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

Curriculum Proposal Cover Sheet - University-Wide Undergraduate Curriculum Committee

Contact Person SOUNDARARAJAN EZEKIEL	Email Address EZEKIEL
Proposing Department/Unit COMPUTER SCIENCE	Phone 7-6102

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

__X__Course Revision _____Course Number and/or Title Change _____Catalog Description Change

COSC 460 THEORY OF COMPUTATION

***Current** Course prefix, number and full title* ***Proposed** course prefix, number and full title, if changing*

2. Additional Course Designations: check if appropriate

_____ This course is also proposed as a Liberal Studies Course.

_____ This course is also proposed as an Honors College Course.

_______ Other: (e.g., Women's Studies, Pan-African)

3. Program Proposals

_____Catalog Description Change _____Program Revision

_____New Degree Program _____Program Title Change _____Other

_____New Minor Program _____New Track

***Current** program name* ***Proposed** program name, if changing*

4. Approvals		**Date**
Department Curriculum Committee Chair(s)		5 Dec 06
Department Chair(s)	Wm. Oblitey	12/7/06
College Curriculum Committee Chair		05/17/07
College Dean	Gerald Buriok	9/24/07
Director of Liberal Studies *		
Director of Honors College *		
Provost *		
Additional signatures as appropriate: (include title)		
UWUCC Co-Chairs	Gail Sechrist	10/30/07

* where applicable

Part-II **Description of the Curriculum Change**

1. New Syllabus of Record

I. Catalog Description

COSC 460 Theory of Computation 3c-01-3cr

Prerequisite: COSC 310 or instructor permission
Formal methods for describing and analyzing programming languages and algorithms. Backus-Naur forms; productions; regular expressions; introduction to automata theory; Turing machines; recent concepts in algorithm theory computability.

II. Course Outcomes

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

- Define languages by abstract, recursive definitions and by regular expressions.
- Design a finite automaton to recognize a given regular language.
- Transform a language into regular expression or finite automaton or transition graph.
- Define deterministic and nondeterministic finite automata.
- Prove properties of regular languages and classify them.
- Determine decidability, finiteness and equivalence properties.
- Define relationship between regular languages and context-free grammars.
- Building a context-free grammar for pushdown automata.
- Determine whether a given language is context-free language or not.
- Prove properties of context-free languages.
- Design Turing machine and Post machine for a given language.
- Discuss the concept of computability.

III. Detail Course Outline
Part I: AUTOMATA THEORY

A. Languages 3 hours
   a. Languages in the abstract
   b. Kleene closure
   c. Recursive definition
   d. Arithmetic expression—language
   e. Defining languages by regular expression
   f. Languages associated with regular expression
   g. Finite languages

B. Finite Automata 2 hours
   a. Defining languages by Finite Automata
   b. Finite Automata and their languages
   c. EVEN-EVEN languages

C. Transition Graph 4 hours
   a. Defining transition graphs
   b. Generalized transition graphs
   c. Nondeterminism
   d. Unification
   e. Turning transition graphs into regular expressions
   f. Converting regular expressions into Finite Automata

D. Finite Automata with Output 3 hours
   a. Moore machines
   b. Mealy machines
   c. Moore = = Mealy
   d. Examples

E. Regular languages 3 hours
   a. Closure properties

   b. Complement and Intersection
   c. Pumping Lemma

Part II: THEORY OF FORMAL LANGUAGES

F. Context-free Grammars	5 hours
a. Define languages	
b. Parse trees	
c. The Total Language Tree of the CFG	
d. Regular grammar	
e. Ambiguity	
f. Chomsky Normal Form	
g. Derivations	
G. Pushdown Automata	5 hours
a. Building a PDA for a CFG	
b. Building a CFG for a PDA	
H. Context-free languages	5 hours
a. Self embeddedness	
b. Pumping lemma	
c. Closure properties	
d. Intersection and complement	
e. Finiteness, emptiness and membership	

Part III: THEORY OF TURING MACHINES

I. Turing Machines	4 hours
a. Define Turing machine	
b. Post Machines	
c. Simulating PM on a TM	
d. Simulating TM on PM	
J. Variations of Turing Machines	3 hours
a. The k-track TM	
b. Recursively enumerable languages	
c. Universal TM	
d. Decidability	
K. The Chomsky Hierarchy and Computers	3 hours
a. Phrase structure grammar	
b. Type 0=TM	
c. Defining the computer	
d. Church thesis	
Midterm Exams	2 hours
Total =	**42 hours**
Final Exam	**During Final Exam Week**

IV. Evaluation Methods

The final grade for the course is determined as follows:

Two class tests	30%
Final Exam	30%
Homework and Projects	25%
Quizzes	15%

V. Grading Scale: 90-100% A, 80-89% B, 70-79% C, 60-69% D, 0-59% F

VI. Attendance:-

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

VII. Required Textbook

Introduction to Computer theory, Second Edition, Daniel I.A. Cohen., John Wiley & Sons, Inc., New York, 1997

VIII. Special Resource Requirements

None.

IX. Bibliography

Davis, Ruth and E. Truth, Deduction, and Computation. New York: Computer Science Press, 1989.

Drobot, V. Formal Languages and Automata Theory. Rockville, MD: Computer Science Press, 1989.

Dougherty, E. R. and Giardina, C. R. Mathematical Methods for Artificial Intelligence and Autonomous Systems. Englewood Cliffs, NJ: Prentice Hall, 1988.

Floyd, R. W. and Beigel, R. The Languages of Machines: An Introduction to Computability and Formal Languages. New

Freivalds, Rusins, Fundamentals of Computation Theory, Springer, 2001.

York: Computer Science Press, 1994.

Gregory, Taylor R., Models of Computation, Oxford University Press, 1998.

Gurari, E. M. An Introduction to the Theory of Computation. New York: Computer Science Press,1989.

Hein, James L., Theory of Computation, Jones & Bartlett, 1996.

Hennessy, M. Algebraic Theory of Processes. Cambridge, MA: MIT Press 1988.

Hopcroft, John E., and Ullman, Jeffery D., Introduction to Automata Theory, Languages, and Computation Second Edition, Addison-Wesley, 2001.

Lewis H., and Papadimitriou, C., Elements of the Theory of Computation, Prentice Hall, 1981.

Lingas, Andrzej and Nilsson, Bengt J., Fundamentals of Computation Theory, Springer, 2003.

Linz, Peter, An Introduction to Formal Languages and Automata, Jones and Bartlett Publishers, 2001.

Manna, Zohar, Mathematical Theory of Computation, Courier Dover Publications, 2003.

Martin, John C., Introduction to Languages and the Theory of Computation, McGraw-Hill, 2002.

Moll, R. N., Arbib, M. A. & Kfoury, A. J. An Introduction to Formal Language Theory. New York: Springer-Verlag, 1988.

Papadimitriou C., Computational Complexity, Addison-Wesley, 1994.

Rogers, Hartley, Jr, Theory of Recursive Functions and Effective Computability, MIT Press, 1987.

Sipser, Michael, Introduction to the Theory of Computation Second Edition, PWS Publishing, 2006.

2. Summary of the proposed revisions

N/A

3. Justification/rationale for the revision

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.

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**

Not applicable