CSE 2010 Algorithms and Data Structures (4 credits)

Primary instructor: Ronaldo Menezes **Supporting faculty:** Ryan Stansifer

Textbooks and references:

R. Sedgewick and K. Wayne, Algorithms. Addison-Wesley, 2011. (T)

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms. MIT Press, 2009. (R)

Course information:

2014–2015 Catalog description: CSE 2010 Algorithms and Data Structures (4 credits). Expands CSE 1002 to include algorithms and data structures fundamental to software systems development. Includes abstraction, recursion, algorithm design and complexity analysis, linked lists, stacks, queues, trees, and sorting and searching methods. (CL) Prerequisites: CSE 1002, CSE 1400 or MTH 2051.

Prerequisites by topic: Object-oriented program design concepts, Java Collections for dynamic data structures, Java methods for parsing, formatting and converting data, recursion, basic performance analysis, exception handling

Place in program: Required, grade of C or better. Prerequisite for: CSE 2410, 4081, 4083, 4101, 4520, and all advanced electives.

Course outcomes & related student outcomes: The student will be able to

1. Demonstrate and use algorithmic paradigms (including brute force, divide-and-conquer, greedy, dynamic programming). (1: Fundamental knowledge)
2. Explain efficiency measures (average and worst cases), rates of growth, and asymptotic behavior. (1: Fuudamental knowledge)
3. Use data processing algorithms (sorting, searching, hash table, etc.). (1: Scientific, computing, aud engiueering problem solving & 4a: Skillful software construction)
4. Explain graph theory and graph theoretic algorithms (shortest paths, spanning trees, etc.). (1: Fundamental knowledge & 2: Scientific, computing, and engineering problem solviug)
5. Employ recursiou. (2: Scientific, computing, and engineering problem solving)
6. Describe, implemeut and use basic data structures: stacks, queues, linked lists, trees, graphs, etc., iu the developmeut of programs. (1: Fundamental knowledge & 4a: Skillful software construction)

Topics covered:

1. Abstract data types: linked lists, stacks, queues, trees, graphs, sets, etc. (10 hours)
2. Sorting algorithms: bubblesort, insertion sort, selection sort, quicksort, etc. (5 hours)
3. Searching: brute force, hashing, etc. (7 hours)

4. Time and space complexity (10 hours)
5. Recursion in programming (5 hours)
6. Dynamic programming (4 hours)

Approved by: Ronaldo Menezes, Associate Professor & Ryan Stansifer, Associate Professor and Director of Computer Science Programs

Signature: _____ **Date:** 2/10/15

Signature: Ryan Stansifer **Date:** 10 Feb 2015