COURSE SYLLABUS

******************************************************************************

YEAR COURSE OFFERED: 2015

SEMESTER COURSE OFFERED: Fall

DEPARTMENT: Computer Science

COURSE NUMBER: COSC2320

NAME OF COURSE: Data Structures and Algorithms

NAME OF INSTRUCTOR: Carlos Ordonez

******************************************************************************

The information contained in this class syllabus is subject to change without notice. Students are expected to be aware of any additional course policies presented by the instructor during the course.

******************************************************************************

Learning Objectives

Students will gain a good understanding of both theory and programming of fundamental data structures and algorithms. This course will provide a foundation for more advanced courses in computer science. Students will learn how data structures help solving many problems in all areas in computer science, including compilers, database systems and operating systems. Students will understand how to derive a function that quantifies the efficiency of an algorithm and will be able to understand how to analytically prove a program works correctly. Finally, students will become experienced C++ programmers, understanding how to develop correct and robust code and how to test it.

Major Assignments/Exams

This is a course that gives more weight to homeworks. Grading is as follows:

- 70%: 7 programming assignments,
- 30%: Midterm,

Required Reading


Recommended Reading

Any CS book on C++ (I recommend Stroustrup's) and discrete mathematics are helpful for additional reference. A good reference for the theory part is the following book:

**List of discussion/lecture topics**

Topics include the following:

1. Basics of C++: pointers, arrays, parameter passing, class encapsulation, inheritance, methods, function overloading, I/O
2. Abstract Data Types (ADTs) and OOP
3. linked lists, queues,
4. stacks, recursion
5. trees, balanced trees, multi-way search trees
6. hashing functions and hash tables,
7. Sorting and search algorithms: selection sort, heap sort, quick sort, shell sort, merge sort, linear and binary search, hash-based search.
8. graphs, graph traversal, search, path problems, transitive closure

Theory covered in the course:

1. time complexity analysis
2. algorithm design techniques
3. recursive functions
4. proving algorithm correctness