Course number and name: **CS 04222: Data Structures and Algorithms**

Credits and contact hours: 4 credits / 5 contact hours

Instructor’s or course coordinator’s name: Seth Bergmann


Specific course information

**Catalog description:** This course features programs of realistic complexity. The programs utilize data structures (string, lists, graphs, stacks, trees) and algorithms (searching, sorting, etc.) for manipulating these data structures. The course emphasizes interactive design and includes the use of microcomputer systems and direct access data files.

**Prerequisites:** CS 04114 Object Oriented Programming and Data Abstraction and (MATH 03160 Discrete Structures or MATH 03150 Discrete Mathematics)

**Type of Course:** ☒ Required ☐ Elective ☐ Selected Elective

Specific goals for the course

1. **big O algorithm analysis.** Students have produced an informal big O run time analysis of searching and sorting algorithms
   - ABET (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

2. **big O data structure analysis.** Students have produced an informal big O run time analysis of various methods relating to the data structures (e.g., lists, stacks, queues, maps)
   - ABET (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

3. **data structure selection.** Students have demonstrated the ability to make decisions on the proper use of the data structures (e.g., lists, stacks, queues, maps) for time and space criteria and discuss the advantages and disadvantages of different data structures/formats.
   - ABET (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

4. **primitive data structures.** Students have demonstrated the ability to construct, implement and utilize common data structures fusing three primitive structures: arrays, integers, and references and are able to list the most common structures and data formats for storing data in
a computer system

- ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

5. **search and sort.** Students have coded and tested searching and sorting algorithms

- ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

Required list of topics to be covered

1. Data structures
   a. Lists
      i. Array-based
   b. Stacks
   c. Queues:
      i. Priority queues
   d. Trees:
      i. Binary Search Trees
   e. Heaps
   f. Hash Tables
   g. Sets
      i. TreeSet
      ii. HashSet
   h. Maps
      i. TreeMap
      ii. HashMap

2. Algorithms
   a. Sorting
      i. Selection Sort
      ii. Bubble Sort
      iii. Quick Sort
      iv. Heap Sort
   b. Searching
      i. Sequential Search
      ii. Binary Search
      iii. Hash Tables
Optional list of topics that could be covered

3. Backtracking algorithms (Eight Queens Problem)
4. Balanced Binary Search Trees (AVL trees)
5. Dequeues
6. Introduction to NP-complete problems
7. Data structures
   a. Lists - Link-based (simply, doubly, circular doubly-linked)
   b. Queues:
      i. ArrayList-based queues
      ii. LinkedList-based queues
      iii. Array-based (including array used in circular fashion)
      iv. Simply-Linked-structure based
   c. Trees - Expression Trees
   d. Iterators
      i. Lists: Iterator and ListIterator
      ii. Tree Iterators (Traversals)
      iii. Hash Table Iterators
      iv. Set Iterators
   e. Directed Graphs – Path algorithm (non-optimal)
8. Algorithm
   a. Sorting
      i. Bubble Sort - Applied to LinkedList
      ii. MergeSort
      iii. TreeSort
   b. Searching – Graph Path (non-optimal)
9. Algorithm Analysis
   a. Iterative
   b. Recursive
10. Recursion