COP 3530 – Data Structures and Algorithm Analysis

Credits: 3 credits

Text book, title, author, and year: *Larry Nyhoff, ADT, Data Structures and Problem Solving with C++, Pearson Prentice Hall, 2005*

a. **Supplemental materials:** none.

Specific course information

a. **Catalog description:** The design, implementation and run-time analysis of important data structures and algorithms. The data structures considered include sorted arrays, linked lists, stacks, queues, and trees. An approach based on abstract data types and classes will be emphasized. The use of recursion for algorithm design. Class design and implementation in C++. Programming assignments in the C++ language.
b. **Prerequisites:** COP 3014
c. **Corequisites:** MAD 2104
d. **Required, elective, or selected elective:** Required

Specific goals for the course

a. **Specific outcomes of instruction:** By the end of the course students will be able to: (i) Demonstrate the ability to produce correct code; (ii) Demonstrate the ability to produce clear and well-structured code; (iii) Demonstrate the ability to produce code that is space and time efficient; (iv) Demonstrate the ability to choose and implement data structures; (v) Demonstrate understanding of the entire software life cycle including design, implementation, testing, maintenance, and documentation;(vi) Demonstrate understanding of the overall structure of an operating system, and the data structures and programming constructs used in operating systems.

Brief list of topics to be covered:

a. Principles of Programming
b. Review of C++ concepts
c. Recursion (fundamentals)
d. Data abstraction: Abstract Data Types, C and C++ classes
e. Linked lists: singly-linked, circular, dummy header, doubly-linked
f. Pointers and dynamic allocation
g. Stacks: Stack ADT, various implementations, applications
h. Queues: Queue ADT, various implementations, applications
i. C++ Classes
j. Inheritance and Object-oriented Design
k. Virtual functions
l. Template classes
m. Operator overloading
n. Algorithm efficiency: growth rates and big-O notation
o. Sorting: comparison of various algorithms
p. Trees: Binary Tree ADT, binary search tree ADT, implementation and applications

q. Graph ADT, implementation, DFS, BFS.
r. Hash Table ADT
s. Priority Queue ADT, heaps, heap-sort