CSCI 2210

Data Structures

Credit Hours: 4

Contact Hours: 4

Course Coordinator: Don Bailes

Text(s):


Catalog Description:

Strings, vectors, lists, stacks, queues, arrays, trees, hash tables and association containers, algorithm and elementary analysis. Laboratory use of the computer in designing, coding, debugging, and executing programs is an integral part of the course.

Prerequisite(s): CSCI 1260, CSCI 1900, and CSCI 2020

CS: REQUIRED

IS: MAJOR ELECTIVE

IT: MAJOR ELECTIVE

Course Outcomes:

Make appropriate data structure and algorithm design decisions with respect to program size, execution speed, and storage efficiency - ETSU Outcomes 4a, CS-2; ABET Outcomes c, CS-j

Understand common data structures (such as arrays, lists, linked lists, stacks, queues, priority queues, trees, hash tables, associative containers) and the algorithms that build and manipulate them including various sorting, searching, and hashing algorithms - ETSU Outcomes 4a, 4b, 5c, CS-2, IS-1b; ABET Outcomes a, b, c, i

Use appropriate classes and algorithms found in a common library - ETSU Outcomes 5c, CS-2; ABET Outcome i

Do basic work with random numbers, simulation and modeling, and string processing - ETSU Outcomes 4a, CS-1, CS-2; ABET Outcome a

And use basic algorithm analysis tools including both theoretical and empirical methods such as BigOh, Little-Oh, Big Omega, Big Theta, and execution profiling tools - ETSU Outcomes 4a, CS-1; ABET Outcome a
**Major Topics:**

C# and .NET. Syntax, I/O, properties, operator overloading, GUI; Use of a modern IDE with intellisense, snippets, refactoring, and other tools; debugging tools

Generics; type-casting; exception handling; inheritance; strings and regular expressions; Enumerators and interfaces

Common data structures that have existing implementations in the .NET library and some we must build ourselves. Determining which data structure is appropriate in a given situation

Recursion. Sorting, searching, and other algorithms.

Performance issues for common data structures and algorithms. Use of an execution profiling tool and algorithm analysis with big-Oh notation.