2016-2017

Data Structures
3CCI02

Dr. Purohit Shrinivasacharya
Assistant Professor
Dept of Information Science and Engineering
Siddaganga Institute of Technology, Tumakuru-572103

[COURSE PLAN]

Siddaganga Institute of Technology Tumakuru 03
Department of Information Science and Engineering

Course Plan for the Academic Year 2016-2017

Semester: **III**

Course Title: **Data Structures**	*Course Code*: **3CCI02**
Total Contact Hours: **52L+13T (65) Hrs.**	*Duration of SEE*: **3 Hrs**
SEE Marks: **100**	*CIE Marks*: **50**
Lesson Plan Author : **Dr. Purohit Shrinivasacharya**	*Date*: 29-07-2016

Prerequisites

Knowledge of fundamentals of programming, problem-solving, design, and implementation skills and basic mathematics is essential.

Course Overview

- The primary objective is to provide the student with an advanced treatment of computer programming with an emphasis on design and implementation of abstract data structures.
- Choose the data structures that effectively model the information in a problem.
- Judge efficiency trade-offs among alternative data structure implementations or combinations.
- To write Program effectively with pointers, arrays, structures, and dynamically allocated memory and describe their internal representations.
- To demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, and trees.
- Implementations of various data structures in more than one manner.
- Comparisons of different implementations of data structures and to recognize the advantages and disadvantages of the different implementations.
- This course also covers file management concepts like basic file I/O operations, file write and read operations.

Course Learning Objectives - CLO

1. Explain how different data structures stores data on physical memory
2. Explain how different operations like managing and retrieval of data.
3. Identify characteristics of linear versus nonlinear data structures with basic operations performed on it.
4. Compare different implementations of the same data structure.
5. Design the required data structure for the real world problems.

Course Outcomes - COs

At the end of the course the student will be able to

1. **Able** to understand the concepts of data structure, data type, arrays, files data structure.
2. **Implement** different data structures such as stacks, queues using C language and

Applications.

3. **Able** to explain and implement linear data structure like linked list.
4. **Apply** the knowledge of implementing the different data structures using list and its applications.
5. **Describe** and **Analyze** the nonlinear data structures like trees.

Mapping between Course Outcomes with Program Outcomes:

Course Outcomes	a	b	c	d	e	f	g	h	i	j	k
Able to understand the concepts of data structure, data type, arrays, files data structure.		M		M							M
Implement different data structures such as stacks, queues using C language and Applications.		M		M							M
Able to implement and explain linear data structure like linked list.		L		M							H
Apply the knowledge of implementing the different data structures using list and its applications.		L	L	M							H
Describe and **Analyze** the nonlinear data structures like trees.		L									M

6. Degree of compliance **L**: Low **M**: Medium **H**: High
7. **Refer last page for Program outcomes(a to k)**

SCHEDULE PLAN FOR ACADEMIC YEAR 2016-17

Sl. No.	Date	Topic	Remarks
Module – I			
C Programming Language Features:			
1	02-08-2016	Review of C programming concepts. • Branching and Decision Making • Looping	
2	02-08-2016	Review of C programming concepts. • User defined functions	
3	03-08-2016	Structures and Unions: Defining structures, Declaring structure variables, • Type-Defined Structures	

4	04-08-2016	Accessing structure members, structure initialization. • Rules for Initializing Structures • Different ways to access members	
5	09-08-2016	Copying and comparing structure variables, Operations on individual members	
6	09-08-2016	Array of structures, Unions, size of structures	
		Bit fields in structures, Structures and functions	
7	10-08-2016	**File management:** Defining & opening a file	
8	11-08-2016	Closing a file, Input/output operations on files. • getc and putc functions • fscanf and fprintf functions	
9	16-08-2016	Error handling during files operations,	
10	16-08-2016	Random access to files, Command line arguments and binary files.	Assignment1
Module – II			
Stack, Recursion, Queues and List			
11	17-08-2016	**The Stacks:** Definition and examples, Primitive operations, examples.	
12	18-08-2016	Representing stacks in C: Representing expressions in infix, postfix and prefix notations. • Implementing the pop operation • Implementing the push operation	
13	23-08-2016	• Conversion from infix to postfix examples	
	23-08-2016	• Conversion from infix to prefix and examples	
14	24-08-2016	Evaluating a postfix expression.	
Recursion			
15	25-08-2016	Use of stack while executing recursive routines.	
16	30-08-2016	Writing recursive programs: • Towers of Hanoi problem • Factorial of a number • Sum of N natural numbers	
	31-08-2016	Efficiency of Recursion	
The Queues			
17	01-09-2016	Definition of Queue and its sequential representation: Types of queues	
18	06-09-2016	C implementation of Queue (Ordinary queues) : insertion, deletion and display operations	
19	06-09-2016	Circular queues and insertion, deletion and display operations	

20	07-09-2016	Double ended queue, Priority queue and insertion, deletion and display operations.	Assingment2
Module – III			
Queues and Lists Continued			
21	08-09-2016	**Linked Lists** : Definition, Creating, Inserting and removing nodes from a list	I Test 10,11,12 Oct.
22	13-09-2016	Linked implementation of stacks, getnode and freenode operations,	
23	14-09-2016	Linked implementation of queues	
24	15-09-2016	Linked list as a data structure, examples of list operations	
25	20-09-2016	List implementation of priority queues, header nodes.	
26	20-09-2016	Lists in C: Array implementation of lists, limitations.	
27	21-09-2016	Allocating and freeing dynamic variables.	
28	22-09-2016	Linked lists using dynamic variables, Queues as lists in C.	
29	29-09-2016	Examples of list operations in C, non- integer and non-homogeneous lists.	
30	04-10-2016	Comparing dynamic and array implementations of lists, implementing header nodes	Assingment3
Module – IV			
Other List Structures			
31	04-10-2016	Other list structures: Circular lists	
32	05-10-2016	Stack as a circular lists.	
33	06-10-2016	Queue as a circular list.	
34	13-10-2016	Examples on circular list.	
35	18-10-2016	Primitive operations on circular lists.	
36	18-10-2016	The Josephus problem.	
37	19-10-2016	Header nodes.	
38	20-10-2016	Additions of long positive integers using circular lists.	
39	25-10-2016	Doubly linked lists.	
40	25-10-2016	Addition of long integers using Doubly linked lists.	Assingment4
Module – V			
Non- Linear Data Structures			

41	26-10-2016	**Trees**: Introduction, Types of trees, tree and a forest.	
42	27-10-2016	Array representation.	
43	02-11-2016	Linked representation.	
44	03-11-2016	Binary trees, binary tree traversals: Inorder, preorder, postorder.	
45	08-11-2016	Examples	
46	08-11-2016	Binary search tree.	
47	09-11-2016	Operations on binary search tree: Creation.	
48	10-11-2016	Searching a node, inserting a node.	
49	15-11-2016	Deleting a node from binary search trees.	
50	15-11-2016	Threaded binary trees.	
51	16-11-2016	Types of threaded binary trees and their representation.	
52	19-11-2016	Heaps, Expressions trees.	

TEXT BOOKS:

1	E. Balaguruswamy	**Programming in ANSI C,** 4th Edition, Tata McGraw-Hill Publications
2	Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum	Data Structures using C and C++, PHI/Pearson.

REFERENCE BOOK:

1	Jean-Paul Tremblay, Paul G. Sorenson	An Introduction To Data Structures With Applications, 2nd edition, McGraw-Hill International Editions

TUTORIAL CLASSES

Sl.No	Date	Lesson/Topic covered
1.	06-08-2016	C programs on • Structures and unions. • Demonstrate pointers concept
2.	13-08-2016	C programs to demonstrate • File management concepts like file I/O, • Error handling.
3.	20-08-2016	C programs on stack data structure and applications of stack. C programs and examples on • conversion from infix to postfix • conversion from infix to prefix

4.	27-08-2016	C programs to solve problems using recursive functions. • Binary search • Finding GCD of two numbers
5.	03-09-2016	C programs on queue data structure and different types of queue data structures. • Insertion and Deletion • Implementation of Priority queues
6.	10-09-2016	C programs to demonstrate linked list and to implement different types of linked list data structures dynamically. • Inserting the item in a specified position • Inserting the item in the ordered list • Reversing the list • Concatenation of two lists
7.	17-09-2016	C programs to implement array representations of Linked list. • Use of header nodes
8.	24-09-2016	C programs to implement array representations of Linked list. • Implementation of ascending priority queues • Implementation of descending priority queues
9.	01-10-2016	C programs to implement applications of circular linked list. Examples on the circular linked lists.
10.	08-10-2016	C programs to implement applications of linked list using dynamic variables.
11.	22-10-2016	C programs to demonstrate the concepts of binary trees.
12.	05-11-2016	C programs to implement the threaded binary trees.
13.	12-11-2016	C programs on • Implementation of expression trees. • Implementation of priority queue using binary search trees.

Evaluation Scheme - CIE Scheme

Assessment	Weightage of Marks
Quiz - 1	03
Test – I	17
Quiz - 2	03
Quiz - 3	03
Test - II	17
Quiz - 4	03
Assignments	4
Total	**50**

Course Unitization for I, II Test and Semester End Examination

Modules		Topics	Teaching Hours	No. of Questions in		No. of Questions in SEE
				First Test	Second Test	
1	1	Structures and Unions, Array of structures, Unions, size of structures, bit fields.	6	1 OR 2	--	1
	2	File management: Defining & opening a file.	4	1	--	
2	3	The Stacks	5	1	--	1
	4	Recursion and The Queues	6	1	--	
3	5	Linked List and List in C	12	-	1 OR 2	1
4	6	Other List structures - Doubly and Circular linked lists and Applications	9	--	1 OR 2	1
5	7	Non- Linear Data Structures - Trees, Binary search tree, Applications	10	--	--	1
				All questions are compulsory	All questions are compulsory	First question is compulsory and questions from **two** to **six** answers any 4 questions.

UNIT I
MODULE WISE PLAN

Course Code & Title: **3CCI02, Data Structures**	
Chapter Number & Title: **Structures and Unions, Array of structures, Unions, bit fields. File management: Defining & opening a file,**	*Planned Hours:* **10L+02T Hrs**

Learning Objectives

At the end of the chapter the student should be able to:

1. **Explain** a mechanism of packing data of different types of variables under a single name. (L2)
2. **Draw** the logical structure of the members of the structure.(L2)

3. **Differentiate** between an array and structures.(L2)
4. **Implementation** of the unions and bit fields.(L3)
5. **Differentiate** between the unions and structures**.**(L2)
6. **Define** a file, what are the different operations performed with file and their syntaxes (L2).
7. **Design a** new structure for a given application problem.(L3)
8. **Explain** the concepts of bit fields and its advantages. (L3).
9. **Write** programs various operations performed with files.(L3)
10. **Write** program by using command line arguments.(L3)

Model Questions

1. What is data structure? (L1)
2. Explain the different techniques to declare and initialization of structures. (L2)
3. Write a program to implement the structure which consists of student information and print the data on the console. (L2)
4. Distinguish between (L2)
    a. Arrays and structures
    b. Unions and structures
5. Write a program to read the employee information from a file and delete the 5^{th} employee information from the file. (L2)
6. Write a program to read the numbers from a file and write odd and even numbers into two separate files. (L1)

UNIT 2
MODULE WISE PLAN

Course Code & Title: **3CCI02, Data Structures**	
Chapter Number & Title: **The Stacks, Recursion, The Queues**	*Planned Hours:***11L+03T Hrs**

Learning Objectives

At the end of the chapter the student should be able to:

1. **Distinguish between** stacks and queues.(L2)
2. **Describe** the structure of a Stack and its applications.(L3)
3. **Distinguish between** recursive and non recursive algorithms. (L3)
4. **Explain** primitive operations that can performed on stack.(L2)
5. **Implementation** of stacks using structures. (L1)
6. **List** various types of queues.(L1)
7. **Distinguish between** ordinary queue and priority queue.(L3)

8. **Explain** primitive operations that can performed on queues.(L2)
9. **Write** a recursive functions for various problems. (L4)
10. **Distinguish between** dequeue and circular queue.(L3)

Model Questions

1. Differentiate between stack and queue.(L2)
2. Write C functions and explain the PUSH and POP operations.(L4)
3. Write a C program to implement queue operations on different data types using structures.(L4)
4. Explain the different types of queues and its applications.(L2)
5. What is recursion? Write a recursive function to compare nth term of a Fibonacci series. Give the trace along with stack contents for n=4.(L3)
6. Write C functions and explain the INSERT, DELETE and DISPLAY operations with the circular queue.(L2)

UNIT 3
MODULE WISE PLAN

Course Code & Title: **3CCI02, Data Structures**	
Chapter Number & Title: **Linked Storage Representation – Linked Lists**	*Planned Hours:* **12L+03T Hrs**

Learning Objectives

At the end of the chapter the student should be able to:

1. **Explain** the need of the Dynamic memory allocation.(L3)
2. **Explain** the different types of lists and their storage representation.(L1)
3. **Implementation** of the stacks and queues using dynamic memory allocation.(L2)
4. **Distinguish** between the array implementation and linked list representations.(L3)
5. **Write** the list examples using C to store both integer and non-integer form.(L4)
6. **Define** a list and the basic operations performed on it.(L1)
7. **Differentiate** between the static memory allocation and dynamic memory allocation.(L2)
8. **Differentiate** between the calloc()and malloc dynamic memory allocation funcitons. (L2)
9. **Write** programs to implement the singly linked list with the header node.(L4)
10. **Write** programs to implement primitive operations with the singly linked list.(L4)

Model Questions

1. Give the syntax of different types of dynamic memory allocation functions in C.(L3)
2. Explain how linked list representation of stacks is more efficient than array representation. (L1)
3. Write a C program to count the no. of nodes in a singly linked list.(L1).
4. Distinguish between the malloc() and calloc() functions.(L3)
5. Write a C program to delete a node form the front end and insert it at the end of the list.(L3)
6. Write a program to delete a node from a singly linked list whose position is specified. (L3).
7. Distinguish between the array implementation and linked list representations.(L3)
8. Write a program to delete all occurrences of a given node from a singly linked list.(L3).

UNIT 4
MODULE WISE PLAN

Course Code & Title: : **3CCI02, Data Structures**	
Chapter Number & Title: **Other List Structures – Doubly and Circular linked lists and Applications**	*Planned Hours:* **9L+02T Hrs**

Learning Objectives

At the end of the chapter the student should be able to:

1. **Define** doubly linked list.(L1)
2. **Write** programs to implement the primitive operations with doubly linked list.(L6)
3. **Differentiate** between circular linked lists and doubly linked list.(L2)
4. **List** different types of lists.(L1)
5. **Explain** the purpose of using the circular lists and doubly linked lists.(L2)
6. **Describe** Josephus problem.(L2)
7. **Differentiate** between circular singly linked list and circular doubly linked list (L2)
8. **Describe** the usage of header nodes with lists.(L2)
9. **Define** circular doubly linked list.(L1)
10. **Apply** the concept of doubly linked list to add two long integer numbers. (L4)

Model Questions

1. Define a circular linked list with header.(L1)

2. Explain the Josephus problem with example.(L2)
3. Write a program to add/subtract two polynomials.(L3)
4. Write functions to perform the following.
    a. To find the sum of all the elements in a list.
    b. To insert a new node in the specified position in the list.
5. How long positive numbers can be represented using circular singly linked lists? Explain with an example.
6. Write a program to implement the following functions with the circular doubly linked lists with the header node. (L3)
    a. To insert a new node in the beginning
    b. To insert a node X in the position Y.
    c. To display contents of the list in reverse order.

UNIT 5
MODULE WISE PLAN

Course Code & Title: : **3CCI02, Data Structures**	
Chapter Number & Title: **Non- Linear Data Structures - Trees, Binary Search Trees, Threaded binary trees**	*Planned Hours:* **10L+03T Hrs**

Learning Objectives

At the end of the chapter the student should be able to:

1. **Define** the following (L1)
    a. Binary tree.
    b. All most complete binary tree
2. **Explain** the concept of representation of both array and linked list representation of trees. (L2)
3. **Write** program for creation of binary search tree. (L3)
4. **Describe** the benefits of using the trees and tree traversal techniques.(L2)
5. **Design** the heap trees and expression trees.(L3)
6. **Explain** the in-threaded binary tree with examples.(L2)

Model Questions

1. What is a binary tree? (L1)

2. Define the following terms: (L1)
   a. Forest
   b. Depth of the tree
   c. Ancestors of a node
3. Write C functions to perform the following operations. (L3)
   a. Creation of binary tree
   b. Insertion of a new node
   c. Deleting a new node
4. Write a program to evaluate the expression using tree. (L3)
5. Explain different types of threaded binary trees with an example for each type. (L2)
6. Define a heap? Explain max heap and min heap with examples. (L3)
7. Select a data structre to sort the file content (numbers) and implement using C program. (L4)

Faculty

29/07/16
Head of the Department

29.07.16
Principal

Programme Outcomes (POs):

To achieve the above objectives, Information Science and Engineering degree programme strives to obtain the following outcomes which should be achieved by all graduates at the time of their graduation. Each graduate should posses:

a. Ability to apply sound fundamental knowledge of mathematics, basic sciences, electrical, electronics, mechanical, civil engineering and environmental engineering concepts to both hardware and software design problems.

b. A comprehensive brainstorming knowledge in fundamental concepts of computer and Information Science Engineering

c. An understanding of principles of analysis, design, programming, architecture and issues related to information management.

d. In depth learning and practicing programming languages that are used for development and design of software.

e. Learning system software concepts like loaders, linkers, compilers, process activities, process communication etc.

f. Familiarity with computer network related issues like reference models, protocols, and communication and security issues in network.

g. To acquire necessary management and leadership qualities and skills to manage resources like people, information, finance etc.

h. An understanding of current and advanced concepts and an ability to demonstrate the skills to use modern engineering tools etc.

i. To understand the Artificial Intelligence (AI) systems those informs about the utilization and organization of an object, multi dimensional data etc.

j. Fostering ability to demonstrate knowledge of analytical ability, soft skills and use of social awareness in the profession ethically.

k. Exposure towards research, industrial working environment and skills to develop software system.