Algorithm Engineering Syllabus
Instructor: Dr. David W. Matula
(Last updated 8/25/2016 by Zizhen Chen)

**Catalog Description:** Methods for evaluating algorithm efficiency, data type specification, numeric representation and data structure implementation, algorithm design paradigms, fundamental algorithm case studies in sorting and searching, arithmetic and matrix computation, graphs and networks, bio informatics, computational geometry, introduction to problem complexity, certificates and verification, survey of NP-complete problems. Reduction to practice: term project to design, test and validate, illustrate and display results, and measure efficiency of an algorithm implementation.

**Prerequisites:** CSE3353 (For non-majors: fluency in discrete mathematics and data structures)


**Reference:** Finite Precision Number Systems and Arithmetic by Kornerup and Matula, 2010

**Course Topics:**

I. Measuring Algorithm Efficiency (9 classroom hours):
Implementation independent measurement of algorithm efficiency, time and space resources, growth in terms of input size, polynomial vs. exponential growth algorithms, worst and average case efficiency, big Oh notation, algorithm efficiency vs. inherent problem (any algorithm) complexity, certificates, verification algorithms, decision trees, table lookup, popular algorithm notations, deterministic and nondeterministic algorithms, algorithm analysis techniques, induction, recurrence equations, amortization, standards and implementation dependent resource measurement.

II. Data Type Specification and Data Structure Implementation (9 classroom hours):
Tools for algorithm design, abstract data types, selecting data structures, stacks, queues, priority queues, trees, heaps, ROM lookup tables/trees, hash tables, radix, residue and rational number representation, arrays and linked structures, data structure conversion and compression, data structure search and traversal, binary search, balanced data structures.

III. Algorithm Design Paradigms (18 classroom hours):
Characterization of algorithm design paradigms, greedy, divide-and-conquer, dynamic programming, backtracking, branch-and-bound, utilization of design paradigms for problems across application areas of sorting, selection, computer arithmetic and algebraic computation, graphs and networks, bio informatics, computational geometry.

IV. Algorithm Implementation Project (6 classroom hours):
Project description, specifying computational environments, data structure and algorithm selection, test problem design, walkthrough, illustrations, implementation validation, measuring implementation efficiency, result display.

V. Course Grading Distribution:

Exams – Quiz, Midterm and Final ≈ 500
Homework – 5 sets ≈ 450
Project – Algorithm Development and Implementation ≈ 400
Total ≈ 1350