September 13, 2010

METROPOLITAN STATE COLLEGE *of* DENVER
Office of Academic Affairs

# REGULAR COURSE SYLLABUS

**School of:** Letters, Arts, and Sciences

**Department:** Mathematical and Computer Sciences

**CIP Code:** 11.0701

**Prefix & Course Number:** CS 3240 **Crosslisted With*:** _____

**Course Title:** Introduction to the Theory of Computation

**Check All That Apply:** Required for Major: X Required for Minor: X Specified Elective: X
Required for Concentration: X Elective: X Service Course: _____

**Credit Hours:** 2 ( 2 + 0 )

**Total Contact Hours per semester (assuming 15-16 week semester):**

Lecture 30 Lab 0 Internship 0 Practicum 0 Other (please specify type and hours): 0

**Schedule Type(s):** Lecture **Grading Mode(s):** Letter

**Variable Topics Courses** (list restrictions, including the maximum number of hours that can be earned**):

_____

**** NOTE: This information must be included in the course description.**

**Restrictions (Variable Topics Course):** _____

**Prerequisite(s):** CS 2050 and MTH 3170 each with a grade of "C" or better, or permission of instructor.

**Corequisite(s):** none

**Prerequisite(s) or Corequisite(s):** _____

**Banner Enforced:**

**Prerequisite(s):** _____
**Corequisite(s):** _____
**Prerequisite(s) or Corequisite(s):** _____

**Catalog Course Description:**
This course explores language theory and computability. Language theory includes: regular expressions, regular languages, and finite automata (deterministic and non-deterministic); context-free languages and pushdown automata; and language grammars. Computability includes: Turing machines and their computing power; unsolvable problems; and intractable problems (NP-Completeness).

APPROVED:

| | |
|---|---|
| Department Curriculum Committee | Date 10/14/10 |
| Department Chair **OR** Program Director | Date 10/14/10 |
| Dean **OR** Associate Dean | Date 11/12/10 |
| Associate VP, Academic Affairs | Date 3/4/11 |

*If crosslisted, attach completed Course Crosslisting Agreement Form

Prefix and Course Number: CS 3240 September 13, 2010

**Required Reading and Other Materials will be equivalent to:**
*Introduction to Theory of Computation*, by Michael Sipser, Course Technology, 2006,
ISBN: 0-534-95097-3

**Specific, *Measurable* Student Behavioral Learning Objectives:**
Upon completion of this course the student should be able to

1. Determine the language represented by a regular expression
2. Create a regular expression from a language description
3. Construct a deterministic finite automaton accepting a given language
4. For a given finite automaton (deterministic or nondeterministic) determine if a string is accepted
5. Draw a state diagram for a finite automaton (deterministic or nondeterministic) that accepts a given language
6. For a given finite automaton (deterministic or nondeterministic) find a minimum-state equivalent deterministic finite automaton
7. Use a grammar to generate or accept a string
8. For a given grammar give the derivation(s) of a specified string and draw the corresponding parse tree(s)
9. Construct a context-free grammar for a language
10. Construct a pushdown automaton that accepts a given language
11. Show that a specific language is context-free
12. Trace the computation for a specified Turing machine
13. Create a Turing machine to solve a specified problem
14. Determine whether a problem about Turing machines is solvable or undecidable
15. Discuss the Chomsky hierarchy and arrange classes of languages in order of increasing generality
16. Discuss relative computational power of language recognizers (e.g., pushdown automata, regular expressions)

**Detailed Outline of Course Content** (Major Topics and Subtopics):

I. Sets and Relations
   A. Sets, relations, and functions
   B. Closures and algorithms
II. Languages
   A. Alphabets and languages
   B. Regular expressions
III. Regular Languages
   A. Deterministic finite automata
   B. Nondeterministic finite automata
   C. Languages that are and are not regular
   D. State minimization
   E. Algorithmic aspects of finite automata
IV. Context-free Languages
   A. Context-free grammars and languages
   B. Parse trees
   C. Pushdown automata
   D. Determinism and parsing
V. Programming Languages and Language Theory
   A. Compilation issues
VI. Turing Machines
   A. Computing with Turing machines
   B. Extensions of Turing machines – random access, nondeterministic, 2 tapes, etc.
   C. Universal Turing machines
   D. Grammars

VII. Undecidability
- A. The Church-Turing thesis
- B. The halting problem
- C. Unsolvable problems about Turing machines
- D. Unsolvable problems about grammars
- E. Properties of recursive languages

VIII. Introduction to Computational Complexity
- A. P versus NP

**Evaluation of Student Performance**

A combination of the following:

1. Homework and Programming Assignments
2. Quizzes and Examinations
3. Final Examination