Syllabus for

CSC 255—Data Structures

4 Credit Hours

Fall 2024

I. COURSE DESCRIPTION

A study of the design of structures for representing information and the design of algorithms for manipulating that information. Expertise in the design of structures is developed through consideration of abstract structures and implementation techniques and implementing various structures in specific programming languages. Develops expertise in the design of algorithms by solving problems, including searching and sorting. Programming projects throughout the course provide a synthesis experience in which the student designs data structures and algorithms to solve a series of increasingly complex problems. The class is three credit hours of lecture and one credit hour of lab.
Prerequisite: CSC 206 with a grade of "C" or higher.

II. STUDENT LEARNING OUTCOMES

A. COURSE OUTCOMES

As a result of successfully completing this course, the student:

1. Can explain the purpose and value of developing efficient structures for data storage, retrieval, and manipulation.
2. Can describe the nature of fundamental data structures and associated algorithms, including lists, stacks, queues, binary search trees (BSTs), balanced BSTs, hash tables, and graphs.
3. Can explain and implement sorting algorithms, including insertion sort, quicksort, and heapsort.

Unit Objectives

1. Unit I Objectives
   As a result of successfully completing this unit, the student will be able to do the following:
   a. Write the definition, draw the logical diagram, design a storage structure, and write manipulation algorithms for the following data structures: list, linked list, stack, queue, circular list, binary search tree, and AVL tree.
   b. Utilize recursion in solving a problem.

2. Unit II Objectives
   As a result of successfully completing this unit, the student will be able to do the following:
   a. Write the definition, draw the logical diagram, design a storage structure, and write manipulation algorithms for 2-3 trees, hash tables, quadratic sorting, heapsort, Shellsort, mergesort, quicksort, dynamic order statistics.

3. Unit III Objectives
   As a result of successfully completing this unit, the student will be able to do the following:
   a. Write the definition, draw the logical diagram, design a storage structure, and write manipulation algorithms for B-Trees, Disjoint Sets, and Huffman trees.
   b. Draw graph structure diagrams and write graph traversal algorithms.

© Copyright Oral Roberts University 2010. All rights reserved. (Fall 2019-SW)

B. PROFESSIONAL OUTCOMES
This course aligns with and evaluates the following professional outcomes as indicated on the last page:
Computer Science
1. ABET Outcome 1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. ABET Outcome 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.
3. ABET Outcome 6: Apply computer science theory and software development fundamentals to produce computing-based solutions.

C. UNIVERSITY OUTCOMES
This course aligns with and evaluates the following University outcomes as indicated on the last page:
1. Personal Resilience
2. Intellectual Pursuit

III. TEXTBOOKS AND OTHER LEARNING RESOURCES

A. Required Materials
1. Textbook
Introduction to Algorithms, 4th Edition (MIT Press) Publisher: The MIT Press; 2022; ISBN-13: 978-0262046305. Also available in Paperback.
2. Other
None

B. Optional Materials
1. Textbooks
None
2. Other
None

IV. POLICIES AND PROCEDURES

A. Department Policies and Procedures
1. Attendance and Excessive Absences - Attendance at each class or laboratory is mandatory. Excessive absences can reduce a student's grade or deny credit for the course.
2. Unexcused Absences - Any student whose unexcused absences total 33% or more of the total number of class sessions will receive an F for the course grade.
3. Computer Resources - Each Student who uses the computer is given access to the appropriate computer resources. These limited resources and privileges are given to allow students to perform course assignments. Abuse of these privileges will result in their curtailment. Students should note that the contents of computer directories are subject to review by instructors and the computer administrative staff.
4. Late Exams - Each instructor has his or her own late-exam policy, so an instructor may decide that an exam missed because of an unexcused absence cannot be made up.
5. Incompletes – As stated in the University catalog, incompletes are granted only for "good cause," such as extended hospitalization, long-term illness, or a death in the

© Copyright Oral Roberts University 2018. All rights reserved.

family. Students must petition for an incomplete using the "Petition for Incomplete" form at petitions.oru.edu. Very few incompletes are granted.

B. Course Policies and Procedures

1. Evaluation Procedures
   a. The following distribution determines the composite score:

Early Activity	5%
Programs	35%
Exams	35%
Final Exam	25%

   b. Grading scale:

      A=90%
      B=80%
      C=70%
      D=60%
      F=59% and below

   The student who wants to know his or her grade in the course should keep a record of all points earned at all times.

2. Originality of submitted work.
   All work for graded and ungraded submissions shall be original work by the student. The use of third-party contributions to solving an assignment's key objective(s), in part or in whole, whether cited or otherwise is prohibited.
   Whether or not used in a submission, using any resource that addresses the core objective of any submission is prohibited.
   A student is expected to be able to describe any part of a submission in detail at any time.
   Evidence of this policy being violated can include third-party sources and/or the lack of the student's ability to demonstrate complete understanding of their submission.
   Violations of this policy may result in the assignment receiving a diminished grade, including a failing grade. It may also result in the student receiving a diminished grade, including an F, for the course.
3. Late work
   The instructor may choose to accept late work and to what extent the grade would be assessed for lateness. Acceptance of late work does not imply that other late work would be accepted.
4. Other Policies and/or Procedures
   Homework assignments and programming problems are given regularly in class. Details of specific requirements will be given at that time.
5. Other Policies and/or Procedures
   Homework assignments and programming problems are given regularly in class. Details of specific requirements will be given at that time.

© Copyright Oral Roberts University 2018. All rights reserved.

V. COURSE CALENDAR

Week 1	C++ Basics, Arrays, and Pointers
Week 2	Circular List, Linked Lists, and complexity analysis
Week 3	Doubly linked lists, stacks, and queues
Week 4	Binary Search Trees and AVL trees
Week 5	AVL Trees and Exam 1
Week 6	2-3 Trees and Heaps
Week 7	Priority Queues, Hashing, and Quadratic Sorting.
Week 8	Heapsort, Shellsort, and Mergesort.
Week 9	Quicksort and Order Statistics
Week 10	Spring Break
Week 11	Dynamic Order Statistics and Exam 2
Week 12	B-Trees and Disjoint Sets
Week 13	Huffman Encoding
Week 14	Graph Representation, Directed/Undirected, DFS, and BFS
Week 15	Topological Sort and Dijkstra

© Copyright Oral Roberts University 2018. All rights reserved.

Primary Program: Computer Science
CSC 255—Data Structures
Fall 2024

This course contributes to the University and program outcomes as indicated below:
Significant Contribution – Addresses the outcome directly and includes targeted assessment.
Moderate Contribution – Addresses the outcome directly or indirectly and includes some assessment.
Minimal Contribution – Addresses the outcome indirectly and includes little or no assessment.

OUTCOMES	**Significant**	**Moderate**	**Minimal**
Spiritual Integrity			
ABET Outcome 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.		**X**	
Personal Resilience			
CSC 255: Can explain the purpose and value of developing efficient structures for data storage, retrieval, and manipulation.			**X**
CSC 255: Can describe the nature of fundamental data structures and associated algorithms, including lists, stacks, queues, binary search trees (BSTs), balanced BSTs, hash tables, and graphs.			**X**
CSC 255: Can explain and implement sorting algorithms, including insertion sort, quicksort, and heapsort.			**X**

Intellectual Pursuit			
ABET Outcome 1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.	**X**		
ABET Outcome 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.	**X**		
ABET Outcome 6: Apply computer science theory and software development fundamentals to produce computing-based solutions.	**X**		
CSC 255: Can explain the purpose and value of developing efficient structures for data storage, retrieval, and manipulation.		**X**	
CSC 255: Can describe the nature of fundamental data structures and associated algorithms, including lists, stacks, queues, binary search trees (BSTs), balanced BSTs, hash tables, and graphs.		**X**	
CSC 255: Can explain and implement sorting algorithms, including insertion sort, quicksort, and heapsort.	**X**		
Global Engagement			
ABET Outcome 1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.		**X**	
ABET Outcome 6: Apply computer science theory and software development fundamentals to produce computing-based solutions.		**X**	
Bold Vision			
ABET Outcome 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.	**X**		
ABET Outcome 6: Apply computer science theory and software development fundamentals to produce computing-based solutions.	**X**		

© Copyright Oral Roberts University 2018. All rights reserved.