LL and LR parsers | |
| 4. Names, Bindings, Type Checking, and Scopes | 1.0 weeks |

5. Data Types	1.0 week
TEST-I	one class period
6. Functional Languages, such as ML and Lisp	3.0 weeks
7. Object-Oriented (OO) Language, such as SmallTalk with comparison to other OO-languages	2.0 weeks
8. Annotation and Aspect-oriented programming	1.0 week
TEST-II	one class period
9. Declarative Language, such as Prolog	2.0 weeks
10. Exception Handling	1.0 week
Total =	14 weeks
FINAL EXAM	During Final Exam Week

IV. Evaluation Methods:

The final grade of the course will be determined as follows:

Two Class Tests	30%
Comprehensive Final Exam	25%
Language Projects	25%
Assignments	20%

Sample written projects related to languages that students must do in this course:

- SmallTalk and its difference from other OOP Languages, such as JAVA and Visual Basic
- Ada 95- Object-Oriented Programming Language
- Fortran 90 and 95 and their differences
- Differences between functional languages ML, Miranda, and Haskell
- Perl- Programming Language
- Eiffel- Programming Language
- Differences between scripting languages, such as Java Script, PHP, and Python
- Or any other interesting programming languages (upon instructor's approval)

V. Example Grading Scale:

The grading scale will be:

90-100% = A, 80-89% = B, 70-79% = C, 60-69% = D, and < 60% = F.

VI. Undergraduate Course Attendance policy:

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

VII. Required Textbook, Supplemental Books and Readings:

Sebastia, Robert W. *Concepts of Programming Languages* (7/e), Addison Wesley, 2006, ISBN #0-321-33025-0

Webber, Adam B. *Modern Programming Languages: A Practical Introduction*, Franklin, Beedle & Associates, 2003, ISBN #1-887902-76-7

VIII. Special Resource Requirements:

None

IX. Bibliography:

Pratt, Terrence W. and Zelkowitz, Marvin V. *Programming Languages: Design and Implementation* (4/e), Prentice Hall, 2001, ISBN #0-13-027678-2

Sethi, Ravi *Programming languages: Concepts and Constructs* (2/e), Addison-Wesley, 1996, ISBN #0-201-59065-4

Tucker, Allen and Noonan, Robert *Programming Languages: Principles and Paradigms*, McGraw Hill, 2007, ISBN #0-07-286609-8

Wilson, Leslie B. and Clark, Robert G. *Comparative Programming Languages* (3/e), Addison-Wesley, 2001, ISBN #0-201-71012-9

2. Summary of the proposed revisions

Since no previous syllabus of record is found for this course, the new syllabus of record reflects the proposed revisions of this course.

3. Justification/rationale for the revision

Computer Science is a rapidly changing field. During recent years many new high-level programming languages have been developed. Each language has its unique applications in real world problems. This course must include the comparative studies of several modern programming languages. Also the department could not find any old syllabus of record. Hence, it is required that we make a new syllabus of record because it is a required course for Language and System track (LAS) to meet the ABET criteria.

Change of Prerequisite: The contents of the new syllabus depend on materials taught in its prerequisite course, COSC 310 Data Structure and Algorithms, not on COSC 220 Applied Computer Programming, which uses the COBOL language. COBOL is not a required topic for ABET.

Note: All though the textbook *Concepts of Programming Languages* by Robert W. Sebasta uses the word "concepts", it is a high standard textbook for senior level course.

4. The old syllabus of record

Does not exist. We are applying under the provision of Syllabus of Record Amnesty for expedited review.

Part-III Letters of Support or Acknowledgement

N/A