

CS2700: Programming and Data Structures

July Nov 2021

C Slot; **Slots: Mon (10.00am – 10.50am); Tue (9.00am – 9.50am);
Wed (8.00am – 8.50am); Fri (12.00pm – 12.50pm)**

Meghana Nasre

Email: meghana@iitm.ac.in

Updated on August 11, 2021

1 Course objectives

The objective of the course is to teach programming (with an emphasis on problem solving) and introduce elementary data structures. The student should, at a rudimentary level, be able to prove correctness (loop invariants, conditioning, etc) and analyze efficiency (using the 'O' notation).

2 Learning Outcomes

- Design correct programs to solve problems.
- Choose efficient data structures and apply them to solve problems.
- Analyze the efficiency of programs based on time complexity.
- Prove the correctness of a program using loop invariants, pre-conditions and post-conditions in programs.

3 Mode of Teaching

Live Lectures which are also recorded. The lectures are conducted on Cisco Webex.

4 Textbooks

Data Structures and Algorithm Analysis in C / C++ by Mark Allen Weiss.

5 Course Requirements

You are *required* to attend all the lectures. Class participation is strongly encouraged to demonstrate an appropriate level of understanding of the material being dis-

cussed in the class. Regular feedback from the class regarding the lectures will be very much appreciated.

6 Planned Syllabus

- Review of Problem Solving using computers, Abstraction, Elementary Data Types. Algorithm design- Correctness via Loop invariants as a way of arguing correctness of programs, preconditions, post conditions associated with a statement.
- Complexity and Efficiency via model of computation (notion of time and space), mathematical preliminaries, Elementary asymptotics (big-oh, big-omega, and theta notations).
- ADT Array – searching and sorting on arrays: Linear search, binary search on a sorted array. Bubble sort, Insertion sort, Merge Sort and analysis; Emphasis on the comparison based sorting model. Counting sort, Radix sort, bucket sort.
- ADT Linked Lists, Stacks, Queues: List manipulation, insertion, deletion, searching a key, reversal of a list, use of recursion to reverse/search. Doubly linked lists and circular linked lists.
- Stacks and queues as dynamic data structures implemented using linked lists. Analyse the ADT operations when implemented using arrays. ADT Binary Trees: Tree representation, traversal, application of binary trees in Huffman coding. Introduction to expression trees: traversal vs post/pre/infix notation. Recursive traversal and other tree parameters (depth, height, number of nodes etc.)
- ADT Dictionary: Binary search trees, balanced binary search trees - AVL Trees. Hashing - collisions, open and closed hashing, properties of good hash functions.
- ADT Priority queues: Binary heaps with application to in-place sorting

- Graphs: Representations (Matrix and Adjacency List), basic traversal techniques: Depth First Search + Breadth First Search (Stacks and Queues)

7 Tentative Grading Policy

The following allocation of points is tentative.

Quiz 1 : Sept 4, 2021, Saturday	25%
Quiz 2 : Oct 9, 2021, Saturday	25%
Quiz 3 : Nov 21, 2021, Sunday	25%
Tutorials (6 : best 5 out of 6):	25%
(Aug 16, Aug 27, Sept 17, Oct 1, Oct 22, Nov 5).	

All tutorials are in class. The tutorial dates may be moved and announced based on a logical break in the topics covered.

8 Academic Honesty

Academic honesty is expected from each student participating in the course. NO sharing (willing, unwilling, knowing, unknowing) of assignment code between students, submission of downloaded code (from the Internet, geeksforgeeks, or anywhere else) is allowed.

Academic violations will be handled by IITM Senate Discipline and Welfare (DISCO) Committee. Typically, the first violation instance will result in ZERO marks for the corresponding component of the Course Grade and a drop of one- penalty in overall course grade. The second instance of code copying will result in a 'U' Course Grade and/or other penalties. The DISCO Committee can also impose additional penalties.