COSC 2430
Data Structures and Algorithms

Course Information

Term and Year: **Fall 2020**
Location: **Online**
Meeting Days/Times: Tuesdays -Thursday 10:00 AM -11:15 AM && - 11:30 AM- 12:45 PM

Instructor Information

Name: **Nouhad Rizk**
Email Address: njrizk@uh.edu
Office Hours: *Tuesday – Thursday 12:45 PM – 1:15 PM*

Course Description

Covers the design, analysis, and implementation of data structures and algorithms to solve problems using an object-oriented programming language. Topics include elementary data structures, (including arrays, stacks, queues, and lists), advanced data structures (including trees and graphs), the algorithms used to manipulate these structures, and their application to solving practical engineering problems.

Course Materials

- **Text Book**

  Required Weiss, Mark A. *Data Structures and Algorithm Analysis in C++*. 4th Edition. Pearson 2014. ISBN-13: 978-0-13-284737-7

  Open Book:
  https://open.umn.edu/opentextbooks/textbooks/open-data-structures-an-introduction

  Webcam/camera is REQUIRED for exams

- Reference
  - Malik, D S. *Data Structures in C++,* 2nd Edition, CengageLearning

- **Compiler**
  - Microsoft Windows Visual Studio 15 or higher
  - GCC 5.3 for Linux
  - Clang or g++ compilers for OS X (check online for latest versions)

Program Objectives

The objective is to prepare students such that within a few years of graduation they will be able to secure for themselves a successful computer science career and contribute decisively to the improvement and development of technology by demonstrating their ability to:

1. Address and solve complex broadly defined problems related to their discipline and field of specialization
2. Work as team members, show leadership, and communicate technical concepts and ideas effectively
3. Manifest a high level of professional integrity, and make ethical decisions that will have a positive impact on the organization and society
4. Embrace and practice lifelong learning, continue personal growth, and professional self-improvement.

Student Learning Outcomes

Based on satisfactory completion of the course, a student should be able to:

- Demonstrate an understanding of basic data structures (such as an array-based list, linked list, stack, queue, binary search tree) and algorithms.
- Demonstrate the ability to analyze, design, apply and use data structures and algorithms to solve problems and evaluate their solutions.
- Demonstrate an understanding of analysis of algorithms. Study an algorithm or program code segment that contains iterative constructs and analyze the asymptotic time complexity of the algorithm or code segment.

Important Notes:

1. Our time together is very valuable; please treat it accordingly. By enrolling in this course, you make a personal contract with me and your classmates to attend and diligently participate in every class activity. Students are expected to be courteous toward the instructor and their classmates throughout the duration of this course.

2. Being mostly inattentive (in the judgment of the instructor or the TA's), arriving late, and leaving the classroom in and out will disqualify you from participation credit for the course (5%). It is important to be respectful of your fellow students and the shared course environment. It is a professional learning situation, not your living room.

3. Homework/labs/GA must be submitted on the due date. **No** late or email submissions will be accepted. If ever accepted (Logical reason) a **1% penalty will be applied per late hour**.

4. **MAKE-UPEXAMS**: **There are no make- up exams.**

5. **3-Day Policy**: One has **3 days** starting from the end of the class time in which the graded assignment/exam papers have been distributed and/or posted in order to object to the score of that assignment or exam. The objection shall be submitted electronically by emailing the TA and the instructor.

6. **Academic Honor Code:** As a student, you join a community of scholars who are committed to excellence in learning. I assume that students will pursue their studies with integrity and honesty. **zero-tolerance for cheating, whether in exams, quizzes or programming assignments.** Plagiarism, copying and other anti- intellectual behavior are prohibited by the university regulations. Violators will face serious consequences.
7. **Student Conduct:** Disruptive behavior inside or outside class may result in disciplinary actions and academic failure. Students must refrain from disturbing the peace and good order of the university. For more details, please refer to http://www.uh.edu/dos/pdf/codeofconduct.pdf
8. **Academic Integrity**: Cheating or any other suspected violations of academic integrity will not be tolerated and will be reported to the Department of Computer Science, Director of Undergraduate/Graduate Studies and if substantiated may result in significant penalty. It is each student's responsibility to read and understand the Academic Honesty Policy found in the Student Handbook (http://www.uh.edu/academics/catalog/policies/academ-reg/academic-honesty/).
9. Plagiarism: Plagiarism is using someone else's work without proper acknowledgement. This includes getting help from a friend or colleague and online material. When using someone else's work, always cite the source. Plagiarism is considered a serious breach of academic integrity. **Any breach of academic integrity or plagiarism would result in a minimum of one full letter grade reduction over the final score and possible expulsion from university.**

10. Mentors can do a replacement quiz; the mentor should send the grade to your TA.

Expectations

- ✓ Workload: This is a *15-week-three semester credit* course. Students should expect 2.5 weekly hours of classroom faculty instruction, and a minimum of 5 hours of out-of-class studentwork per week.
- ✓ Lab Work is 5% of the total grade

- ✓ Students are expected to meet with mentors Room PGH 224 for any additional help.

Course Methodology

This course emphasizes the choice and use of appropriate data structures and efficient algorithms in implementing applications. Instructor and students interact directly in class. During the lectures, the instructor discusses the behavior of the data structures and the efficiency of algorithms that operate on the data.

This course will combine traditional lecturing with hands-on exercises to reinforce student learning. Students are expected to attend classes regularly, take tests, and submit assignments and other work at the times specified by the instructor.

The instructor reserves the right to make changes to the course schedule as needed under unexpected circumstances. These changes will be announced in class and on Blackboard.

Week	Dates	Topic	Reading
1	8/25 – 8/27	Basic C++ Recursion	Chapter 1 includes material that serves as a review of basic C++. Included is a discussion of templates, recursions and important constructs in C++ class design.

2	9/1 – 9/3	Array-Based Implementations Link-Based Implementations	Chapter 3: 3.1,3.2,3.3 Implementations of a significant list class
3	9/8 – 9/10	Stacks Stack Implementations	Chapter 3: 3.6 Implementations of a significant stack class **Wednesday September 9 DROP DEADLINE**
4	9/15 –9/17	Queues Queue Implementations	Chapter 3: 3.7 Implementations of a significant Queue class
5	9/22 –9/24	Algorithm Efficiency Sorting Algorithms and Their Efficiency	Chapter 2 Explains asymptotic analysis and its major weaknesses. in-depth understanding of logarithmic running time. Analyze recursive programs by intuitively converting them into iterative programs. Chapter 7: 7.2,7.3 All the important general-purpose sorting algorithms are covered and compared. Four algorithms are analyzed in detail: insertion sort, selection,Shellsort, heapsort, and quicksort.
6	9/29-10/1	**Exam 1** Sorted Lists and Their Implementations	Chapter 7: 7.4, 7.5, 7.6 7.7
7	10/6- 10/8	Queues and Priority Queues Heaps	Chapter 6: priority queues. Binary heaps are covered, and the implementations of priority queues.
8	10/13 – 10 /15	Queue Implementations	

9	10/20-10/22	Trees	Chapter 4 Emphasis on search trees, including external search trees (B-trees)
10	10/27– 10/29	Heaps **Exam 2**	
11	11/3 – 11/5	Hashing	Chapter 5 hash tables, including the classic algorithms such as separate chaining and linear and quadratic probing, **Tuesday November 3rd: DROP DEADLINE**
12	11/10– 11/12	Tree Implementations Balanced Search Trees	Chapter 4
13	11/17– 11/19	AVL Tree	Chapter 4 Emphasis on AVL trees.
14	11/24–	Graphs Greedy (Huffman code Optional)	Chapter 9 covers graph algorithms. Chapter 10 Greedy algorithms
15	12/5	Last Day of the semester December 5th, 2020 Saturday …**NOTHING IS ACCEPTED AFTER THIS DATE**	(Regrade)

Grading Policy

The final numeric grade is computed based on student's performance in weekly assignments and exams/quizzes. The final numeric grade for the course will be determined as follows:

- ✓ Homework assignments (NO drop of any HW) 30%
- ✓ Lab work (drop the lowest) 15%
- ✓ Exam 1 (Tuesday 9/29) 15%
- ✓ Exam 2 (Thursday 10/29) 15%
- ✓ Final Exam 25%
- ✓ Participation: GA Assignment and/or In class Quizzes 5%

Grading Merit

A>=92.5 Excellent	**A->= 89.5 and < 92.5 Outstanding**	**B+>=86.5 and < 89.5 Very Good**
B > = 83.5 and <86.5 Good	**B->=79.5 and < 83.5 Above Average**	**C+>=76.5 and < 79.5 High Average**
C>=72.5 and <76.5 Average	**C->=69.5 and <72.5 Low Average**	**D+>=65.5 and <69.5 Below Average**
D >=62.5 and <65.5 Poor	**F < 62.5 Failing**	

Programming Assignments/ HW with topics(tentative)

HW: NO drop of any HW	**Posted Mondays**	**Due Mondays**
HW 1 (Array &recursion)	Beginning of week 1	Beginning of week 2
HW 2 (List)	Beginning of week 2	Beginning of week 4
HW 3 (Stack &list)	Beginning of week 5	Beginning of week 8
HW 4 (Queue)	Beginning of week 8	Beginning of week 11
HW 5(Trees)	Beginning of week 11	Beginning of week 13
(regrade)		

Labs with topics(tentative)

Labs	**Posted (Wednesdays)**	**Due (Wednesdays)**
Lab1 on files	**Mid of week 3**	**Mid of week 4**
Lab2 on linked list as Arrays and recursion	**Mid of week 5**	**Mid of week 6**
Lab3 sorting with clock	**Mid of week 7**	**Mid of week 8**
Lab 4 Round robin queue	**Mid of week 9**	**Mid of week 10**
Lab5 Hashing	**Mid of week 11**	**Mid of week 12**
Lab6 On BST or AVL	**Mid of week 13**	**Mid of week 14**
Lab 7 Adjacency Matrix	**Mid of week 14**	**Mid of week 15**

Group Assignments with topics(tentative)

GAs	Posted (Tuesdays)	Due (Tuesdays)
GA1 on linked list as Arrays and recursion and stacks	**Mid of week 4**	**Mid of week 7**
GA2 on sorting with clock and queue	**Mid of week 8**	**Mid of week 11**
GA3 on Trees and/or graphs	**Mid of week 11**	**Mid of week 14**

Resubmission Policy

Students can resubmit each homework for a regrade in the next 48 hours after the deadline with 20% penalty. Students do not need to notify anybody; the homework will be regraded automatically.

Regrading Policy

No HW will be dropped, however you can repeat any previous homework for a full grade, through final regrading.

Extra programs

Students can submit any program done in the class/lab or from the end of the book chapter as extra program and upload to the server under separate folders.

Each extra program should be in a separate folder's name named as **extra_m_w_n**. (m means the month when the extra was given, should between 1-5. w means the week of the extra was given, should between 1-5. n means the number of the extra, if only one extra was given in that week, use 1, if two were given, use 1, 2, like extra_2_1_1 and extra_2_1_2. The order is important.)

All the extras should be **under the root of your Linux server** **(upload .cpp and .h files)**

For example the dayType is extra_1_3_1

Overloading extra_1_3_2

Linetype extra_1_3_3

Extras are NOT included in the Grading system

Extras are useful when the grade is on the upper end of an interval. For example, if a student has 82.9 upper end of B- and who has done many programs as extras, the grade might be pushed to B. However, if the student grade is 79.5 EXTRAS WILL NOT BE USEFUL

CAPS

Counseling and Psychological Services (CAPS) can help students who are having difficulties managing stress, adjusting to college, or feeling sad and hopeless. You can reach CAPS (www.uh.edu/caps) by calling 713-743-5454 during and after business hours for routine appointments or if you or someone you know is in crisis. No appointment is necessary for the “Let's Talk” program, a drop-in consultation service at convenient locations and hours around campus. http://www.uh.edu/caps/outreach/lets_talk.html

Wishing you a pleasant and a fruitful semester