# **COURSE PLAN**

Department : Data Science and Computer Applications

Course Name & code : Data Structures and Algorithms Lab & MCA 4261

Semester & branch : II Semester & M.C.A.

Name of the faculty : Dr. Raghurama Holla & Dr. Natesha B V

No of contact hours/week: L T P

### **Course Outcomes (COs)**

		No. of	
	At the end of this course, the student should be able to:	Contact	Marks
		Hours	
CO1:	Demonstrate the working of basic searching, sorting algorithms, and recursion	02	Marks
CO2:	Demonstrate the memory representation of data structures like sparse matrices and polynomials	02	Marks
CO3:	Demonstrate data structures like stack, queue, circular queue, linked lists, trees and graphs	05	Marks
CO4:	Apply the data structure stack to solve some probelms	02	Marks
CO5:	Demonstrate the working of advanced sorting methods	01	Marks
	Total	12	100

(Page 1 of 4) MIT/GEN/F-01/R2

## **Assessment Plan**

1. Continuous Evaluation	60%
--------------------------	-----

2 evaluations of 10 marks each : 20 marks 2 quizzes of 10 marks each: 20 marks

1 Execution check(Lab exercises): 10 marks

1 Viva: 5 marks

Observation Book: 5 marks

### 2. Lab Examination 40%

• 2 questions of 20 marks each

# **Lesson Plan**

L. No. Topics  Mapping of 2-D arrays to 1-D arrays:  1. Obtain the Row-major and Column-major representation of the given input matrix.  2. Map the following 2-D arrays (matrices) to 1-D arrays (lists).  a) Upper triangular matrix b) Lower triangular matrix c) Diagonal matrix d) Tri-diagonal matrix e) Row-major f) Column-major Display the element at any specified position (row, column).  L2 1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression  CO3,C 2) Evaluation of Prefix expression			1 6			
1. Obtain the Row-major and Column-major representation of the given input matrix.  2. Map the following 2-D arrays (matrices) to 1-D arrays (lists).  a) Upper triangular matrix b) Lower triangular matrix c) Diagonal matrix d) Tri-diagonal matrix e) Row-major f) Column-major Display the element at any specified position (row, column).  1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  1. Implementation of Stack using arrays 2. Conversion of Infix expression to Postfix expression (using stack) 3. Conversion of Infix expression to Prefix expression (using stack) 2. Evaluation of Postfix expression 2. Evaluation of Prefix expression 3. Implementation of Queue using arrays 2. Conversion of Infix expression 3. Implementation of Queue using arrays 3. Conversion of Infix expression 4. In Implementation of Queue using arrays 4. In Implementation of Queue using arr	L. No.	Topics	Course Outcome Addressed			
1. Obtain the Row-major and Column-major representation of the given input matrix.  2. Map the following 2-D arrays (matrices) to 1-D arrays (lists).  a) Upper triangular matrix b) Lower triangular matrix c) Diagonal matrix d) Tri-diagonal matrix e) Row-major f) Column-major Display the element at any specified position (row, column).  1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  1. Implementation of Stack using arrays 2. Conversion of Infix expression to Postfix expression (using stack) 3. Conversion of Infix expression to Prefix expression (using stack) 4. I) Evaluation of Postfix expression 2. Evaluation of Prefix expression 3. Implementation of Queue using arrays 4. I) Implementation of Queue using arrays 4. I) Implementation of Queue using arrays 5. II Implementation of Queue using arrays 6. II Implementation of Queue using arrays 7. In Implementation of Queue using arrays 8. In Implementation of Queue using arrays 9. In Implement		Mapping of 2-D arrays to 1-D arrays:				
matrix.  2. Map the following 2-D arrays (matrices) to 1-D arrays (lists).  a) Upper triangular matrix b) Lower triangular matrix c) Diagonal matrix d) Tri-diagonal matrix e) Row-major f) Column-major Display the element at any specified position (row, column).  L2 1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  L3 Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression 3) Implementation of Queue using arrays 2) Evaluation of Oueue using arrays 3) Evaluation of Queue using arrays 4 CO3,C C						
a) Upper triangular matrix b) Lower triangular matrix c) Diagonal matrix d) Tri-diagonal matrix e) Row-major f) Column-major Display the element at any specified position (row, column).  1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  14		matrix.				
a) Upper triangular matrix b) Lower triangular matrix c) Diagonal matrix d) Tri-diagonal matrix e) Row-major f) Column-major Display the element at any specified position (row, column).  1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  14		2. Map the following 2-D arrays (matrices) to 1-D arrays (lists).				
c) Diagonal matrix c) Diagonal matrix d) Tri-diagonal matrix e) Row-major f) Column-major Display the element at any specified position (row, column).  1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  14 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  1. Evaluation of Postfix expression 2) Evaluation of Postfix expression 10 Implementation of Queue using arrays 20 Cost		, 11				
d) Tri-diagonal matrix e) Row-major f) Column-major Display the element at any specified position (row, column).  1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  2. Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  2. CO3,C 2. Conversion of Infix expression to Postfix expression (using stack) 3. Conversion of Infix expression to Prefix expression (using stack)  1. Evaluation of Postfix expression 2. Evaluation of Prefix expression 3. Implementation of Queue using arrays 4. CO3,C 4. Di Implementation of Queue using arrays 4. CO3,C 5. CO3,C 6. CO3,C 6. CO3,C 6. Di Implementation of Queue using arrays 6. CO3,C 7. CO3,C 7. CO3,C 8. CO3,C 8. CO3,C 8. CO3,C 8. CO3,C 9. Evaluation of Poetfix expression 9. Evaluation of Queue using arrays 9. CO3,C	L1					
e) Row-major f) Column-major Display the element at any specified position (row, column).  1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  2. Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  2. CO3,C 2. Conversion of Infix expression to Postfix expression (using stack) 3. Conversion of Infix expression to Prefix expression (using stack)  1. Evaluation of Postfix expression 2. Evaluation of Prefix expression 3. Implementation of Queue using arrays 4. CO3,C 4. Display the element at any specified position (row, column).  CO2  CO3  CO3  CO3  CO3  CO3  CO3  CO3		c) Diagonal matrix				
f) Column-major Display the element at any specified position (row, column).  L2 1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix. 2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  L3 Solving problems using Recursion: a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression CO3,C  L6 1) Implementation of Queue using arrays CO3,C  L6 1) Implementation of Queue using arrays		d) Tri-diagonal matrix				
Display the element at any specified position (row, column).  1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix.  2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  2. Solving problems using Recursion:  a) Tower of Hanoi for n disks(Recursion application)  b) Factorial of a given number  c) GCD of 2 numbers  d) Fibonacci series upto nth term  2. CO3,CO3,CO3,CO3,CO3,CO3,CO3,CO3,CO3,CO3,		e) Row-major				
1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix.  2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  CO1  a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  CO3  1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5  1) Evaluation of Postfix expression 2) Evaluation of Prefix expression  CO3,C  L6  1) Implementation of Queue using arrays  CO3,C		f) Column-major				
original matrix.  2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  L3 Solving problems using Recursion:  a) Tower of Hanoi for n disks(Recursion application)  b) Factorial of a given number  c) GCD of 2 numbers  d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays  2) Conversion of Infix expression to Postfix expression (using stack)  3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression  2) Evaluation of Prefix expression  CO3,C  L6 1) Implementation of Queue using arrays  CO3,C		Display the element at any specified position (row, column).				
2. Represent a polynomial using 1-D array and perform addition operation on two polynomials.  L3 Solving problems using Recursion:  a) Tower of Hanoi for n disks(Recursion application)  b) Factorial of a given number  c) GCD of 2 numbers  d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression  C03,C	L2	1. Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the	CO2			
polynomials.  L3 Solving problems using Recursion:  a) Tower of Hanoi for n disks(Recursion application)  b) Factorial of a given number  c) GCD of 2 numbers  d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays  2) Conversion of Infix expression to Postfix expression (using stack)  3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression  C03,C  L6 1) Implementation of Queue using arrays  C03		original matrix.				
Solving problems using Recursion:  a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression CO3,C 2) Evaluation of Queue using arrays CO3,C 3) Implementation of Queue using arrays		2. Represent a polynomial using 1-D array and perform addition operation on two				
a) Tower of Hanoi for n disks(Recursion application) b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression CO3,C 2) Evaluation of Queue using arrays  CO3		<u> </u>				
b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression  CO3,C 2) Evaluation of Queue using arrays  CO3	L3		CO1			
c) GCD of 2 numbers d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression CO3,C 2) Evaluation of Prefix expression CO3,C 3) Co3,C 4 1) Implementation of Queue using arrays		a) Tower of Hanoi for n disks(Recursion application)				
d) Fibonacci series upto nth term  L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression  L6 1) Implementation of Queue using arrays  CO3,C						
L4 1) Implementation of Stack using arrays 2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression  L6 1) Implementation of Queue using arrays  CO3,C						
2) Conversion of Infix expression to Postfix expression (using stack) 3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression  L6 1) Implementation of Queue using arrays  CO3,C		d) Fibonacci series upto nth term				
3) Conversion of Infix expression to Prefix expression (using stack)  L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression  L6 1) Implementation of Queue using arrays  CO3,C	L4		CO3,CO4			
L5 1) Evaluation of Postfix expression 2) Evaluation of Prefix expression  L6 1) Implementation of Queue using arrays  CO3,C						
2) Evaluation of Prefix expression  L6 1) Implementation of Queue using arrays  CO3		3) Conversion of Infix expression to Prefix expression (using stack)				
L6 1) Implementation of Queue using arrays CO3	L5	1) Evaluation of Postfix expression	CO3,CO4			
1) implementation of Queue using urrays		2) Evaluation of Prefix expression				
2) Implementation of Circular Queue using arrays	L6		CO3			
		2) Implementation of Circular Queue using arrays				

(Page 2 of 4) MIT/GEN/F-01/R2

L7	<ol> <li>Implement a sorted singly linked list. Include the following options: inserting a node, deleting a node and displaying the list.</li> <li>Reverse a singly-linked list using recursion.</li> <li>Implementing stack using Singly linked list.</li> <li>Implementing queue using Singly linked list</li> </ol>	CO3
L8	<ol> <li>Implement a sorted doubly linked list. Include the following options: inserting a node, deleting a node and displaying the list in both directions.</li> <li>Create a binary search tree and traverse it in preorder, inorder and postorder traversal methods</li> </ol>	CO3
L9	1.Represent a directed graph in the following ways:  a) Adjacency matrix b) Adjacency list 2.) Represent an undirected graph in the following ways: a) Adjacency matrix b) Adjacency list	CO3
L10	Implement the following sorting techniques.  (a) Quick sort  (b) Bubble sort  (c) Selection sort  (d) Insertion sort	CO3, CO5
L11	<ol> <li>Merge Sort</li> <li>Implement the following searching techniques.</li> <li>Sequential search.</li> <li>Binary search (Iterative method).</li> <li>Binary search (Recursive method).</li> </ol>	CO3, CO5
L12	End-Semester Laboratory Examination	-
L13	Click or tap here to enter text.	СО
L14	Click or tap here to enter text.	СО

#### References:

- 1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, "Data Structures and Algorithms", 4th Edition, Addison Wesley, 2009.
- 2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein "Introduction to Algorithms", 3rd Edition, PHI Publications, 2009.
- 3. Sartaj Sahani, "Data Structures, Algorithms and Applications in C++", 2nd Edition, Universities Press, 2005. J. P. Trembley and Sorenson, "An Introduction to Data Structures with Applications" 2nd Edition, 36th Reprint, McGraw Hill, 2008.
- 4. J. P. Trembley and Sorenson, "An Introduction to Data Structures with Applications" 2nd Edition, 36th Reprint, McGraw Hill, 2008.

(Page 3 of 4) MIT/GEN/F-01/R2

	Click or to	Click or tap here to enter text.					
-	Click or to	ap here to	enter text.				
_	Click or to	Click or tap here to enter text.					
	Submitted by:		Dr. Raghurama	Holla			
	(Signature of the faculty)						
<b>Date:</b> 30-01-2023							
	Approved by: Click or tap here to			re to enter text.			
	(Signature of HOD)						
	Date:	30-01-20	)23				
	FACULTY MEMBERS TEACHING THE COURSE (IF MULTIPLE SECTIONS EXIST):						
		FA	CULTY	SECTION	FACULTY	SECTION	
	N 4: a a 1	inda Vara	h	Λ			

FACULTY	SECTION	FACULTY	SECTION
Miss Linda Varghese	Α		
Dr. Raghurama Holla	В		
	*****		

(Page 4 of 4)