CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT201	DATA STRUCTURES	PCC	3	1	0	4

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates capable of understanding essential concept of data structures, designing algorithms to perform operations involving these data structures and to choose appropriate data structures to solve real world problems.

Prerequisite: programming in C

Course Outcomes: After the completion of the course the student will be able to

CO_No.	Course Outcome(CO)	Bloom's Category
CO 1	**Summarize** different categories of data structures	Level 2 : Understand
CO 2	**Identify different** parameters to analyze the performance of an algorithm.	Level 3 : Apply
CO 3	**Explain** the significance of dynamic memory management Techniques.	Level 2 : Understand
CO 4	**Design** algorithms to perform operations with Linear and Nonlinear data structures	Level 3 : Apply
CO 5	**Illustrate** various technique to for searching, Sorting and hashing	Level 2 : Understand
CO 6	**Choose** appropriate data structures to solve real world problems efficiently.	Level 3 : Apply

Mapping of course outcomes with program outcomes

COs	PROGRAMME OUTCOMES (PO)											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	2	2	-	-	1	-	-	-	-	-	1
CO 2	3	2	2	2	1	1	-	-	-	-	-	1
CO 3	3	3	3	2	1	1	-	-	-	-	-	1
CO 4	3	3	3	2	1	1	-	-	-	-	-	1
CO 5	3	2	2	1	1	-	-	-	-	-	-	1
CO 6	3	3	3	2	1	1	-	-	-	-	-	1

3/2/1: high/medium/low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	10
Understand	20	20	20
Apply	20	20	70
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Differentiate CDT and ADT
2. Classify classical datstructures
3. Compare array and linked list
4. Represent single and double dimensional array
5. Describe any three applications of array

Course Outcome 2 (CO2):

1. Identify the needs of algorithm analysis
2. Select two parameters to do the performance analysis of an algorithm
3. Identify 3 possible cases of time complexity

Course Outcome 3 (CO3):

1. Classify linked list
2. Illustrate different operations on singly, doubly and circular linked list
3. Represent linked list in memory (static and dynamic).

4. Summarize different dynamic memory management schemes.
5. Demonstrate the first fit , best fit , worst fit and next fit allocation of given process queue and free list

Process queue	85 K	35 K	70 K	100 K

	80K	130K	90 K	40 K
Free list				

Course Outcome 4 (CO4):

1. Design the algorithms to perform PUSH()and POP() and STATUS() operations on stack using array and linked list
2. Apply Stack data structure in infix to postfix conversion, expression evaluation, recursion and delimiter matching.
3. Design the algorithms for pre-order, in-order and post-order traversal on binary trees
4. Develop the algorithms for ENQUEUE(), and DEQUEUE() operations on queue data structures
5. Construct the algorithms for graph traversal(BFS, DFS)

Course Outcome 5 (CO5):

1. Classify Sorting Techniques (internal and external, n2 and n logn)
2. Compare Linear and binary search
3. Illustrate bubble, selection and insertion sort.
4. Describe quick and merge sort
5. Represent the following values in the given order in a hash table (Size of hash table is 7 and hash function used is h(k)=k mod 7) for each of the scenario.
   19, 26, 13, 48, 17
   a) When collisions are handled by linear probing
   b) When collisions are handled by double using second hash function h=5-(k mod 5)

Course Outcome 6 (CO6):

1. Develop an application program which is to be used in Ticket counter, where First person gets ticket first and go out first, using suitable data structure.
2. Make use of suitable data structure to store the details of pass percentage of the college in chronological order of years (oldest to newest) and retrieve the information in reverse chronological order of years, using suitable data structure.

Model Question Paper

PART A **(10*3=30)**

(Each Question carries 3 Marks)

1. Classify Sorting Techniques
2. Differentiate CDT and ADT
3. Classify linked list
4. Compare First fit and Next Fit Algorithms
5. Design the algorithms to perform PUSH()and POP()
6. Summarize operations Queue data structure
7. List out the features of binary tree
8. Explain binary sorting
9. Define hashing
10. Illustrate separate chaining with an example

PART B **(5*14=70)**

11. Classify classical data structures

OR

12. Illustrate Quick sort with the help of an example

13. Illustrate different operations on singly linked list

OR

14. Demonstrate the first fit , best fit , worst fit and next fit allocation of given process queue and free list

Process queue

85 K	35 K	70 K	100 K

Free list

80K	130K	90 K	40 K

15. Apply Stack data structure in infix to postfix conversion

OR

16. Develop the algorithms for ENQUEUE(), and DEQUEUE() operations on queue data structures

17. Construct the algorithms for graph traversal(BFS, DFS)

OR

18. Explain 3 types of binary tree traversal

19. Explain Any 3 types of hash functions

OR

20. Represent the following values in the given order in a hash table (Size of hash table is 7 and hash function used is h(k)=k mod 7) for each of the scenario.

    19, 26, 13, 48, 17

    a) When collisions are handled by linear probing

    b) When collisions are handled by double using second hash function h=5-(k mod 5)

Syllabus

Module 1: Introduction to data structures (9 Hours)
Data Structures-Introduction and Overview- Arrays, Algorithm/Program Development, , Searching and Sorting.
Module 2: Linked lists (10 Hours)
Linked lists, singly linked list, Doubly linked list, Circular linked list, Applications of linked list, Dynamic Memory management.
Module 3 : Stacks and Queues (9 Hours)
Stack, Applications of stacks, Queues, Types of queues
Module 4 : Trees and graphs (10 Hours)
Trees, Binary Tree Traversals, Binary tree Applications, Graph, and Graph Applications.
Module 5 : Hash Table (7 Hours)
Hash Tables, Different Hash Functions, Collision Resolution Techniques, closed hashing and Open Hashing (Separate Chaining).

Text Books

T1. Samanta D., Classic Data Structures, Prentice Hall India, 2/e, 2009.

T2. Ellis horowitz, Sartaj Sahni, Fundamentals of Data structures, Galgotia Booksource

Reference Books

R1. Horwitz E., S. Sahni and S. Anderson, Fundamentals of Data Structures in C, University Press (India), 2008.

R2. Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication,1983**.**

R3. Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill, 1995.

R4. Peter Brass, Advanced Data Structures, Cambridge University Press, 2008

R5. Lipschuts S., Theory and Problems of Data Structures, Schaum’s Series, 1986.

R6. Wirth N., Algorithms + Data Structures = Programs, Prentice Hall, 2004.

R7. Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI, 1987.

R8. Martin Barrett, Clifford Wagner, And Unix: Tools For Software Design, John Wiley, 2008 reprint.

Course Contents and Lecture Schedule

	Module 1: Introduction to data structures	**9hrs**
1.1	**Data Structures-Introduction and Overview**: Definitions, Concept of data structure, classifications of data structure- ADT and CDT- Linear and nonlinear.	1
1.2	**Arrays:** definition, Representation of Single/Two dimensional arrays, Applications of array – searching –Sorting - Sparse Matrix- conversion of sparse matrix into 3 tuple form.	2
1.3	**Algorithm/Program Development:** Analysis of algorithms. Space Complexity, Time Complexity - Best case, worst case, average case. **Searching** : linear and binary search – Complexity Analysis (Detailed analysis is not required)	2
1.4	**Sorting**: classifications- Internal sorting – External sorting , N^2 Sorting : Selection, bubble and insertion- Complexity analysis (Detailed analysis is not required)	2
1.5	N log $_n$ Sorting : Quick Sort and Merge Sort (Recursive Algorithms)- Complexity Analysis (Detailed analysis is not required)	2
	Module 2: Linked lists	**10 hrs**
2.1	**Linked lists**: static and dynamic representation, Classification -Singly linked list- Doubly linked list- Circular linked list, array and linked list. **Singly linked list**: Operations on Singly linked list- Traversal-Insertion-deletion, copying -searching - Merging.	2
2.2	**Doubly linked list**: Operations on doubly linked list- Insertion-deletion.	2
2.3	**Circular Linked list** : Operations on circular linked list-Insertion and deletion	2
2.4	**Applications of linked list**: Polynomial representation and manipulation (addition)- Dynamic Memory management.	2
2.5	**Dynamic Memory management**: Fixed sized and variable sized memory allocation and de-allocation. First-fit, best-fit and worst-fit allocation schemes and problems.	2
	Module 3: Stacks and Queues	**9 hrs**
3.1	**Stack:** Definition, Schematic Diagram of stack, Array and Liked list representation of stack , operations on stack using array and linked list (PUSH(),POP(),STATUS()) .	2
3.2	**Applications of stacks**: Infix to postfix conversion- post fix evaluation, string reversal, delimiter matching.	3
3.3	**Queues**: Definition, Schematic Diagram of queue, Array and Liked list representation of queue , operations on queue using array and linked list (EQUEUE(),DEQUEUE(),STATUS()) .	2
3.4	**Types of queue** : circular queue-priority queue- doubly ended queue	2

	Module 4: Trees and graphs	**10 hrs**
4.1	**Trees:** Basic terminologies, Binary Trees, Properties of binary trees, linear and linked representations, Complete and full Binary Tree.	2
4.2	**Binary Tree Traversals**: Preorder -In order and post order (Recursive, non-recursive)-problems	1
4.3	**Binary tree Applications:** Expression tree creation, heap trees (concepts), Binary search tree – creation, insertion and deletion and search operations	3
4.4	**Graph**: Terminologies, set representations, linked/adjacency list representation, Adjacency matrix linear representation **Graph traversal**: Breadth First Search (BFS), Depth First Search (DFS) - related problems.	2
4.5	**Graph Applications:** Shortest Path Problem-Dijkstras Algorithm	2
	Module 5: Hash Table	**7 hrs**
5.1	**Hash Tables**-Hash Functions- Features of hash function.	1
5.2	**Different Hash Functions:** Division Method- Multiplication Method - Mid Square Method, Folding Method- related problems.	2
5.3	**Collision Resolution Techniques:** Closed hashing (Linear probing) and Open Hashing (Separate Chaining) . Closed hashing(Linear probing) -Drawbacks- Remedies - Radom Probing – Double hashing/Re-hashing –Quadratic Probing, problems to create hash tables using linear probing and Random probing, double hash and quadratic probing .	3
5.4	Open Hashing (Separate Chaining)	1