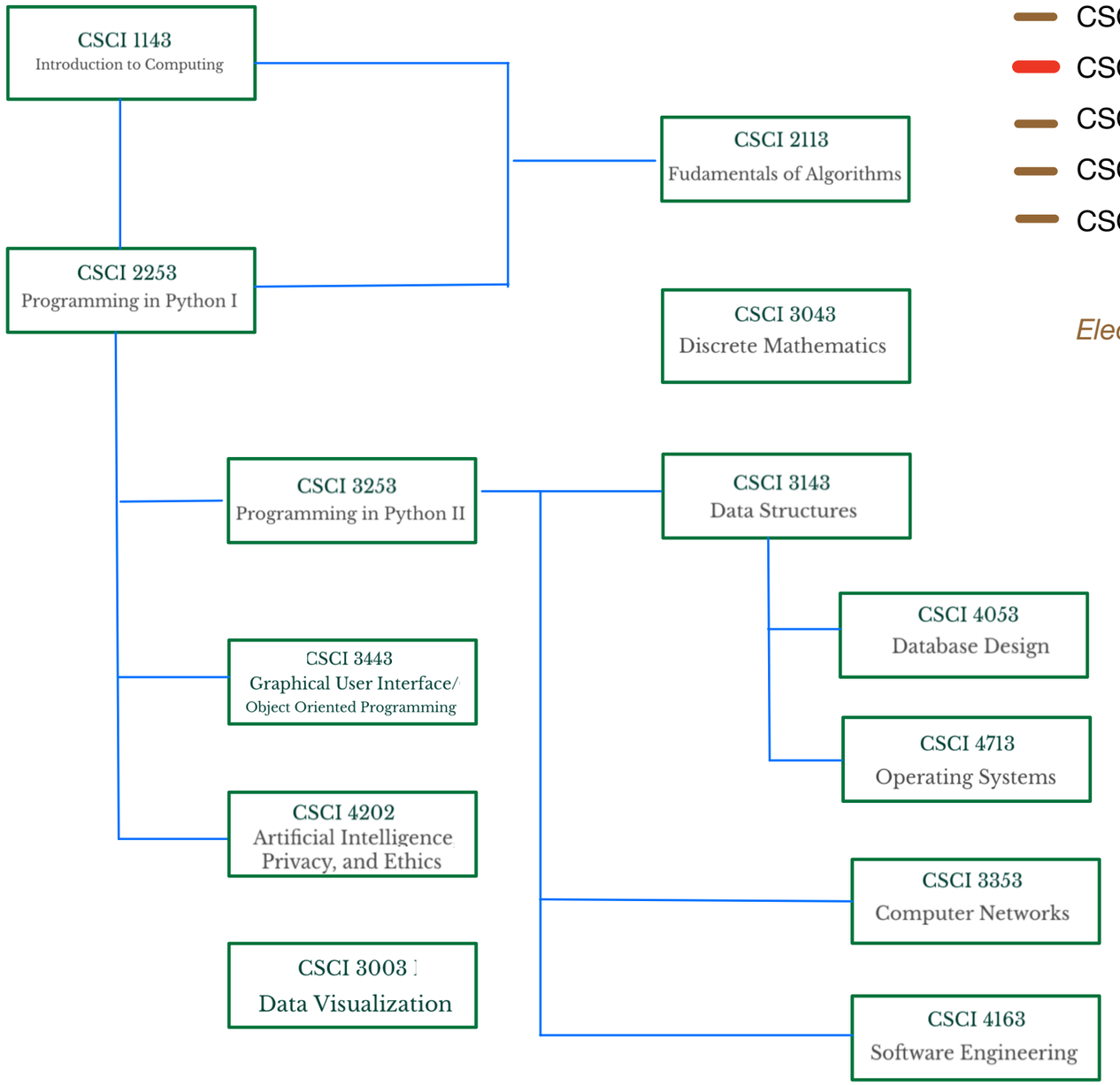

Falls

- CSCI-1143 Intro to Comp
- CSCI-2253 Prog Python I
- CSCI-3003 Data Visualization
- CSCI-3143 Data Structure
- CSCI-3353 Computer Networks
- CSCI-3443 Graphical User Interface/OOP
- CSCI-4163 Software Engineering

Springs

- CSCI-1143 Intro to Comp
- CSCI-2113 Fundamental Algorithms
- CSCI-3043 Discrete Mathematics
- CSCI-3253 Prog Python II
- CSCI-4053 Database Design
- CSCI-4202 AI Privacy & Ethics
- CSCI-4713 Operating Systems

Elective / Special topics / Internship / Independent Study / Workshop / Tutorial

Suggested Elective:

CSCI 3443 GUI/OOP
Parallel programming
Machine learning fundamentals
Applied machine learning
Text analytics
Intro to natural language processing
Computer Vision
Computer architecture

Numerical analysis
Applied probability
Linear optimization
Advanced data analysis
Graph theory

Faculty

- **Doughan**
- **Gheibi**
- **Malmstorm**
- **Regier**
- **Tran**

DSA & Alg1
DSA & Alg2

DS

AL

Data Structures:
Linear Data Structures:
- Arrays (fixed-size, dynamic)
- Linked Lists (singly, doubly, circular)
- Stacks (LIFO)
- Queues (FIFO, priority queues, deques)

Non-Linear Data Structures:
- Trees (binary trees, binary search trees, AVL trees, Red-Black trees, heaps, tries)
- Graphs (directed, undirected, weighted, unweighted)
- Hash Tables (hashing, collision resolution)

Sorting Algorithms:
- Comparison-based Sorts:
  - Bubble Sort
  - Selection Sort
  - Insertion Sort
  - Merge Sort
  - Quick Sort
  - Heap Sort
- Non-comparison-based Sorts:
  - Counting Sort
  - Radix Sort
  - Bucket Sort

Searching Algorithms:
- Linear Search
- Binary Search
- Hashing-based Search: (using hash tables)

Tree Traversal Algorithms:
- Breadth-First Search (BFS)
- Depth-First Search (DFS)

Other Important Topics:
Algorithm Analysis (Big O Notation): Time and space complexity analysis
Recursion
Dynamic Programming
Greedy Algorithms
Graph Algorithms: (e.g., Dijkstra's, Prim's, Kruskal's, Bellman-Ford, Floyd-Warshall)
Bit Manipulation: (sometimes included)

Course Title: Data Structures
Course Description:
This course provides a comprehensive introduction to fundamental data structures essential for efficient algorithm design and problem-solving. Students will explore both linear and non-linear data structures, understanding their implementation, operations, and applications. Topics include arrays, linked lists, stacks, queues, trees, graphs, and hash tables. Emphasis is placed on understanding the characteristics, advantages, and limitations of each structure, along with practical coding exercises to solidify comprehension. By the end of the course, students will be equipped with the knowledge to select and implement appropriate data structures for various computational problems.
Topics Covered:
Linear Data Structures: Arrays (fixed-size, dynamic), Linked Lists (singly, doubly, circular), Stacks (LIFO), Queues (FIFO, priority queues, deques)
Non-Linear Data Structures: Trees (binary, binary search, AVL, Red-Black, heaps, tries), Graphs (directed, undirected, weighted, unweighted), Hash Tables (hashing, collision resolution)
Basic algorithms related to data structures, such as traversal techniques and operations

Course Title: Algorithm Analysis and Design
Course Description:
This course delves into the principles of designing, analyzing, and implementing efficient algorithms to solve computational problems. Students will learn to evaluate algorithm performance using Big O notation, and explore core algorithmic paradigms including divide-and-conquer, greedy algorithms, dynamic programming, and recursion. The course covers a wide range of algorithms, including sorting, searching, graph algorithms, and tree traversals, with an emphasis on understanding their theoretical underpinnings and practical applications. Additionally, students will study advanced topics such as bit manipulation and fundamental algorithm analysis techniques. By the end of the course, students will be capable of analyzing and developing optimized algorithms for complex problems.
Topics Covered:
Algorithm Analysis: Time and space complexity, Big O notation
Searching Algorithms: Linear Search, Binary Search, Hashing-based Search
Sorting Algorithms: Comparison-based sorts (Bubble, Selection, Insertion, Merge, Quick, Heap), Non-comparison sorts (Counting, Radix, Bucket)
Graph Algorithms: Dijkstra's, Prim's, Kruskal's, Bellman-Ford, Floyd-Warshall
Tree Traversals: BFS, DFS
Core Algorithmic Paradigms: Divide-and-Conquer, Greedy algorithms, Dynamic Programming
Recursion and Backtracking
Bit Manipulation (optional, depending on course depth)

Intro Comp > Python I > Python II > AI & Ethics > Machine Learning (selective)
> Data visualizatian > data analysis (selective)
> Discrete Math
> Computer Networks
> Software Engineering

Intro Comp > Data Structure (Java) > Algorithm Analysis (java) > OOP&GUI (abstraction,Java) (selective)

> Operating systems
> Database Design