CSED 5320
Data Structures and Algorithms

Course Learning Outcomes

Course Catalog Description

Introduction to the structure and application of common data structures used in computer science and the algorithms used with/for these structures. Includes an ongoing discussion on algorithm analysis.

The Mission of the UNI Educator Preparation Program

The UNI Educator Preparation Program provides an authentic and challenging education that empowers candidates to serve as reflective, professional educators who advocate for students, schools, communities, and the profession in a dynamic and changing world.

Belief Statements UNI Educator Preparation Program:

1. Candidates must deeply understand and reflect on their content and pedagogy.
2. Candidates must engage in rich, purposeful, and authentic field-based experiences to develop appropriate dispositions and practices.
3. Candidates have a responsibility to understand historical, social, cultural, and political contexts and how they impact education.
4. Candidates must understand the importance of diversity and equity and engage in opportunities to promote social justice.
5. Candidates must develop competence in the skills and dispositions that allow them to engage in effective leadership and advocacy.
6. Candidates must develop strong skills in order to effectively collaborate with all stakeholders for student learning.

Course Competencies

The material in this course has been designed to align with

The CSTA K-12 Standards (https://csteachers.org/k12standards/)

The CSTA Standards for CS Teachers (https://csteachers.org/teacherstandards/).

The material in this course has been divided into seven major competencies that well-educated CS teachers should be able to demonstrate. Each of these competencies is subdivided into one or more measurable outcomes. Students should be able to:

- **Competency 1** — Object Oriented Code
  - Identify and explain the key concepts of object-oriented programming, including classes, objects, methods, inheritance, polymorphism, encapsulation, and abstraction. [3A-AP-15]
  - Recognize and describe the purpose and structure of Python classes and objects in a provided code snippet. [3B-AP-16]

- **Competency 2** — Algorithm Analysis
  - Employ appropriate vocabulary to discuss algorithmic efficiency, including terms like Big-O notation, time complexity, and space complexity.
  - Analyze code to determine its execution-time (big-oh notation) and storage utilization. [3A-AP-15, 3B-AP-11, Teacher 1e]
- **Competency 3** — Linear Data Structures
  - Trace, identify and explain common "linear" data structures constructed using "arrays" (i.e., contiguous block of memory) and "linked nodes" as appropriate: stack, queue, and list. [3B-AP-12, Teacher 1e]
- **Competency 4** — Searching and Sorting Algorithms
  - Trace, explain, and analyze common search/sort techniques such as linear search, binary search, closed-address hashing. [3B-AP-10, Teacher 1e]
  - Explain and analyze simple and advanced sorts such as bubble, selection, insertion, merge, and quick sorts. [3B-AP-10, Teacher 1e]
- **Competency 5** — Recursive Algorithms
  - Define recursion and identify the components of a recursive function.
  - Trace the execution of a recursive function, demonstrating understanding by outlining the calls and return values step-by-step. [3B-AP-13]
- **Competency 6** — Searching Algorithms and Tree Data Structures
  - Trace, identify, and explain the appropriate use of common "tree" data structures. [3B-AP-09]
- **Competency 7** — Additional “Traditional” Algorithms/Structures
  - Research a new data structure or algorithm not previously covered in the course, using credible sources, to understand its purpose and application. Summarize the algorithm’s key components for peers/students [2-IC-22].
  - Trace, identify, and explain common graph algorithms: depth-first search, breadth-first search, Prim’s algorithm, Dijkstra’s algorithm, and topological sort.
- **Across all competencies**
  - Apply appropriate terminology when describing the characteristics, advantages, and limitations of different data structures such as array, stack, queue, tree, graph, dictionary, and hash table.

Course Grading

Grading Structure

I use a grading system that is a combination of standards-based grading and Grading for Equity. My main beliefs are:

- As the teacher, my job is to design a variety of chances for you to learn material, practice required skills, and demonstrate overall competency.
- As the student, your job is to show that you can meet the course outcomes by demonstrating the defined competencies.
- Your final grade in the class should indicate how well you were able to meet course objectives/competencies by the end of the semester.

I WANT you to succeed in the course. That means giving you multiple opportunities to show that you can demonstrate course competencies. In most cases, if you can't do this the first time, you will be able to re-study and try again.

You will complete a variety of activities in this course to demonstrate your abilities. For each, your deliverable will be converted to a competency evaluation score from 1-4 which is a summarization of the overall competency you have demonstrated based on the following rubric.

Score	Meaning
1	UNASSESSABLE - You submitted deliverables for the activity but what you submitted shows little understanding of the competency being assessed.
2	NEEDS WORK - You have made significant progress towards demonstrating competency but there are limited items that remain unsatisfied.
3	SATISFACTORY - You have **met** the standards of the competency. [Your results show reasonable competency with few mistakes or remaining issues.]
4	EXCELLENT - You have **exceeded** the standards of the competency. [You have met the standards of competency and shown considerable understanding/knowledge of the material.]

While it might be tempting to view these categories as similar to GPA categories (which also use a 4-point scale) that is not the way they are used or interpreted.

If you are unsatisfied with your score on any competency or activity (in particular, if you did not earn at least a 3), you may meet with me to discuss the situation, restudy the material, and make a second attempt to demonstrate your ability to meet the standards of the competency/activity.

Final Grades

Final course grades will be determined using the following evaluation criteria.

Grade Earned	Average Score	Additional Conditions
A	> 3.5	All scores 3-4
A-	> 3.5	All scores 2-4
B	> 3	All scores 3-4
B-	> 3	All scores 1-4
C	> 2.5	All scores 2-4
C-	> 2.5	All scores 1-4
D	> 2	All scores 2-4
D-	> 2	
F	≤ 2	

Because I allow – and encourage – retakes, most students do well in this course. It is rare for a student who has been an active participant in the class to not earn at least a C for a final grade.

Additional Comments:

- Communication is key: I cannot help you if I do not know what is going on. If you are having trouble with a topic in the class, please reach out to me early. Do not wait until the situation is out of control. I am very willing to help. However, I must know that you need and want that help.
- If I feel there are specific and individual circumstances where "mathematically" you earned a grade slightly lower than I feel your overall competence has demonstrated, then I reserve the right to raise your grade one level from that published above.
- To be responsive to your needs I reserve the right to modify the structure of this course as we are in progress. If there is significant deviation from the policies described in this syllabus, the new policy will be clearly discussed with you and in a timeframe that gives you time to plan accordingly.