Davenport University

Course Syllabus

1. CSCI312 Data Structures/Algorithms

2. Credit Hours: 3 / Contact Hours: 45

3. Course Coordinator: Kowalczk, Brian

4. Textbook and Supplemental Materials

   9781111530532
   Java Programming: From Problem Analysis to Program Design
   D. Malik
   5th, 12 / Cengage

   OR

   9780134601540
   Visual C# 2016: How to Program
   Deitel, Paul & Deitel, Harvey
   6TH 17 / Pearson

5. Specific course information
   a. This course is a continuation of object-oriented programming that investigates advanced topics in technically oriented programming. Algorithmic analysis using computational complexity and big-O notation will be applied to classic data structures, including but not limited to arrays, vectors, linked lists, stacks, queues, trees, binary trees, binary search trees, and graphs. The computational complexity of classic searching and sorting algorithms will also be investigated.Prerequisite(s): CSCI232, CSCI234, or CSCI239
   b. Prerequisite(s): CSCI232, 234 or 239
   c. This is a required course.

6. Specific goals for the course
   a. Utilize sorting algorithms to order objects contained within linear data structures.
      i. Utilize algorithms to search for objects within linear and non-linear data structures.

ii. Implement a solution that utilizes the appropriate data structure(s) that meets the goals of a specific program.
iii. Analyze the strengths and weaknesses of linear data structures including arrays, vectors, linked lists, Stacks, and Queues.
iv. Compare and contrast the use of Trees, Binary Search Trees, and Graphs.
v. Discuss the asymptotic complexity of a retrieval and insertion operation on a given data structure utilizing Big-O notation.
vi. Utilize tools and methodologies that integrate security best practices throughout the software development lifecycle to mitigate risks, reduce attack surfaces, and increase the quality of software development efforts

b. This course measures the student outcome(s)

#2: design, implement and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

c. Mapping of Course Learning Outcomes to ABET's Student Outcomes

i. Utilize algorithms to search for objects within linear and non-linear data structures. (ABET SO #1)
ii. Implement a solution that utilizes the appropriate data structure(s) that meets the goals of a specific program. (ABET SO #2)
iii. Analyze the strengths and weaknesses of linear data structures including arrays, vectors, linked lists, Stacks, and Queues. (ABET SO #1)
iv. Compare and contrast the use of Trees, Binary Search Trees, and Graphs.
v. Discuss the asymptotic complexity of a retrieval and insertion operation on a given data structure utilizing Big-O notation. (ABET SO #1)
vi. Utilize tools and methodologies that integrate security best practices throughout the software development lifecycle to mitigate risks, reduce attack surfaces, and increase the quality of software development efforts. (ABET SO #6)

7. A brief list of topics to be covered
   a. Asymptotic Complexity (2)
   b. Arrays (2)
   c. Vectors (2)
   d. Linked Lists (2)
   e. Stack (2)
   f. Queue (2)
   g. Tree (2)
   h. Binary Tree (2)
   i. Binary Search Tree (2)
   j. Graphs (2)
   k. Linear Search (2)
   l. Binary Search (2)

m. Sorting (4)

   i. bubble sort
   ii. shell sort
   iii. quicksort
   iv. merge sort