AMITY UNIVERSITY

JHARKHAND

LAB MANUAL

Course Title: Data Structure

Course Level: UG

Course Code: CSIT124

Program: BCA/B.Sc. (IT)

Semester: II

Faculty:

MR. MOHAN DEHURY

Assistant Professor (CS)

Lab Instructor:

MR. RAHUL KUMAR LOHRA

Sr. Technical Assistant (CS), AUJ

General instructions to students

1. Students should be regular and come prepared for the lab practice.
2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).
3. Students should bring the observation book, lab journal and lab manual.
4. Prescribed textbook and class notes can be kept ready for reference if required.
5. They should implement the given experiment individually.
6. Once the experiment(s) get executed, they should show the program and results to the instructors and copy the same in their observation book.
7. Questions for lab tests and exam need not necessarily be limited to the questions in the manual but could involve some variations and / or combinations of the questions.
8. All the students must maintain silence inside the lab.
9. All the students must carry their id card before entering the lab, and college uniform is strictly mandatory, otherwise students will not be permitted to sit inside the lab.
10. No food or beverage items are allowed inside the lab.
11. Keep your bags outside the lab.
12. Do not use cell phone inside the lab. (If anybody found using cell phone inside the lab, his/her mobile phone will be seized by the responsible authority.)
13. Shut down the system and arrange your chair before leaving the lab.
14. While using lab sign in the register with your name, enrollment number and branch to mark your attendance.
15. Do not plug any device without permission.
16. Do not use internet without permission.

Note: Above mentioned instructions can be modified based on the context of the lab.

Credit Unit

L (Lecture)	T	P/S (Practical/Studio)	SW/FW	Total Credit Units
3	-	2	-	4

Lab/ Practical/ Studio Assessment

	Continuous Assessment/Internal Assessment					End Term Examination	
Components (Drop down)	**Continuous Assessment**	**Mid Term**	**Viva**	**Lab Record**	**Attendance**	**Practical**	**Viva**
Weightage (%)	10	10	5	10	5	40	20

Course Objectives:

The aim of this course is to

- Impart in-depth knowledge of data structure and its implementation in computer programs.
- Make students understand the concepts of linear and nonlinear data structure.
- Illustrate asymptotic notations and their usage.

Pre-requisites: Basics of C Programming.

Student Learning Outcomes:
After successful completion of this course, the student will be able to

- Apply advance C programming techniques such as pointers, dynamic memory allocation, structures to developing solutions for problems.
- Describe and implement abstract data types such as linked list, stack, queue and tree by using 'C 'for static and dynamic implementations.
- Analyze, and evaluate appropriate abstract data types and algorithms to solve problems.

Pedagogy for Course Delivery:

Subject will be taught based on class room lectures and practical. Emphasis will be given on practical explaining use case scenario for various algorithms. Focus will be on student's involvement while imparting the course contents.

LIST OF EXPERIMENTS

Week 1: Overview of C

1. Implement the string operations of strcat, strcmp, strcpy functions using pointer.
2. Create a structure called employee with members Name, Age, Salary. Sort the array of such employee records:
    a. In alphabetical order of their names.
    b. In order of their seniority (age).
3. Implement length (in feet and inches) using structure. Write routines to add, multiply and subtract the lengths.

Week 2: Overview of C & Recursion

1. Create a sparse matrix.
2. Find the GCD of two numbers.
3. Simulate the working of Tower of Henoi problem for n disks. Print the number of moves.
4. Convert a given decimal number to binary using recursion.

Week 3: Searching & Sorting

1. Write a program to sort the given array using Bubble Sort.
2. Write a program to sort the given array using Selection Sort.
3. Write a program to sort the given array using Insertion Sort.
4. Write a program to search an element using Linear Search.

Week 4: Searching & Sorting

1. Write a program to search an element using Binary Search.
2. Write a program to sort the given array using Quicksort.
3. Write a program to sort the given array using Merge Sort.

Week 5: Arrays

1. Write a program to insert a new element in the given unsorted array at k^{th} position.
2. Write a program to delete an element from given sorted array.
3. Write a program to merge to given sorted arrays.

Week 6: Stack & Queue

1. Write a program to implement Stack using array, also show overflow and underflow in respective push and pop operations.
2. Write a program to implement Queue using array, which shows insertion and deletion operations.
3. Write a program to implement Circular Queue using array, which shows insertion and deletion operations.

Week 7: Linked List

1. Write a program to implement Linear Linked List, showing all the operations, like creation, display, insertion, deletion and searching.
2. Write a program to implement Stack, using Linked List. Implement Push, Pop and display operations.
3. Write a program to implement Queue, using Linked List. Implement Insertion, deletion and display operations.

Week 8: Linked List

1. Write a program to count the number of times an item is present in a linked list.
2. Write a program to increment the data part of every node present in a linked list by 10. Display the data both before incrimination and after.
3. Write a program to implement Doubly Linked List, showing all the operations, like creation, display, insertion, deletion and searching.

Week 9: Trees

1. Write a program to create a Binary Search Tree and display its contents using recursive preorder, postorder and inorder traversal.
2. Write a program to implement deletion of a node in binary search tree.
3. Write a program to implement Binary tree and display the contents using non-recursive preorder, postorder and inorder traversal techniques.

Week 9 & 10: Graph

1. Write a program to sort the given array using Heapsort.
2. Write a program of Graph Traversal-Depth first search and Breadth first search.
3. Write a program to implement Prim's algorithm.
4. Write a program to implement Kruskal algorithm.

****

Text Books:

- Yashwant Kanetkar," Data Structure using C", BPB Publication, 5th Edition ,2011
- A.Tannenbaum,Y. Lanhgsam and A.j. Augenstein ," Data Structures Using C And C++ ",Prentice Hall of India,2nd Edition,2009.
- Jean-Paul Tremblay, P.G Sorenson, "An Introduction to Data Structures with applications", Mcgraw-Hill ,2nd Edition ,1984.

References:

- Robert L Kruse, "Data Structure and Program Design in C", Prentice Hall (1991).
- Noel Kalicharan ,"Data Structure in C" ,Ist Edition Create space publisher, 2008.
- Mark Allen Weiss,"Data Structure and algorithm Analysis in C",2nd Edition AddisonWesley,1996.
- E. Balagurusamy, "Problem Solving through C language", TMH publication, Fourth Edition, 2008.
- R.S Salaria ,"Data Structures & Algorithms using C",Khanna Publication,4th Edition,2009
- E.Horowitz and S.Sahni,"Fundamentals of Data Structures in C ",2nd Edition,Universities Press,2008.