University of Texas at El Paso
Course Syllabus

COURSE DESCRIPTION

Dept., Number	CS 2302	**Course Title**	Data Structures
Approval Date	April 2022	**Course Coordinator**	Daniel Mejia

CATALOG DESCRIPTION

Data Structures: [TCCN COSC 2336] Abstract data types, representation of data using sets, lists trees and graphs. Storage allocation and collection techniques.

TEXTBOOK

Introduction to Python Programming and Data Structures, 3rd Edition
Y. Daniel Liang

COURSE OUTCOMES

Level 1: Knowledge and comprehension:
Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. On successful completion of this course, students will be able to:

1. Identify and explain the following algorithm design techniques:
   a. Greedy algorithms
   b. Divide and conquer
   c. Dynamic programming
   d. Backtracking
   e. Randomized algorithms

Level 2: Application and analysis:
Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details.
Upon successful completion of this course, students will be able to:

1. Describe, implement, and use the following data structures:
   a. Heaps
   b. Balanced trees
   c. Graphs
2. Solve problems using hashing, specifically using language-specific data structures (e.g., sets and dictionaries in Python)
3. Describe, implement, and apply the following graph algorithms:
   a. Breadth-first search
   b. Depth-first search
   c. Topological sorting
   d. Minimum spanning trees (Kruskal's and Prim's)
   e. Single-source shortest paths (Dijkstra's algorithm)

4. Assess space requirements of algorithms in relation to the size of their inputs.

Level 3: Synthesis and evaluation:
Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. On successful completion of this course students will be able to:

1. Given a problem, judge which data structures are required to solve it efficiently and justify the selection
2. Solve problems using arrays and lists
3. Given a non-recursive algorithm examine its loop structure, assess its asymptotic running time in relation to the size of the input, and express it using big-O notation
4. Given a recursive algorithm, examine its structure, formulate, and solve a recurrence equation defining its running time in relation to the size of the input, and express it using big-O notation
5. Design and implement solutions to computational problems based on iteration and recursion
6. Trace the behavior of functions and algorithms involving iteration and recursion

ABET STUDENT OUTCOMES MAPPING

Course Outcomes	Student Outcome
2.4, 3.3, 3.4	1
3.1, 3.2, 3.5	2 (ABET 1)
2.1, 2.2, 2.3	3 (ABET 2)
	4 (ABET 5)
	5 (ABET 4)
	6 (ABET 3)
	7
	8
1.1, 2.1, 2.2, 2.3	9
3.6	10 (ABET 6)

PREREQUISITES BY TOPIC

CS 2401 with C or better
MATH 2300 or (CS 2101 & CS 2202) with C or better