USMAN INSTITUTE OF TECHNOLOGY

CS211 – Data Structures and Algorithms
Credit Hours: 3 + 1
Fall 2019

People and Class Times:

Course Instructor: Muhammad Qasim Pasta
Office: Faculty Room, 3rd Floor
Email: mqpasta@uit.edu
Office Hours: Mon 11:30 – 12:25 pm, Wed 01:30 - 02:25 pm, Thu 09:30 – 10:25 am

Lab Instructor: Ms. Asma Idress Mala
Office: Faculty Room, Ground Floor (adjacent to lift)
Email: TBA
Office Hours: TBA

Course Webpage: https://piazza.com/uit/fall2019/cs211/home OR http://tiny.cc/ds-fall19

Registration URL: http://tiny.cc/ds-register

Course Objectives:

Data Structures are at the heart of all efficient software. While transparent to the user, the choice of data structure plays a crucial role in the adoption of software. In addition, the study and design of data structures pose an interesting intellectual challenge.

This course has two main objectives: to make students proficient in the use of common data structures and to introduce students to the implementation of common data structures. This is a programming intensive course.

Specifically, this course aims to:

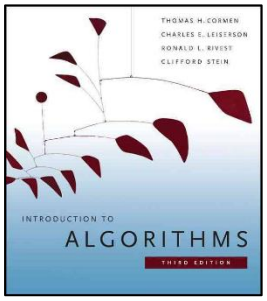
- impart proficiency in basic data structures and related operations
- introduce higher-level data structures
- develop critical judgment regarding the choice of data structures for a given situation
- develop skills to implement basic data structures using any high-level language

Course Learning Outcome:

At the end of the course the students will be able to:

1. Implement various data structures and their algorithms, and **apply** them in implementing simple applications. (Domain C, BT = 2,3)
    - perform basic operations on elementary data structures: list, tree, stack, queue.
    - perform basic operations on higher level data structures: trees, and graph
    - implement the above data structures and supporting operations in any high level language
2. **Analyze** simple algorithms and determine their complexities. (Domain C, BT = 4,5)
3. **Apply** the knowledge of data structures to other application domains (Domain C, BT = 3)
    - choose an efficient data structure for a given problem
    - reason about the trade-off between data structure efficiency and programmer effort
4. **Design** new data structures and algorithms to solve problems (Domain C, BT = 6)
    - implement a moderately complex project in any high level language

Course Texts



Text Book:

1. Thomas H. Cormen, Introduction to Algorithms, MIT Press, 3rd Edition
   (Image source: Wikipedia Articles)

Reference Books:

1. Pat Morin, Open Data Structures, Edition 0.1G beta (available online for free)
2. Michael T. Goodrich, Data Structures and Algorithms in Python, 1st Edition

Reference Material:

1. Data Structure and Algorithms in Python, Udacity (free online course)

Grading Procedures

Theory Component	100 Marks	
	Midterm Examination	20%
	Presentation	05%
	Assignments (4)	05%
	Quizzes (4)	10%
	Final	60%
Lab Component	50 Marks	
	Lab Files	10%
	Lab Exam I	20%
	Lab Exam II	30%
	Lab Quizzes (5)	20%
	Project	20%

Project

Each group will build a small project to demonstrate the ability to solve a real-world problem using data structure. The group will decide a topic and submit a brief proposal. On the approval of the proposal, group will start working on the project. The final submission will be before lab exams. A viva will be conducted to evaluate the performance of individual group members.

Presentation

Each group has to make a presentation from given list of advances data structures. The presentation will highlight the working and usage of respective data structure.

Academic Integrity

Each student in this course is expected to make sure that any work submitted by a student in this course for academic credit will be the **student's own work**. Scholastic dishonesty shall be considered a serious violation of these rules and regulations and is subject to strict disciplinary action. Scholastic dishonesty includes, but is not limited to, cheating on exams, plagiarism on assignments, and collusion.

PLAGIARISM: Plagiarism is the act of taking the work created by another person or entity and presenting it as one's own for the purpose of personal gain or of obtaining academic credit. Plagiarism includes the submission of or incorporation of the work of others without acknowledging its provenance or giving due credit according to established academic practices. This includes the submission of material that has been appropriated, bought, received as a gift, downloaded, or obtained by any other means. Students must not, unless they have been granted permission from all faculty members concerned, submit the same assignment or project for academic credit for different courses.

CHEATING: The term cheating shall refer to the use of or obtaining of unauthorized information in order to obtain personal benefit or academic credit.

COLLUSION: Collusion is the act of providing unauthorized assistance to one or more person or of not taking the appropriate precautions against doing so. Any student caught violating academic integrity will suffer an academic penalty. All violations of academic integrity will also be immediately reported to the Disciplinary Committee. Any student violating academic integrity a second time in this course will receive a failing grade for the course, and additional disciplinary sanctions may be administered through the Disciplinary Committee.

Conclusively, each student need to be take care of:

1. You must not share your solutions with other students. You are encouraged to discuss the problems but each student is supposed to take care of his or her own solution.
2. You must not submit solution of other students as yours.
3. You must duly cite all resources you used in development of your solution.

Tentative Course Schedule

Week	Lectures	Reading	Lab	Assignment
1.	Introduction to Class, Revision, Introduction to Arrays	TBA	Revision of basic programming constructs	Asg. 0: Release
2.	Linear Searching, Binary Searching, Sorting Algorithms (Bubble sort and Insertion sort)	TBA	Revision of Types and Objects	Asg. 1: Due Asg. 1: Release
3.	Big O Notation and Analysis of algorithms	Ch#3, Coreman	Implementation of searching algorithm	
4.	Sorting Algorithms (Selection Sort), Multidimensional Array, Sparse Matrix	TBA	Implementation of sorting algorithms	Asg. 1: Due
5.	Introduction to Stack, Stack Operations, Introduction to Queue, Queue operations	Ch#10, Coreman	Matrix handling	Asg. 2: Release
6.	Intro to Linked List, Linked List Representation and operations	Ch#10,Coreman	Implementation of Stack and Queue operations	Asg. 2: Due
7.	Double Linked List and Circular Linked List	Ch#10,Coreman	Implementation of Linked List	
8.	Recursion and its implementation , Algorithms using recursion (Quick Sort), Algorithms using recursion (Merge Sort)	Ch#7, Coreman	Implementation of Quick Sort	Submission of Project Topics
9.	**Midterm Exam**			
10.	Tree Structures, n-trees, binary tree, traversing in binary tree, binary search tree (BST), insertion and deletion in BST	Ch#12,Coreman	**Lab Exam I**	Asg. 3: Release
11.	Graph theory, adjacency matrix, depth first search, breadth first search	Ch#22,Coreman	Implementation of Tree traversing and BST	Asg. 3: Due Asg. 4: Release
12.	Shortest path algorithms: Warshall's algorithm, Dijkstra's algorithm, Strongly connected components,	Ch#24,Coreman	Implementation of BFS and DFS	Asg. 4: Due
13.	String Operations, word processing string, pattern matching	Ch#32,Coreman	Implementation of Dijkstra's algorithm	Poster Session
14.	minimum spanning tree, Union find Sets cover, Huffman coding,	Ch#23,Coreman	Implementation of String algorithms	
15.	Presentation		Project	
16.	Presentation		Project	Project Submission
17.	Presentation		**Lab Exam II**	
Theory Exam				