SWE2001	Data Structures and Algorithms	L	T	P	J	C
		3	0	2	0	4
Pre-requisite	CSE1001	Syllabus version				
		v.1.0				

Course Objectives:

1. To understand the basic concepts of data structures and algorithms in various fields.
2. To learn sorting of and search data items.
3. To comprehend the necessity of time complexity in designing algorithms.
4. To design algorithms to solve real life problems

Expected Course Outcome:

1. Analyze and understanding stack operations and its applications in real world problems.
2. Understand the pros and cons of various queues and its operations
3. Demonstrate linear data structures using dynamic arrays
4. Evaluate algorithms and data structures in terms of time and memory complexity of basic operations.
5. Understand, analyze and design sorting and searching algorithms
6. Understand the importance of hashing
7. Design non-linear data structure operations in real world problems
8. Apply suitable data structures and algorithms for autonomous realization of simple programs or program parts

Student Learning Outcomes (SLO)	1,2,14,17

Module:1	Stack	6 hours

Operations on stack, array implementation of stack, applications of stack-balance of parenthesis in algebraic expressions, converting expressions from infix to postfix or prefix form , evaluating postfix or prefix form, Towers of Hanoi problem

Module:2	Queue	6 hours

Operations on queue , circular queue, array implementation of queue, applications of queue

Module:3	List	6 hours

Singly linked list, doubly linked list, circularly singly linked list, operations on linked lists, Linked representation of stack, Linked representation of Queue

Module:4	Algorithm Analysis	6 hours

Asymptotic notations, Abstract data type, growth rate of functions, running time complexity, best, average and worst case analysis – examples

Module:5	Sorting and Searching	6 hours

Bubble sort, insertion sort, selection sort, radix sort, merge sort, quick sort, heap sort, Shell sort, linear search, binary search, time complexity analysis of sorting and searching algorithms.

Module:6	Hashing	6 hours

Hash functions, open hashing-separate chaining, closed hashing - linear probing, quadratic probing, double hashing, random probing, rehashing, extendible hashing

Module:7	**Tree and Graph**	**7 hours**
Implementation of tree, binary tree traversals, expression tree, binary search tree, AVL tree Graphs, Graph traversals, and shortest path algorithms-Dijkstra's algorithm		
Module:8	**Contemporary issues:** Applications of Data structure in Industry-Case Studies	**2 hours**
	Total Lecture hours:	**45 hours**

Text Book(s)	
1.	Mark Allen Weiss, "Data structures and algorithm analysis in C", 2nd edition, Pearson education, 2013.
Reference Books	
1. 2. 3. 4.	Debasis Samanta, "Classic data structures", PHI, 2nd edition, 2014. Seymour Lipschutz "Data Structures by Schaum Series" 2nd edition, TMH, 2013. Adam Drozdek, "Data structures and algorithms in C++", Cengage learning, 4th edition, 2015. Michael Goodrich, Roberto Tamassta, Michael H.GoldWasser "Data structures and algorithms in Java" 6th edition, 2014.

List of Challenging Experiments (Indicative)		**SLO: 1,2,14**
1.	Implement stack and use it to convert infix to postfix expression	
2.	Evaluate postfix expression	
3.	Implement Towers of Hanoi problem	
4.	Implement Queue and Circular Queue	
5.	Implement singly and doubly linked lists	
6.	Implement Circular Singly Linked list	
7.	Represent a polynomial as a linked list and write functions for polynomial addition.	
8.	Implement Insertion, Bubble, and selection sorts	
9.	Implement heap, merge, quick and radix sorts	
10.	Implement Binary and Linear search	
11.	Implement a Binary tree. Produce its pre-order, in-order, and post-order traversals.	
12.	Implement binary search tree insertion and deletion.	
13	Implement hashing techniques	
14	Perform Graph traversal	
15	Implement Dijkstra's algorithm	
	STACK ADT 1. Students of a Programming class arrive to submit assignments. Their register numbers are stored in a LIFO list in the order in which the assignments are submitted. Write a program using array to display the register number of	

	the ten students who submitted first.
	
	Register number of the ten students who submitted first will be at the bottom of the LIFO list. Hence pop out the required number of elements from the top so as to retrieve and display the first 10 students.
	
	2. To facilitate a thorough net surfing, any web browser has back and forward buttons that allow the user to move backward and forward through a series of web pages. To allow the user to move both forward and backward two stacks are employed. When the user presses the back button, the link to the current web page is stored on a separate stack for the forward button. As the user moves backward through a series of previous pages, the link to each page is moved in turn from the back to the forward stack.
	
	When the user presses the forward button, the action is the reverse of the back button. Now the item from the forward stack is popped, and becomes the current web page. The previous web page is pushed on the back stack. Simulate the functioning of these buttons using array implementation of
	
	Stack. Also provide options for displaying the contents of both the stacks whenever required.
	
	3. Design a program to employ a stack for balancing symbols such as parentheses, flower braces and square brackets, in the code snippet given below.
	
	for(i=0;i<n;i++)
	
	{
	
	if(i<5)
	
	{ z[i]=x[i]+y[i];
	
	p=(((a+b)*c)+(d/(e+f)*g);
	
	}
	
	Ensure that your program works for any arbitrary expression.
	
	4. Most of the bugs in scientific and engineering applications are due to improper usage of precedence order in arithmetic expressions. Thus it is necessary to use an appropriate notation that would evaluate the expression without taking into account the precedence order and parenthesis.	

a) Write a program to convert the given arithmetic expression into
i) Reverse Polish notation
ii) Polish notation
b) Evaluate the above notations with necessary input.

5. Some priests are given three poles and a stack of 4 gold disks, each disk a little smaller than the one beneath it. Their assignment is to transfer all 4 disks from one of the 3 pole to another with 2 important constraints. They can move only one disk at a time, and they can never place a larger disk on top of a smaller one. Design a recursive program for the above Towers of Hanoi puzzle using stack.

QUEUE ADT:

6. In a theme park, the Roller-Coaster ride is started only when a good number of riders line up in the counter (say 20 members). When the ride proceeds with these 20 members, a new set of riders will line up in the counter. This keeps continuing. Implement the above scenario of lining up and processing using arrays with Queue ADT.

7. When burning a DVD it is essential that the laser beam burning pits onto the surface is constantly fed with data, otherwise the DVD fails. Most leading DVD burn applications make use of a circular buffer to stream data from the hard disk onto the DVD. The first part, the 'writing process' fills up a circular buffer with data, then the 'burning process' begins to read from the buffer as the laser beam burns pits onto the surface of the DVD. If the buffer starts to become empty, the application should continue filling up the emptied space in the buffer with new data from the disk. Implement this scenario using Circular Queue.

8. a) There is a garage where the access road can accommodate any number of trucks at one time. The garage is built in such a way that only the last truck entered can be moved out. Each of the trucks is identified by a positive integer (a truck_id). Implement dynamically to handle truck moves, allowing for the following commands:

i) On_road (truck_id); ii) Enter_garage (truck_ id);

iii) Exit_garage (truck_id); iv) Show_trucks (garage or road);

If an attempt is made to get a truck out which is not the closest to the garage entry, the error message "Truck x cannot be moved" should be displayed.

b) For the aforementioned scenario, assume now a circular road and two entries: one for entry, another for exit. Trucks can get out only in the order they

got in. Write a program dynamically to handle truck moves allowing for the following commands

i) Enter garage (truck name)

ii) Exit garage (truck name)
iii) Show trucks

LIST ADT

9. Imagine an effective dynamic structure for storing polynomials. Write operations for addition, subtraction, and multiplication of polynomials.

I/O description. Input:

$p1=3x^7+5x^6+22.5x^5+0.35x^2$

$p2=0.25x^3+0.33x^2-0.01$

10. Given two sorted lists L1 and L2 write a program to merge the two lists in sorted order after eliminating duplicates.

11. Write a program to maintain the records of students in an effective dynamic structure. Search a particular record based on the roll number and display the previous and next values of that node with time complexity of O(1).

12. **Assume FLAMES** game that tests for relationship has to be implemented using a dynamic structure. The letters in the FLAMES stand for Friends, Love, Affection, Marriage, Enmity and Sister. Initially store the individual letters of the word 'flames' in the nodes of the dynamic structure. Given the count of the number of uncommon letters in the two names 'n', write a program to delete every nth node in it, till it is left with a single node. If the end of the dynamic structure is reached while counting, resume the counting from the beginning. Display the letter that still remains and the corresponding relationship

Eg., If Ajay and Jack are the two names, there are 4 uncommon letters in these. So delete 4^{th} node in the first iteration and for the next iteration start counting from the node following the deleted node.

SORTING AND SEARCHING

13. Assume in the Regional Passport Office, a multitude of applicants arrive each day for passport renewal. A list is maintained in the database to store the renewed passports arranged in the increased order of passport ID. The list already would contain there cords renewed till the previous day. Apply

	Insertion sort technique to place the current day's records in the list. Later the office personnel wish to sort the records based on the date of renewal so as to know the count of renewals done each day. Taking into consideration the fact that each record has several fields (around 25 fields), follow Selection sort logic to implement the same. 14. Implement a comparison based sorting algorithm which is not in-place to sort the following strings. best, true, hill, dove, van, good, egg, lap 15. Write a program to implement Bubble sort, Heap sort and Quick sort techniques to arrange the following sequence of elements in descending order. 9, -4, 5, 8,-3, 7, 0, 4, 1, 2. Display the count of number of comparisons and swaps made in each method. Apply the same sorting techniques for sorting a large data set [Randomly generate 5000 integers within the range -50000 to 50000 to build the data set]. From your observation and analysis, determine the best sorting technique for working with large numbers.	

		Total Laboratory Hours	30 hours
Recommended by Board of Studies	12.06.2015		
Approved by Academic Council	No. 37	Date	16.06.2015