Sul Ross State University

A Member of the Texas State University System

CS2315 – Data Structures – Fall 2019

Instructor: Ms. G

Office Location: MAB 109B **Office Phone**: 931-237-3324

Email: tglenn2@sulross.edu

Office Hours:

M & W 8 – 9am & 1-2:30 pm

You may call from 10 am to 10 pm Mon-Sun if you need assistance with your homework or have issues that need to be discussed. Call 931-237-3324 Be brave be bold call if you need help.

Lab:

M 3:00 – 4:50

Program Learning Objective

1. Understand the fundamental concepts of computer science including algorithms and data structures.
2. Understand modern computer systems, databases and networking.
3. Display an understanding and ability to implement current programming methodologies.
4. Become proficient with systems design based on object-oriented programming.
5. Work as a team in workgroup environments.

Course Objectives:

This course covers the modern theory of algorithms, focusing on the themes of efficient algorithms and intractable problems. The course goal is to provide a solid background in algorithms for computer science students, in preparation either for a job in industry or for more advanced courses at the graduate level.

Schedule Breakdown

Week 1

Subject: Course Overview and Programming Overview Part I

Goal: Allow students to understate the full scope of this course and what the expectations are for successful completion of the class.

Learning Objective:

1. Students will know how to implement identifiers, work with basic input/output functions, for and while loops, and know how to construct the basic structure of a typical object oriented programming language.
2. Students will be able to perform basic Boolean math.

Notes:

Reading:

Week 2

Subject: Object Oriented Programming

Goal: This module is designed to ensure students understand object oriented programming will enough to be able to effectively design the systems using UML which is based on object oriented development.

Learning Objective: Students will understand and be able to effectively implement:

1. Classes
2. Member Variables
3. Methods
4. Constructers
5. Protection levels
6. Static vs. Dynamic Classes and Types
7. Abstraction
8. Overriding methods
9. Interfaces

Notes: The goal for this module should be met by taking the object oriented programming class but this should serve as a nice refresher if the student is rusty in this area.

Reading:

Week 3

Subject: Complexity Analysis

Goal: This module will introduce students to amortization and estimating run-times of algorithms using Big O notation.

Learning Objective:

1. Be able to explain the purpose of code complexity and asymptotic growth analysis
2. Able to successfully identify an algorithm's Big O

Notes:

Reading:

Week 4

Subject: Arrays and Stacks

Goal: This module will revisit the Array structure and how to use an array to create a stack ADT.

Learning Objective:

1. Know how to implement a stack using an array
2. Demonstrate the use of a stack in an algorithm

Notes:

Week 5

Subject: Queues

Goal: This module will introduce students to first in first out Queues and how to implement a Queue using an array.

Learning Objective:

1. Know how to implement a queue using an array
2. Demonstrate the use of a queue in an algorithm

Notes:

Reading:

Week 6

Subject: Linked Lists

Goal: This module will introduce students to the List ADT and will demonstrate how to use a singly linked list to create stacks and queues.

Learning Objective:

1. Know how to implement a linked list
2. Demonstrate the use of a list in an algorithm
3. Able to implement an entry

Notes:

Reading:

Week 7

Subject: Sorting Algorithms

Goal: This module will introduce students to various sorting algorithms including selection and insertion sort, merge sort and quick sort. We will also spend time analyzing the runtime complexity of each of the sorting algorithms.

Learning Objective:

1. Know the main sorting algorithms and their respective Big O complexities
2. Understand which sorting algorithm is most appropriate for a given application

Notes:

Week 8

Subject: Priority Queues

Goal: This module will introduce students to the priority queues and segway out of the linear ADTs into graphs and trees.

Learning Objective:

1. Be able to identify the primary methods in a priority queue
2. Be able to implement a priority Queue
3. Be able to effectively use a priority Queue in an application
4. Understand when it is ok to use selection sort, insertion sort or a min heap
5. Be able to correctly provide the Big O for the various methods of sorting and for enqueue, dequeue and overall Big O for the queue using the various sorting algorithms.

Notes:

Week 9

MIDTERM

Week 10

Subject: Trees -- Binary Search Trees -- Heaps

Goal: This module will introduce students to the Heap ADT and how to implement a Heap.

Learning Objective:

1. Know when it is appropriate to use a heap
2. Know the difference between the hash function and the coding function
3. Know how to encode basic strings with minimal hits

Week 11

Subject: Dictionaries -- Hashing

Goal: This module will introduce students to the Heap ADT and how to implement a Heap.

Learning Objective:

1. Know when it is appropriate to use a heap
2. Know the difference between the hash function and the coding function
3. Know how to encode basic strings with minimal hits

Week 12

Subject: Graphs

Goal: This module will introduce students to basic graph theory.

Learning Objective:

1. Know what edges, vertices and nodes are
2. Know how graphs are related to ADTs
3. Know what a tree is
4. Know what the difference is between a balance and unbalance tree is
5. Know how to implement binary search and what its Big O is

Notes:

Reading:

Week 13

Subject: Search

Goal: This module will introduce to the DFS algorithm and what applications the algorithm is used for.

Learning Objective:

1. Know how to find the minimum spanning tree using DFS
2. Know the basic uses of a DFS
3. Know how to find the minimum spanning tree using BFS
4. Know the basic uses of a BFS
5. Know what advantage BFS brings that DFS does not

Notes:

Reading:

Week 14

Subject: Tree Balancing

Goal: This module will introduce to overcoming the speed loss resulting from unbalance trees

Learning Objective:

1. Understand the AVL algorithm
2. Be able to compare and contrast a binary tree with an AVL tree in terms of Big O analysis

Notes:

Reading:

Textbook:

None

Teaching Methods

Assignments: Exercises will be periodically assigned to help support and supplement material found in the textbook. No make ups are allowed, unless medical or extreme conditions are considered. Tests, assignments and final exam will be delivered through Blackboard with a date and time limit. No make ups are allowed, unless medical or extreme conditions are considered.

Grading

Letter grades will be determined using a standard percentage point evaluation as outlined below.

A 90 - 100 points

B 80 - 89 points

C 70 – 79 points

D 60 – 69 points

F Below 60 points The final grade will be computed on the following weights:

Grade breakdown:

Homeworks ==> 30%

Participation ==> 20%

Midterm ==> 25%

Final ==> 25%

Attendance

Any student who accumulates 10 **unexcused** absences (MWF Classes) or 7 **unexcused** absences (MW classes) will be automatically dropped from this course.

Course Policies

Exams, quizzes and assignments: NO MAKE UPS ARE ALLOWED, unless medical or extreme conditions are present.

Academic dishonesty

You are expected to do your own work on all assignments, exams, quizzes, and projects. Any dishonest work will be penalized with a grade of zero.

Need for assistance

Qualified students with disabilities needing academic or other accommodations to ensure full participation in the programs, services and activities at Sul Ross State University should contact the Disabilities Services Coordinator, in Counseling and Prevention Services, Ferguson Hall 112, Box C-117, Alpine, Texas 79832.

Posting of Grades

As soon as assignments, tests and final exam are graded, the grades will be posted in Blackboard.

Additional resources

Stacks:

https://www.youtube.com/watch?v=FNZ5o9S9prU

https://www.youtube.com/watch?v=2JpfHyGhIfs

https://www.youtube.com/watch?v=Z7f3IFqID0s

Queues:

https://www.youtube.com/watch?v=PjQdvpWfCmE

https://www.youtube.com/watch?v=GRzV2D5gslM

Heaps:

https://www.youtube.com/watch?v=v1YUApMYXO4

Priority Queues

https://www.youtube.com/watch?v=QJ_7S1p0Kj8

https://www.youtube.com/watch?v=P4toxusBX9M

https://www.youtube.com/watch?v=w647KdGJmpI

Hash Tables

https://www.youtube.com/watch?v=MfhjkfocRR0

https://www.youtube.com/watch?v=Ke_tII6Y0GE