INSTITUTE OF AERONAUTICAL ENGINEERING

(AUTONOMOUS)

Dundigal, Hyderabad - 500 043

INFORMATION TECHNOLOGY

COURSE DESCRIPTION FORM

Course Title	**DATA STRUCTURES**			
Course Code	A30502			
Regulation	**R13 - JNTUH**			
Course Structure	Lectures	Tutorials	Practicals	Credits
	4	-	-	4
Course Coordinator	Mr. Ch. Suresh Kumar Raju, Associate Professor			
Team of Instructors	Mr. Ch. Suresh Kumar Raju, Associate Professor			

I. COURSE OVERVIEW:

Data Structures is a subject of primary importance to the discipline of Information Technology. It is a logical and mathematical model of sorting and organizing data in a particular way in a computer, required for designing and implementing efficient algorithms and program development. Different kinds of data structures like arrays, linked lists, stacks, queues, etc. are suited to different kinds of applications. Some specific data structures are essential ingredients of many efficient algorithms, and make possible the management of huge amounts of data, such as large databases and internet indexing services. Nowadays, various programming languages like C, C++ and Java are used to implement the concepts of Data Structures, of which C remains the language of choice for programmers across the world.

II. PREREQUISITE(S):

Level	Credits	Periods/ Week	Prerequisites
UG	4	4	Computer Programming

III. MARKS DISTRIBUTION:

Sessional Marks	University End Exam marks	Total marks
Midterm Test There shall be two midterm examinations. Each midterm examination consists of essay paper, objective paper and assignment. The essay paper is for 10 marks of 60 minutes duration and shall contain 4 questions. The student has to answer 2 questions, each carrying 5 marks. The objective paper is for 10 marks of 20 minutes duration. It consists of 10 multiple choice and 10 fill-in-the blank questions, the student has to answer all the questions and each carries half mark. First midterm examination shall be conducted for the first two and half units of syllabus and second midterm examination shall be conducted for the remaining portion. Five marks are earmarked for assignments. There shall be two assignments in every theory course. Assignments are usually issued at the time of commencement of the semester. These are of problem solving in nature with	75	100

Sessional Marks	University End Exam marks	Total marks
critical thinking. Marks shall be awarded considering the average of two midterm tests in each course.		

IV. EVALUATION SCHEME:

S. No	Component	Duration	Marks
1.	I Mid Examination	80 minutes	20
2.	I Assignment	-	5
3.	II Mid Examination	80 minutes	20
4.	II Assignment	-	5
5.	External Examination	3 hours	75

V. COURSE OBJECTIVES:

I. Be familiar with basic techniques of algorithm analysis.
II. Be familiar with writing recursive methods.
III. Master the implementation of linked data structures such as linked lists, binary trees and graphs.
IV. Be familiar with binary tree traversal algorithms such as inorder, preorder and postorder and graph traversal and search algorithms.
V. Be familiar with advanced data structures such as balanced search trees, hash tables and priority queues.
VI. Be familiar with several searching and sorting algorithms including quicksort, mergesort and heapsort
VII. Master analyzing and writing program solutions to problems using the above techniques

VI. COURSE OUTCOMES:

At the end of the course the students are able to:

1. **Understand** the concept of recursion and describe its implementation using a stack.
2. **Compare** iterative and recursive solutions for elementary problems.
3. **Describe** the usage and operations for maintaining various data structures.
4. **Choose** the appropriate data structure for a specific application.
5. **Apply** the notations used to analyze the performance of algorithms.
6. **Describe** various data structures like Stacks, Queues, Linked lists, Trees and Graphs are represented in memory and used by algorithms.
7. **Write** programs that use various data structures like Stacks, Queues, Linked lists, Trees and Graphs.
8. **Compare and contrast** the time complexities of various searching and sorting algorithms.
9. **Design and implement** an appropriate hashing function for an application.
10. **Apply** tree and graph traversal methods in real time applications.
11. **Describe** the concept of recursion, give examples of its use, describe how it can be implemented using a stack.
12. **Apply** basic algorithm strategies and to design algorithms for concrete problems of reasonable difficulty.

VII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	H	Assignments, Tutorials
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	H	Assignments
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	S	Mini Projects
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	S	Projects
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	S	Mini Projects
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	N	--
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	N	--
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	N	--
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	N	--
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	N	--
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	N	--
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	S	Projects

N - None S - Supportive H - Highly Related

VIII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Level	Proficiency assessed by
PSO1	**Professional Skills:** The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.	H	Lectures, Assignments
PSO2	**Software Engineering practices:** The ability to apply standard practices and strategies in software service management using open-ended programming environments with agility to deliver a quality service for business success.	H	Projects
PSO3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	S	Guest Lectures

N - None S - Supportive H - Highly Related

IX. SYLLABUS:

UNIT – I

Basic concepts- Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction Performance analysis- time complexity and space complexity, Asymptotic Notation-Big O, Omega and Theta notations, Introduction to Linear and Non Linear data structures.

Singly Linked Lists-Operations-Insertion, Deletion, Concatenating singly linked lists, Circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations- Insertion, Deletion. Representation of single, two dimensional arrays, sparse matrices-array and linked representations.

UNIT – II

Stacks and Queue- Stack ADT, definition, operations, array and linked implementations in C, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation, Queue ADT, definition and operations ,array and linked Implementations in C, Circular queues-Insertion and deletion operations, Deque (Double ended queue)ADT, array and linked implementations in C.

UNIT – III

Trees – Terminology, Representation of Trees, Binary tree ADT, Properties of Binary Trees, Binary Tree Representations-array and linked representations, Binary Tree traversals, Threaded binary trees, Max Priority Queue ADT-implementation-Max Heap-Definition, Insertion into a Max Heap, Deletion from a Max Heap.

Graphs – Introduction, Definition, Terminology, Graph ADT, Graph Representations- Adjacency matrix, Adjacency lists, Graph traversals- DFS and BFS.

UNIT – IV

Searching- Linear Search, Binary Search, Static Hashing-Introduction, hash tables, hash functions, Overflow Handling.

Sorting-Insertion Sort, Selection Sort, Radix Sort, Quick sort, Heap Sort, Comparison of Sorting methods.

UNIT – V

Search Trees-Binary Search Trees, Definition, Operations- Searching, Insertion and Deletion, AVL Trees-Definition and Examples, Insertion into an AVL Tree ,B-Trees, Definition, B-Tree of order m, operations-Insertion and Searching, Introduction to Red-Black and Splay Trees(Elementary treatment-only Definitions and Examples), Comparison of Search Trees.

Pattern matching algorithm- The Knuth-Morris-Pratt algorithm, Tries (examples only).

Text books:

1. Anderson-Freed, Susan, Ellis Horowitz, and Sartaj Sahni, "Fundamentals of Data Structures in C", Universities Press, 2e, 1992.
2. D. S. Kushwaha and A. K. Misra, "Data structures-A Programming Approach with C", PHI Learning, 2012.

References:

1. R. F. Gilberg And B. A. Forouzan, "Data structures: A Pseudocode Approach with C", 2e, Cengage Learning.
2. M. A. Weiss, "Data structures and Algorithm Analysis in C", 2e, Pearson.
3. A. M. Tanenbaum,Y. Langsam, M. J. Augenstein, "Data Structures using C", Pearson.
4. R. Kruse, C. L. Tondo and B. Leung, "Data structures and Program Design in C", 2e, Pearson.
5. R. Thareja, "Data Structures using C", Oxford University Press.
6. S. Lipscutz, "Data Structures", Schaum's Outlines, TMH.
7. A. K. Sharma, "Data structures using C", 2e, Pearson.
8. D. Samantha, "Classic Data Structures", 2e, PHI.

X. COURSE PLAN:

At the end of the course, the students are able to achieve the following course learning outcomes.

Lecture No.	Topics to be covered	Course Learning Outcomes	Reference
1 – 3	**Basic concepts**- Algorithm Specification-Introduction and Recursive algorithms.	**Differentiate** iterative and recursive algorithm development procedures. Develop algorithms using recursive principle.	T1: 1.3
4 – 6	Performance analysis- time complexity and space complexity. Asymptotic Notation-Big O, Omega and Theta notations.	**Analyze** the algorithm and determine the time and space complexity.	T1: 1.5
7	Introduction to Abstract Data Type, Linear and Non Linear data structures.	**Understand** the importance of structure and abstract data type, and their basic usability in different applications.	T1: 2.1
8 – 14	**Singly Linked Lists**-Operations-Insertion, Deletion, Concatenating singly linked lists. Circularly linked lists-Operations for Circularly linked lists. Doubly Linked Lists- Operations- Insertion, Deletion.	**Understand** basic operations of single, double and circular linked lists.	T1: 4.1, 4.2, 4.5, 4.8
15 – 16	Representation of single, two dimensional arrays, sparse matrices-array and linked representations.	**Apply** the concept of arrays and linked list in problem solving.	T1: 2.1, 2.2, 2.5, 4.7
17 – 19	**Stacks and Queue**- Stack ADT, definition, operations, array and linked implementations in C.	**Identify** basic operations of Stack and its implementation.	T1: 3.1, 3.2, 4.3
20 – 23	Applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation.	**Apply** Stack concepts in algebraic expression conversion.	T1: 3.6
24 – 25	Queue ADT, definition and operations ,array and linked Implementations in C.	**Identify** basic operations of Queue and its implementation.	T1: 3.3, 4.3
26 – 28	Circular queues-Insertion and deletion operations, Deque (Double ended queue)ADT, array and linked	**Identify** basic operations of Circular Queue, DEQUE and its implementation.	T1: 3.4, 4.3

	implementations in C.		
29 – 32	**Trees** – Terminology, Representation of Trees, Binary tree ADT. Properties of Binary Trees, Binary Tree Representations-array and linked representations.	**Understand** basic operations of Tree and its representation.	T1: 5.1, 5.2
33 – 35	Binary tree traversals, Threaded binary trees.	**Construct** and traverse the binary tree.	T1: 5.3, 5.5
36 – 39	Max Priority Queue ADT-implementation-Max Heap-Definition, Insertion into a Max Heap, Deletion from a Max Heap.	**Understand** and Construct Priority Queue and Max Heap.	T1: 5.6
40 – 43	**Graphs** – Introduction, Definition, Terminology, Graph ADT. Graph Representations- Adjacency matrix, Adjacency lists, Graph traversals- DFS and BFS.	**Understand** the graph basics, its representation and traversal and search.	T1: 6.1, 6.2
44 – 45	**Searching**- Linear Search and Binary Search.	**Apply** searching techniques in different scenarios.	T1: 1.3
46 - 48	Static Hashing-Introduction, hash tables, hash functions, Overflow Handling.	**Understand** the need of hashing for fast retrieval of data.	T1: 8.2
49 – 53	**Sorting**-Insertion Sort, Selection Sort, Radix Sort, Quick sort, Heap Sort, Comparison of Sorting methods.	**Use** various kinds of sorting techniques and know when to choose which technique.	T1: 1.3, 7.2, 7.3, 7.6, 7.9
54 – 56	**Search Trees**-Binary Search Trees, Definition, Operations- Searching, Insertion and Deletion. **AVL Trees**-Definition and Examples, Insertion into an AVL Tree.	**Understand** and Summarize the operations of binary search trees and AVL trees.	T1: 5.7, 10.2
57 – 58	B-Trees, Definition, B-Tree of order m, operations-Insertion and Searching.	**Understand** need of height balanced trees and Construct B-Tree of order m.	T1: 11.1, 11.2
59 – 60	Introduction to Red-Black and Splay Trees (Elementary treatment-only Definitions and Examples), Comparison of Search Trees.	**Compare** different types of Search Trees.	T1: 10.3, 10.4
61 - 62	**Pattern matching algorithm**- The Knuth-Morris-Pratt algorithm, Tries.	**Understand** basics pattern matching algorithm.	T1: 2.7

XI. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Objectives	Program Outcomes												Program Specific Outcomes		
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**	**PSO3**
I	H	H										S	H	S	
II	S	H	H										H	S	
III		H	S	S									S	H	
IV	H	S											H	S	
V					S								H		S
VI		H	H		S								H	S	S
VII	S	S	H									S	H	H	S

S - Supportive **H - Highly Related**

XII. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	H	S	S										H	S	
2	H			S									S	H	
3			H		S								H	S	
4	S	H											S	H	
5	H	S											S	H	
6	H			S								S	H	S	
7	S			H									S	H	
8	S	H											H	S	
9			H	H	S							S	S	H	
10	H			S									S	H	S
11	H			S	S								H	S	
12	H		H									S	S	H	S

S - Supportive **H - Highly Related**

Prepared by: Mr. Ch. Suresh Kumar Raju, Associate Professor

Date:

HOD, IT