DATA STRUCTURES LAB

II B. TECH- I SEMESTER

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIE	SEE	Total
A4CS06	PCC	-	-	3	1.5	30	70	100

COURSE OBJECTIVES:

The course should enable the students to:

1. Ability to identify the appropriate data structure for given problem.
2. Design and analyze the time and space complexity of algorithm or program.
3. Effectively use compilers include library functions, debuggers and trouble shooting.
4. Write and execute programs using data structures such as arrays, linked lists to implement stacks, queues.
5. Write and execute programs in C to implement various sorting and searching.
6. Write and execute programs using data structures such as arrays, linked lists to implement trees, graphs, hash tables and search trees.

COURSE OUTCOMES:

The course should enable the students to:

1. Use appropriate data structure for given problem.
2. To analyze the time and space complexity of algorithm or program.
3. Use compilers include library functions, debuggers and trouble shooting.
4. Execute programs using data structures such as arrays, linked lists to implement stacks and queues.
5. Execute write programs in C to implement various sorting and searching.
6. Execute programs using data structures such as arrays, linked lists to implement trees, graphs, hash tables and search trees.

LIST OF EXPERIMENTS

WEEK-1 SINGLE LINKED LIST

Write a C program that uses functions to perform the following:
a) Create a singly linked list of integers.
b) Delete a given integer from the above linked list.
c) Display the contents of the above list after deletion.

WEEK-2 DOUBLE LINKED LIST

Write a C program that uses functions to perform the following:
a) Create a doubly linked list of integers.
b) Delete a given integer from the above doubly linked list.
c) Display the contents of the above list after deletion.

WEEK-3 INFIX TOPOSTFIX CONERSION

Write a C program that uses stack operations to convert a given infix expression into its postfix Equivalent, Implement the stack using an array.

WEEK-4 DOUBLE ENDED QUEUE

Write C programs to implement a double ended queue ADT using
i)array and
ii)doubly linked list

WEEK-5 **BINARY SEARCH TREES USING RESURSION**

Write a C program that uses functions to perform the following:
a) Create a binary search tree of characters.
b) Traverse the above Binary search tree recursively in Postorder.

WEEK-6 **BINARY SEARCH TREES USING NON-RESURSION**

Write a C program that uses functions to perform the following:
a) Create a binary search tree of integers.
b) Traverse the above Binary search tree non recursively in inorder.

WEEK-7 **SORTING**

Write C programs for implementing the following sorting methods to arrange a list of integers in Ascending order :
a) Insertion sort
b) Merge sort

WEEK-8 **SORTING**

Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order:
a) Quick sort
b) Selection sort

WEEK-9 **B-TREES**

1. Write a C program to perform the following operation:

a)Insertion into a B-tree.

1. Write a C program for implementing Heap sort algorithm for sorting a given list of integers in ascending order.

WEEK-10 **HASHING**

Write a C program to implement all the functions of a dictionary (ADT) using hashing.

Week-11 **PATTERN MATCHING ALGORITHM**

Write a C program for implementing Knuth-Morris- Pratt pattern matching algorithm.

Week-12 **GRAPH TRAVERSAL ALGORITHMS**

Write C programs for implementing the following graph traversal algorithms:
a)Depth first traversal b)Breadth first traversal

TEXT BOOKS:

1. C and Data Structures, Prof. P.S.Deshpande and Prof. O.G. Kakde, Dreamtech Press.
2. Data structures using C, A.K.Sharma, 2nd edition, Pearson.
3. Data Structures using C, R.Thareja, Oxford University Press.

WEB REFERENCES:

1. http://www.sanfoundry.com/data structures-examples
2. http://www.geeksforgeeks.org/c
3. http://www.cs.princeton.edu