FR. Conceicao Rodrigues College Of Engineering

Department of Computer Engineering

S.E. (Computer) (semester III)

(2019-2020)

Course Outcomes and Assessment Plan

Subject: Data Structures **Credits-4**

Course code: CSC305

Syllabus:

1. **Introduction to Data Structures**
   Introduction, Types of Data Structure-linear and non linear, operations on data structures
   Concept of ADT (Abstract Data type), Arrays

2. **Stack and Queues**
   Introduction, ADT of Stack, Operations on stack, Array implementation of stack,
   Applications of stack-Well form-ness of parenthesis, infix to postfix conversion,
   evaluation of postfix expression, Recursion
   ADT of Queue, operations on Queue, Array implementation of Queue, Circular Queue,
   Priority Queue, Double ended queue, Applications of Queue

3. **Linked List**
   Introduction, representation of Linked List, Linked List v/s Array, implementations of Linked list, Linked implementation of Stack and Queue, Circular Linked List, Doubly linked list, Application- Polynomial representations and addition

4. **Trees**
   Introduction, tree terminologies, Binary tree, representation, Types of Binary tree, binary tree traversals, Binary search Tree, implementations of binary search tree
   Applications: Huffman encoding, Expression tree, Search trees-AVL,B tree ,B+ tree, Splay tree, Trie

5. **Graphs**
   Introduction, Graph terminologies, Representation, Graph traversals- Depth First Search (DFS) and Breath First Search (BFS) Application-Topological Sorting

6. **Sorting AND Searching**
   Introduction, Bubble sort, Insertion sort, Merge Sort, Quick Sort, Linear search,
   Binary search, Hashing- concepts, hash functions, collision handling
   techniques.

Text Books:

1. Data structures using C, Reema Thareja,Oxford
2. Data Structures using C and C++,Rajesh shukla,Wiley- India
3. Data structures using C, Aaron Tenenbaum, Yedidyah Langsam, Moshe Augustin,Pearson
4. Data structures: A pseudocode approach with C, Richard F. Gilberg & Behrouz A,Fourozen,second edition,CENGAGE learning
5. Introduction to data structure and its applications, JeanPaul Tremblay, P.G. Sorenson

Reference books:

1. C & Data structures, Prof. P.S. Deshpande, Prof. O.G. Kakde, Dreamtech press
2. Data Structures using C, Balaguruswami

3. Data Structures using C, ISRD Group, Second edition,Tata McGraw Hill
4. Data Structures, adapted by: GAV PAI, Schaum's Outlines

Course Objectives:

1. To teach various storage mechanism of data.
2. To design and implement various data structures
3. To introduce various techniques for representation of the data in the real world
4. To teach different sorting techniques.
5. To teach different searching techniques.

Course Outcomes:

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

CSC 305.1	Implement various linear data structures.	(Application)
CSC 305.2	Implement various non linear data structures.	(Application)
CSC 305.3	Select appropriate sorting and searching technique for a given problem and apply them.	(Applicatio
CSC 305.4	Develop solution for real world problems by selecting appropriate data structure and algorithms.	n) (Applicatio n)

Program Outcomes (POs)

Engineering Graduates will be able to

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/Development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling of complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms

of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project Management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognized the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

(A) Program Specific Outcomes (PSOs)

Student will have ability to

1. Apply fundamental computer science knowledge to solve real world problems.
2. Design and implement software systems of varying complexity in multidisciplinary scenarios that meet specified requirements with appropriate consideration to architectural, algorithmic and security aspects.

Mapping of CO and PO/PSO

Relationship of course outcomes with program outcomes: Indicate 1 (low importance), 2 (Moderate Importance) or 3 (High Importance) in respective mapping cell.

	PO1 (Engg Know)	PO2 (Analysis)	PO3 (Design)	PO4 (investiga)	PO5 (tools)	PO6 (engg Soci)	PO7 (Env)	PO8 (Eth)	PO9 (ind Team)	PO10 (comm.)	PO11 (PM)	PO12 (life Long)	PSO1	PSO2
CSC305.1	3											3	3	2
CSC305.2	3											3	3	2
CSC305.3	3	1										3	3	2
CSC305.4	3	3	2						3		1	3	3	2
Course To PO														

Justification

PO1: CSC305.1, CSC305.2, CSC305.3 and CSC305.4 maps to PO1 because engineering graduates will be able to use linear, non linear data structures and algorithms to provide a solution to complex engineering problems.

PO2: CSC 305.3 maps to PO2 because engineering graduates will be able to analyze the problem to identify which sorting and searching method can be used. CSC305.4 also maps to PO2 as engineering graduates will be able to analyze a problem to make a decision about which data structure and algorithm is appropriate.

PO3: CSC 305.4 maps to PO3 since students will be able to develop solution to real world problems with the help of data structure and searching/sorting algorithm.

PO9: CSC 305.4 maps to PO9 because engineering graduates will be able to develop a solution to

real world problem by working in a team.

PSO1: CSC305.1, CSC305.2, CSC305.3 and CSC305.4 maps to PSO1 as Engineering graduates will be able to apply knowledge learned in data structures subject to provide solution to real world problems.

PSO2: CSC305.1, CSC305.2, CSC305.3 and CSC305.4 maps to PSO2 as Engineering graduates will be able to implement a software systems or processes with the help of various data structures and algorithms.

CO Assessment tools:

1) CSC305.1 Implement various linear data structures. Target: 2.5

Direct Method	Weightage	Target	Marks
Unit Test 1	0.25	70% of students will minimum score 70% marks	20 M
Quiz 1	0.1	75% of students will minimum score 60% marks	20M
University Exam (Theory + oral)	0.3	70% of students will minimum score 60% marks	105M
Assignment 1	0.1	75% of students will minimum score 70% marks	10M
Lab Performance	0.25	75% of students will minimum score 75% marks	100M
Indirect Method	**Weightage**	**Target**	
Course Exit Survey	1	85% of students strongly agree and agree	

CSC305.1:

a) **Direct Methods (80%):** Unit Test 1+Quiz 1+ UniExam+ Assignment+ Lab performance

CSC305.1dm = 0.25UT1 +0.1Q1+ 0.3Uniexam+ 0.1Assignment +0.25Lab

b) **InDirect Methods(20%):** Course exit survey

CSC305.1*idm*

CSC305.1 = 0.8* CSC305.1dm + 0.2* CSC305.1idm

2) CSC305.2 Implement various non linear data structures. Target: 2.5

Direct Method	Weightage	Target	Marks
Unit Test 2	0.25	70% of students will minimum score 70% marks	14 M
Quiz 2	0.1	75% of students will minimum score 60% marks	20M

University Exam (Theory + oral)	0.3	70% of students will minimum score 60% marks	105M
Assignment 2	0.15	75% of students will minimum score 70% marks	10M

Lab Performance	0.2	75% of students will minimum score 75% marks	20M
Indirect Method	**Weightage**	**Target**	
Course Exit Survey	1	85% of students strongly agree and agree	

CSC305.2:

a) **Direct Methods (80%):** Unit Test 2+Quiz 2+ UniExam+ Assignment+ Lab performance

CSC305.2dm = 0.25UT2 +0.1Q2+ 0.3Uniexam+ 0.15Assignment +0.2Lab

b) **InDirect Methods(20%):** Course exit survey

CSC305.2*idm*

CSC305.2 = 0.8* CSC305.2dm + 0.2* CSC305.2idm

3) CSC305.3 Select appropriate sorting and searching technique for a given problem and apply them. Target: 2.7

Direct Method	**Weightage**	**Target**	**Marks**
Unit Test 2	0.2	70% of students will minimum score 60% marks	6 M
Mini Project	0.15	75% of students will minimum score 70% marks	20M
University Exam (Theory + oral)	0.3	70% of students will minimum score 60% marks	105M
Case Study	0.15	75% of students will minimum score 70% marks	10M
Lab Performance	0.2	75% of students will minimum score 75% marks	20M
Indirect Method	**Weightage**	**Target**	
Course Exit Survey	1	85% of students strongly agree and agree	

CSC305.3:

a) **Direct Methods (80%):** Unit Test 2+Mini Project + UniExam+ Case study+ Lab performance

CSC305.3dm = 0.2UT2 +0.15MP+ 0.3Uniexam+ 0.15casestudy +0.2Lab

b) **InDirect Methods(20%):** Course exit survey

CSC305.3*idm*

CSC305.3 = 0.8* CSC305.3dm + 0.2* CSC305.3idm

3) CSC305.4 Develop solution for real world problems by selecting appropriate data structure and algorithms.

Target: 2.8

Direct Method	**Weightage**	**Target**	**Marks**
Demonstration of Mini Project	0.75	75% of students will minimum score 70% marks	20M
Report	0.25	75% of students will minimum score 70% marks	
Indirect Method	**Weightage**	**Target**	
Course Exit Survey	1	75% of students strongly agree and agree	

CSC305.4:

a) Direct Methods (80%): Mini Project + Report

CSC305.4dm = 0.75MiniProject+0.25Report

b) InDirect Methods(20%): Course exit survey

CSC305.4*idm*

CSC305.4 = 0.8* CSC305.4dm + 0.2* CSC305.4idm

Course Outcomes Target:

CSC 305.1	Implement various linear data structures. **Target level: 2.5**
CSC 305.2	Implement various non linear data structures. **Target level:2.5**
CSC 305.3	Select appropriate sorting and searching technique for a given problem and apply them. **Target level: 2.7**
CSC 305.4	Develop solution for real world problems by selecting appropriate data structure and algorithms. **Target level: 2.8**

CO Attainment of previous years

Course Outcomes	**2017-18**
CSC 305.1: Implement various linear data structures.	1.96
CSC 305.2: Implement various nonlinear data structures.	2.12

CSC 305.3 Select appropriate sorting and searching technique for a given problem and apply them.	2.52
CSC 305.4: Develop solution for real world problems by selecting appropriate data structure and algorithms.	2.52

Course Outcomes	**2016-17**	**2015-16**	**2014-15**
1) Apply advance C programming techniques such as pointers, dynamic memory allocation, structures, recursion to developing solutions for particular problems.	1.96	1.8	2.24
2) Design and implement abstract data types such as linked list, stack, queue, tree and graph by using C as the programming language using static or dynamic implementations.	2.28	1.8	1.88
3) Apply operations like searching, sorting, insertion, deletion, traversing mechanism etc. on various data structures.	2.28	2.04	1.88
4) Analyze, evaluate and choose appropriate abstract data types and algorithms to solve particular problems.	2.28	2.28	1.88

Rubrics for Lab Experiments:

Sr. No	Performance Indicator	Excellent	Good	Satisfactory	Unsatisfactory
1)	Completeness and correctness [4]	Well commented and formatted, program functions correctly for all input cases. [4M]	Comparatively less use of comments, Inconsistent formatting. Program functions correctly for all input cases.[3M]	Inconsistent comments and formatting. Program functions correctly for most of the input cases.[2M]	Improper formatting, No comments. Program functions correctly for very limited cases [1M]
2)	Efficiency [3]	The code could be reused as a whole or each routine could be reused. It is readable and easy to understand [3M]	Most of the code could be reused in other programs. It is fairly readable and easy to understand [2M]	Only Some parts of the code could be reused in other programs. The code is unnecessarily long and repeated. [1M]	The code lacks reusability. It is huge and repeated at many places[0M]
3)	Post Lab Questions [2]	Answers to all questions are correct and explained in depth. [2M]	Answers to most of the questions are correct but not explained in much depth. [2-1.5M]	Answers of few questions are incorrect and lacks sufficient depth [0-1M]	Answers to most of the questions are incorrect and not explained in depth. [0 mark]
4)	Promptness [1]	The laboratory report is submitted on time [1 mark]	The laboratory report is submitted next day. [0.5 marks]	The laboratory report is submitted in next practical session. [0 marks]	

Rubrics for Assignments:

Performance Indicator	Excellent	Good	Below average
Timeline(2)	submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)
Organization (2)	Well organized, neat and clear handwriting, easy to read.(2)	Organized to some extent, handwriting is neat(1)	Poorly organized(0.5)
Level of content (4)	All points are covered and answered accurately (4)	Some important points are omitted /addressed minimally (3)	Many important points are missing and the answers are not accurate. (2)
Depth and breadth discussion (2)	Each point is illustrated in depth with proper justification wherever required (2)	Few points are not illustrated in depth and have minimal justification (1)	Many points are not illustrated in depth and justification missing.(0.5)

Rubrics for Mini Project:

Contents required in report are:

1) Problem statement
2) Functionalities /features supported by project
3) Description about selected data structure and reason to select it
4) Any special algorithm or method used for implementation
5) Code and output of the program

Total Marks: 10

Sr. No	Performance Indicator	Excellent	Good	Satisfactory	Unsatisfactory
1)	Completeness and correctness [4]	Project is complete. All necessary functionalities are supported. works correctly for all input cases [4M]	Project is complete. Most of the necessary functionalities are supported. Missing few minor details. works correctly for all input cases [3M]	Project is complete. Only major functionalities are supported. Missing minor details. Program functions correctly for most of the input cases.[2M]	Project is incomplete. Supports very few functionalities, Missing many details. Program functions correctly for few input cases.[1M]
2)	Team work [2]	Each member of the team	Each member of the	Only two members	team lacks

		contributed equally and worked efficiently[2]	team contributed almost equally and worked efficiently[1.5]	of the team contributed [1]	contribution by each member [0.5]
3)	Report [3]	Content is very informative and accurate. It is easy to read and understand. [3M]	Content is informative and mostly accurate. It is readable.[2M]	Content is not always related to topic. It is less readable due to poor sequencing and sentence structure. [1M]	Content is not relevant or accurate. It is unreadable and difficult to understand due to illogical

4)	Promptn e ss [1]	The project work submitted on time [1 mark]	The project work is submitted on the next day. [0.5 marks]	The project work is submitted late. [0 marks]	

Lesson Plan: DS
Semester III **Year: 2018-19**

Modes of Content Delivery:

I	Class Room Teaching	v	Self Learning Online Resources	Ix	Industry Visit
Ii	Tutorial	vi	Slides	X	Group Discussion
Iii	Remedial Coaching	vii	Simulations/Demonstrations	xi	Seminar
Iv	Lab Experiment	viii	Expert Lecture	xii	Case Study

1. Introduction to data structures

Lecture No.	Portion to be covered per hour	Planned Date	Actual Date	Content delivery method/Learning activity
1	What is data? What is meant by data structure? Types of data structures ,introduction to linear data structures ,examples of linear and non- linear data structures	1-7-2019	1-7-2019	Chalk and Board
2	Implementation of stack using global array & top variable- push(),pop()	2-7-2019	3-7-2019 (holiday declared on 2/7 due to heavy rain)	Chalk and Board
3	Implementation of stack continued display(), isfull(), isempty() function, introduction to memory allocation, static memory allocation	3-7-2019	5-7-2019	Chalk and Board
4	Introduction to dynamic memory allocation, difference/advantages/disd vantages of static	5-7-2019	8-7-2019	Chalk and Board

	and dynamic and dynamic memory allocation functions			

5	Dynamic memory allocation functions continued, program calculating sum and average of an array using dynamic memory allocation	8-7-2019	9-7-2019	Chalk and Board

Text Books/ Reference Books:

Te x t B o o k s : 1) Data structures using C, Reema Thareja,Oxford
2)Data structures using C, Aaron Tenenbaum, Yedidyah Langsam, Moshe

Augustin,Pearson Reference Books: 1) Data Structures using C, Balaguruswami

Chapter 2. Stack and Queues

Lecture No.	Portion to be covered per hour	Planned Date	Actual Date	Content delivery method/Learning activity
6	Array implementation of stack by creating a structure stack. Perform push(), pop() operations by passing structure variable(call by value mechanism), introduction to stack implementation using call by reference	9-7-2019	10-7-2019	Chalk and Board, Lab experiment, ppt, visualization
7	Call by reference continued , program on two stacks in an array	10-7-2019	12-7-2019	Chalk and Board, Lab experiment,
8	applications of stack DS 1) reversing a string 2) Introduction to Infix, postfix and prefix form of expressions, examples on it	12-7-2019	15-7-2019	Chalk and Board, Lab experiment
9	Algorithm & program to check balanced parentheses, algorithm of postfix evaluation	15-7-2019	17-7-2019	Chalk and Board, Lab experiment, PPT, visualization
10	Conversion from infix to postfix example, Program of Evaluating postfix expression, Conversion from infix to postfix	17-7-2019	18-7-2019	Chalk and Board, lab experiment, PPT, Visualization

11	Conversion from infix to prefix example, algorithm and program., Prefix to postfix conversion algo, Postfix to prefix conversion algo	18-7-2019	19-7-2019	Chalk and Board, lab experiment, PPT, Visualization

12	Introduction to Queue, operations on queue, array implementation of linear queue	19-7-2019	24-7-2019	Chalk and Board, lab experiment
13	drawback of linear queue, Introduction to circular queue, operations on circular queue, Implementation of circular queue using array	24-7-2019	25-7-2019	Chalk and Board, lab experiments
14	Implementation of circular queue using modulo operator logic	25-7-2019	29-7-2019	Chalk and Board, lab experiments

Text Books/ Reference Books:

Te x t B o o k s : 1) Data structures using C, Reema Thareja,Oxford

2)Data structures using C, Aaron Tenenbaum, Yedidyah Langsam, Moshe Augustin,Pearson

Reference Books: 1) Data Structures using C, Balaguruswami

Chapter 3. Linked list

Lecture No.	Portion to be covered per hour	Planned Date	Actual Date	Content delivery method/Learning activity
15	Introduction to linked list, representation of linked list in C. function to create a linked list	29-7-2019	31-7-2019	Chalk and Board, lab experiments
16	Inserting new nodes to linked list, insertion in the beginning, insertion at the end, insertion before a node	31-7-2019	1-8-2019	Chalk and Board, lab experiments,visualization
17	Insertion after a specific node, deletion of a node from linked list, traversal of linked list, sorting LL	1-8-2019	1-8-2019	Chalk and Board, lab experiments, visualization,
18	Linked list v/s array implementation, introduction to circular linked list, operations on circular linked list	2-8-2019	2-8-2019	Chalk and Board, lab experiments, visualization

19	Linked implementation of stack and queue	5-8-2019	5-8-2019	Chalk and Board, lab experiments Think pair share
20	Introduction to Doubly linked list, creation of doubly linked list, insertion of node in doubly linked list	7-8-2019	7-8-2019	Chalk and Board, lab experiments, visualization

21	deletion of a node from a doubly linked list, traversal of doubly linked list	9-8-2019	9-8-2019	Chalk and Board, lab experiments, visualization, Think pair share
22	Polynomial addition using linked list, sparse matrix representation using linked list	19-8-2019	9-8-2019	

Te x t B o o k s : 1) Data structures using C, Reema Thareja,Oxford
2)Data structures using C, Aaron Tenenbaum, Yedidyah Langsam, Moshe Augustin,Pearson

Reference Books: 1) Data Structures using C, Balaguruswami

Chapter 4. Trees

Lecture No.	Portion to be covered per hour	Planned Date	Actual Date	Content delivery method/Learning activity
23	Introduction, tree terminologies, Binary tree, representation, Types of binary tree	1-8-2019	19-8-2019	Chalk and Board, Think Pair share activity
24	Binary search tree, implementation of binary search tree	21-8-2019	21-8-2019	Chalk and Board, lab experiments
25	Implementation of binary search tree continued	22-8-2019	22-8-2019	Chalk and Board, lab experiments
26	Binary search tree traversals	23-8-2019	23-8-2019	Chalk and Board, lab experiments
27	Expression tree construction and program	26-8-2019	26-8-2019	Chalk and Board, lab experiments,visualization
28	Huffman encoding	29-8-2019	29-8-2019	Chalk and Board, lab experiments,visuali

				zation
29	AVL tree	9-9-2019	9-9-2019	Chalk and Board, lab experiments,visualization
30	B tree	11-9-2019	11-9-2019	Chalk and Board,visualization
31	B+ tree	13-9-2019	13-9-2019	Chalk and Board,visualization
32	Splay tree	16-9-2019	16-9-2019	Chalk and Board,visualization
33	trie	18-9-2019	18-9-2019	Chalk and Board,visualization

Text Books/ Reference Books:

Te x t B o o k s : 1) Data structures using C, Reema Thareja,Oxford

2)Data structures using C, Aaron Tenenbaum, Yedidyah Langsam, Moshe

Augustin,Pearson Reference Books: 1) Data Structures using C, Balaguruswami

Chapter 5 : Graphs

Lecture No.	Portion to be covered per hour	Planned Date	Actual Date	Content delivery method/Learning activity
34	introduction, Graph terminologies, representation	19-9-2019	19-9-2019	Chalk and Board
35	Depth first Search Algorithm and program	20-9-2019	20-9-2019	Chalk and Board, lab experiments
36	Breath first Search Algorithm and program	23-9-2019	23-9-2019	Chalk and Board, lab experiments
37	Topological sorting	26-9-2019	26-9-2019	Chalk and Board

Text Books/ Reference Books:

Te x t B o o k s : 1) Data structures using C, Reema Thareja,Oxford

2)Data structures using C, Aaron Tenenbaum, Yedidyah Langsam, Moshe

Chapter 6. Sorting and Searching

Lecture No.	Portion to be covered per hour	Planned Date	Actual Date	Content delivery method/Learning activity
38	Insertion sort	30-9-2019	30-9-2019	Chalk and Board, lab experiments,simulation
39	Merge Sort	1-10-2019	1-10-2019	Chalk and Board, lab experiments, simulation
40	Quick sort	3-10-2019	3-10-2019	Chalk and Board, lab experiments, simulation
41	Hashing concept, hash functions, collision handling techniques	9-10-2019	9-10-2019	Slides, Chalk and Board
42	Collision handling techniques contd..	10-10-2019	10-10-2019	slides Chalk and Board

Text Books/ Reference Books:

Te x t B o o k s : 1) Data structures using C, Reema Thareja,Oxford
2)Data structures using C, Aaron Tenenbaum, Yedidyah Langsam, Moshe

Augustin,Pearson Reference Books: 1) Data Structures using C, Balaguruswami

Subject: Data Structures Lab **Credits-1**

Course code:

CSL303 Syllabus:

1) Array implementation of stack. *
2) Conversion of Infix to Postfix. *
3) Evaluation of Postfix expression.
4) Check continuity of different types of parenthesis using stack.
5) Array implementation of Queue.
6) Array implementation of Circular Queue *
7) Array implementation of Priority Queue.
8) Implementation of Singly linked list.*
9) Linked implementation of Stack.
10) Linked implementation of Queue.
11) Implementation of Circular Linked list.
12) Implementation of Doubly Linked list.
13) Implement Binary Search Tree. *
14) Implementation of Bubble Sort.
15) Implementation of Insertion Sort.
16) Implementation of Merge Sort.
17) Implementation of Quick Sort.*
18) Implementation of Binary Search.*
20) Implementation of Hashing.

21) Implementation of Depth First Search and Breath First Search.

Term Work (25M): Lab Experiments (10M) + Mini Project (5M) + Assignment(5M)

List of Practicals

Sub: Data Structures **Year (2019-20)**

1.	1.1 Static implementation of Stack data structure 1.2 Two Stacks in an Array	First Week
2.	Application of Stack data structure: 2.1 Postfix expression Evaluation. 2.2 Infix to Reverse Polish Notation (Infix to Postfix) 2.3 Infix to Polish Notation (Infix to Prefix)	Second Week
3.	3.1 Static Implementation of Linear Queue data structure 3. 2 Static Implementation of Circular Queue with sacrificing an element	Third Week
4.	Linked List Implementation	Fourth Week
5.	5.1 Sparse Matrix Implementation using LL 5.2 Polynomial Operations using LL	Fourth Week
6.	6.1 Dynamic implementation of Stack 6.2 Dynamic implementation of Queue	Fifth Week
7.	Implementation of Circular Linked List	Fifth Week
8.	Implementation of Doubly Linked List	Sixth Week
9.	Implementation of Priority Queue	Sixth Week
10	Static implementation of Dequeue	Eighth Week
11	11.1 Implementation of BST(insertion, deletion, traversal) 11.2 Expression Tree	Ninth Week
12	Graph Creation. BFS AND DFS Traversal	10^{th} Week
13	Sorting Techniques 13.1 Bubble , Selection and Insertion Sort 13.2 Quick Sort and Merge Sort	10^{th} Week 11^{th} Week
14	Searching Techniques 14.1 Linear Search 14.2 Binary Search	11^{th} Week

Course Outcomes (given in syllabus):

Students will be able to:

1. Implement various linear and non linear data structures
2. Handle operations like insertion, deletion, searching and traversing on various data structures.
3. select appropriate sorting technique for given problem.
4. select appropriate searching technique for given problem.
5. Apply the learned concepts in various domains like DBMS and Compiler construction.
6. Choose appropriate data structure for specified problem domain

Content beyond syllabus:

1) Heap sort
2) Radix sort

Curriculum gap: heap memory allocation

Assignment 1

Sub: Data Structures Class: SE Comps
Assignment Date: 27th July 2019
Date of submission: 10th August 2019

Course outcome: CSC305.1: Co1: Implement various linear data structures

Q. 1) Write a program to simplify an algebraic string of characters having '+', '-' operators and parenthesis. Print the simplified equation without the parenthesis modify the operators while removing parentheses.

Input: String S **Output:** Simplified version of string S

NOTE: do not omit the variables if they are getting canceled out, print them as it is
Example: Input: a-(b+c)
Output: a-b-c
Constraints: 1 <= S <= 100

Sample Test Cases

	Input	Output
Test Case 1	a-(b-c-(d+e))-f	a-b+c+d+e-f
Test Case 2	a-b-(c-d)	a-b-c+d
Test Case 3	b-(c-(a+d))	b-c+a+d
Test Case 4	b+(a-(c-d-(f+k-(g+q))))	b+a-c+d+f+k-g-q
Test Case 5	a-b-c-(g+k-(r))	a-b-c-g-k+r
Test Case 6	c+(d-(f-(e-(r-(s-k-i-o)))))	c+d-f+e-r+s-k-i-o

Test Case 7	f-(r-(q-w-e-r-t-y-u-i-o))	f-r+q-w-e-r-t-y-u-i-o
Test Case 8	l+m+n-(q+r+t-(o-p-h-f))	l+m+n-q-r-t+o-p-h-f
Test Case 9	r+t-(q-w-e-r-(t-y-u-i-(o-p)))	r+t-q+w+e+r+t-y-u-i-o+p

2) Write a program to sort elements of an array using stack.

I/P: Input : 8 5 7 1 9 12 10
Output : 1 5 7 8 9 10 12

3) Write a program to implement stack data structure using queue. Make use of enque( ) and deque( ) operation of queue.

4) Given a queue of integers of even length, rearrange the elements by interleaving the first half of the queue with the second half of the queue. We are allowed to use only queue data structure.

Examples:

Input : 1 2 3 4
Output : 1 3 2 4

Input : 11 12 13 14 15 16 17 18 19 20
Output : 11 16 12 17 13 18 14 19 15 20

Assignment 2
(2019-20)

Sub: Data Structures **Class: SE Comps**
Date of submission: 20 SEP 2019

Course outcome: CSC305.1: CO1: Implement various linear data structures
CSC305.1: CO2: Implement various Non linear data structures

Q. 1) Intersection of two Sorted Linked Lists
Given two lists sorted in increasing order, create a new list representing the intersection of the two lists. The new list should be made with its own memory — the original lists should not be changed.
For example, let the first linked list be 1->2->3->4->6 and second linked list be 2->4->6->8, then your function should create a third list as 2->4->6.

Q. 2) Pair wise swap elements of a given linked list by changing links
Given a singly linked list, write a function to swap elements pair wise. For example, if the linked list is 1->2->3->4->5->6->7 then the function should change it to 2->1->4->3->6->5->7, and if the linked list is 1->2->3->4->5->6 then the function should change it to 2->1->4->3->6->5

Q.3) Find pairs with given sum in doubly linked list
Given a sorted doubly linked list of positive distinct elements, the task is to find pairs in doubly linked list whose sum is equal to given value x, without using any extra space.

Example:

Input : head : 1 <-> 2 <-> 4 <-> 5 <-> 6 <-> 8 <-> 9
x = 7
Output: (6, 1), (5,2)

Q. 4) Interleave the first half of the queue with second half (without using linked list)
Given a queue of integers of even length, rearrange the elements by interleaving the first half of the queue with the second half of the queue. Only a stack can be used as an auxiliary space.
Examples:
Input : 1 2 3 4
Output : 1 3 2 4

Input : 11 12 13 14 15 16 17 18 19 20
Output : 11 16 12 17 13 18 14 19 15 20

Q. 5) Write a function to print In order predecessor and successor for a given key in BST.

Q.6) Write a c program to implement topological sorting.

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

Fr. Agnel Ashram,Bandstand,Bandra (west),Mumbai 400050

I Unit Test

Semester/Branch: (III Computer)
Subject: Data structures **Marks: 20**
Date: 13th August 2019 **Time: 11:30a.m-12:30 p.m.**

CSC305.1 (CO1): Implement various linear data structures

Q.1 a)	What does the following function do for a given Linked List with first node as *head*? **void fun1(struct node* head)** **{** **if(head == NULL)** **return;** **fun1(head->next); printf("%d ", head->data);** **}** (A) Prints all nodes of linked lists (B) Prints all nodes of linked list in reverse order (C) Prints alternate nodes of Linked List (D) Prints alternate nodes in reverse order	**(6M) CO1**

Q. 1 b)	Suppose you are given an implementation of a queue of integers. The operations that can be performed on the queue are:
	i. isEmpty (Q) — returns true if the queue is empty, false otherwise.
	ii. delete (Q) — deletes the element at the front of the queue and returns its value.
	iii. insert (Q, i) — inserts the integer i at the rear of the queue.
	
	Consider the following function:
	void f (queue Q) {
	int i ;
	if (!isEmpty(Q)) {
	i = delete(Q);
	f(Q);	

Q. 1 c)	Which data structure is used in redo-undo feature? (A) Stack (B) Queue (C) Tree (D) Graph	
Q. 1 d)	Which of the following permutation can be obtained in the same order using a stack assuming that input is the sequence 5, 6, 7, 8, 9 in that order? (A) 7,8,9,5,6 (B) 5,9,6,7,8 (C) 7,8,9,6,5 (D) 9,8,7,5,6	
Q. 2 a) Q. 2b)	Write a program to implement Queue data structure using stack **OR** WAP to evaluate a postfix expression	**(6M) CO1**
Q.3 a)	A Library wants to maintain records of available books. Each book has book_id, book_name and book_cost. Select appropriate data structure to maintain this information. Books should be stored in increasing order of book_id. Write functions for following operations a) Adding a new book as per book_id. b) Delete a book having specific book_id	**(6M) CO1**
	OR	
Q. 3 b)	Write functions to implement singly linked list a) Delete all occurrences of a specific node b) Sort elements of linked list in increasing order	
Q.4)	Differentiate between malloc() and calloc()	**(2M) CO1**

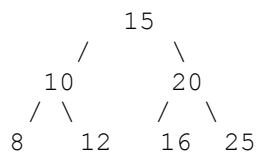
FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING
UNIT TEST 2

CLASS: SE COMPUTER **Time: 11:30a.m.-12:30p.m**
DATE: 14/10/2019 **Marks:20**

CO2: Implement Various Nonlinear Data Structures
CO3:Select and Apply appropriate sorting and searching algorithms

Q.1a)	The pre order traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the post order traversal sequence of the same tree? **(A)** 10, 20, 15, 23, 25, 35, 42, 39, 30 **(B)** 15, 10, 25, 23, 20, 42, 35, 39, 30 **(C)** 15, 20, 10, 23, 25, 42, 35, 39, 30	**1M CO2**

	(D) 15, 10, 23, 25, 20, 35, 42, 39, 30	

Q. 1b)	Consider the following code snippet in C. The function print() receives root of a Binary Search Tree (BST) and a positive integer k as arguments.
	`// A BST node`
	`struct node {`
	`  int data;`
	`  struct node *left, *right;`
	`};`
	
	`int count = 0;`
	
	`void print(struct node *root, int k)`
	`{`
	`  if (root != NULL && count <= k)`
	`  {`
	`  print(root->right, k);`
	`  count++;`
	`  if (count == k)`
	`  printf("%d ", root->data);`
	`  print(root->left, k);`
	`  }`
	`}`
	
	What is the output of print(root, 3) where root represent root of the following BST.
	
	 
	
	(A) 10
	
	(B) 16
	(C) 20
	(D) 20 10
		1M
		CO2

Q. 1 c)	Traversal of a graph is different from tree because **(A)** There can be a loop in graph so we must maintain a visited flag for every vertex **(B)** DFS of a graph uses stack, but in order traversal of a tree is recursive **(C)** BFS of a graph uses queue, but a time efficient BFS of a tree is recursive. **(D)** All of the above	**1M** **CO2**
Q.1 d)	Make is a utility that automatically builds executable programs and libraries from source code by reading files called makefiles which specify how to derive the target program. Which of the following standard graph algorithms is used by Make. **A)** Strongly Connected Components **(B)** Topological Sorting **(C)** Breadth First Search **(D)** Depth first Search	**1M** **CO2**

Q.2)	Write Functions 1) To create Expression tree 2) To Print equivalent Infix expression **OR**	6M **CO2**
	Write Functions 1) to delete a node from BST 2) To count Leaf Nodes Of BST	
Q. 3)	Write a program to create a graph using Adjacency matrix representation and perform BFS traversal to find the shortest path between nodes U and V.	**6M** **CO2**
Q.4)	Which sorting algorithm you will use if the input is nearly sorted? Write function of the selected sorted method	**4M** **CO3**