INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad -500 043

ELECTRICAL AND ELECTRONICS ENGINEERING
COURSE DESCRIPTOR

Course Title	**DATA STRUCTURES**				
Course Code	ACS002				
Programme	B.Tech				
Semester	II	CSE | IT | ECE | EEE			
Course Type	Foundation				
Regulation	IARE - R16				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3	1	4	3	2
Chief Coordinator	Ms. B Padmaja, Associate Professor				
Course Faculty	Dr. J Sirisha Devi, Professor Ms. N Jayanthi, Assistant Professor Ms. G Vasavi, Assistant Professor Ms. K Radhika, Assistant Professor Ms. G Geetha, Assistant Professor Ms. B Rekha, Assistant Professor Ms. A Soujanya, Assistant Professor Mr. D Rahul, Assistant Professor				

I. COURSE OVERVIEW:

This course covers some of the general-purpose data structures and algorithms, and software development. It is aimed at helping students understand the reasons for choosing structures or algorithms. Topics covered include managing complexity, analysis, lists, stacks, queues, trees, graphs, balanced search trees and hashing mechanisms. The main objective of the course is to teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter in real life. This course in reached to student by power point presentations, lecture notes, and lab involve the problem solving in mathematical and engineering areas.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites	Credits
UG	ACS001	I	Computer Programming	3

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Data Structures	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✔	Chalk & Talk	✔	Quiz	✔	Assignments	✘	MOOCs
✔	LCD / PPT	✔	Seminars	✔	Mini Project	✔	Videos
✔	Open Ended Experiments						

V. EVALUATION METHODOLOGY:

The course will be evaluated for a total of 100 marks, with 30 marks for Continuous Internal Assessment (CIA) and 70 marks for Semester End Examination (SEE). Out of 30 marks allotted for CIA during the semester, marks are awarded by taking average of two CIA examinations or the marks scored in the make-up examination.

Semester End Examination (SEE): The SEE is conducted for 70 marks of 3 hours duration. The syllabus for the theory courses is divided into FIVE units and each unit carries equal weightage in terms of marks distribution. The question paper pattern is as follows. Two full questions with "either" or "choice" will be drawn from each unit. Each question carries 14 marks. There could be a maximum of two sub divisions in a question.

The emphasis on the questions is broadly based on the following criteria:

50 %	To test the objectiveness of the concept.
50 %	To test the analytical skill of the concept OR to test the application skill of the concept.

Continuous Internal Assessment (CIA):
CIA is conducted for a total of 30 marks (Table 1), with 25 marks for Continuous Internal Examination (CIE), 05 marks for Quiz/ Alternative Assessment Tool (AAT).

Table 1: Assessment pattern for CIA

Component	Theory		Total Marks
Type of Assessment	CIE Exam	Quiz / AAT	
CIA Marks	25	05	30

Continuous Internal Examination (CIE):
Two CIE exams shall be conducted at the end of the 8th and 16th week of the semester respectively. The CIE exam is conducted for 25 marks of 2 hours duration consisting of two parts. Part–A shall have five compulsory questions of one mark each. In part–B, four out of five questions have to be answered where, each question carries 5 marks. Marks are awarded by taking average of marks scored in two CIE exams.

Quiz / Alternative Assessment Tool (AAT):
Two Quiz exams shall be online examination consisting of 25 multiple choice questions and are be answered by choosing the correct answer from a given set of choices (commonly four). Marks shall be awarded considering the average of two quizzes for every course. The AAT may include seminars, assignments, term paper, open ended experiments, five minutes video and MOOCs.

VI. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		Strength	Proficiency assessed by
PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3	Assignments/Quiz
PO 2	**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	3	Assignments/Quiz
PO 3	**Design/development of solutions**: Design solutions for complex engineering problems anddesign 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.	2	Mini Project
PO 5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modernengineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	2	Mini Project
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage independent and life-long learning in the broadest context of technological change.	2	-

3 = High; 2 = Medium; 1 = Low

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Strength	Proficiency assessed by
PSO 1	**Problem Solving:** Exploit the knowledge of high voltage engineering in collaboration with power systems in innovative, dynamic and challenging environment, for the research based team work.	1	-
PSO 2	**Professional Skills:** Identify the scientific theories, ideas, methodologies and the new cutting edge technologies in renewable energy engineering, and use this erudition in their professional development and gain sufficient competence to solve the current and future energy problems universally.	2	Assignments
PSO 3	**Modern Tools in Electrical Engineering:** Comprehend the technologies like PLC, PMC, process controllers, transducers and HMI and design, install, test, maintain power systems and industrial applications.	1	-

3 = High; 2 = Medium; 1 = Low

VIII. COURSE OBJECTIVES (COs):

The course should enable the students to:	
I	Learn the basic techniques of algorithm analysis.
II	Demonstrate searching and sorting algorithms and analyze their time complexities.
III	Implementlinear data structures viz. stack, queue and linked list.
IV	Demonstrate non-linear data structures viz. tree and graph traversal algorithms.
V	Study and choose appropriate data structure to solve problems in real world.

IX. COURSE LEARNING OUTCOMES (CLOs):

CLO Code	CLO's	At the end of the course, the student will have the ability to:	PO's Mapped	Strength of Mapping
CCS002.01	CLO 1	Understand algorithms and data structures in terms of time and space complexity of basic operations.	PO 1	3
ACS002.02	CLO 2	Analyze a given problem; choose an appropriate data structure and an algorithm to solve the problem.	PO 3,PO4	2
ACS002.03	CLO 3	Choose a suitable algorithm to organize the data in ascending or descending order.	PO 2,PO5	2
ACS002.04	CLO 4	Understand the difference between iterative and recursion approaches to solve problems.	PO 1	3
ACS002.05	CLO 5	Explore an algorithm to find the location of an element in a given list.	PO 2, PO3,PO5	3
ACS002.06	CLO 6	Understand the usage of divide and conquer strategy in searching and sorting applications.	PO 1,PO2	3
ACS002.07	CLO 7	Compare the time complexities of various searching and sorting algorithms.	PO 1,PO5	2
ACS002.08	CLO 8	Understand the working principle of linear data structures and their real time applications.	PO1,PO 2, PO5	3
ACS002.09	CLO 9	Organize the data in various linked representation format.	PO1,PO 2, PO5	3
ACS002.10	CLO 10	Design and implement abstract data types for linear and non-linear data structures.	PO1,PO 2	3
ACS002.11	CLO 11	Describe the concept of non-linear data structures viz. trees and graphs and their applications.	PO 1,PO2	2
ACS002.12	CLO 12	Compare and Contrast the operations of binary search trees and AVL trees.	PO 1	3
ACS002.13	CLO 13	Understand the concept of M-way search trees, operations and applications.	PO1,PO 2, PO5	3
ACS002.14	CLO 14	List out different tree and graph traversal techniques.	PO1,PO 2,PO5	2
ACS002.15	CLO 15	Understand the implementation of hashing using hash table and hash function.	PO1,PO 2, PO5	3
ACS002.16	CLO 16	Describethe concept of collision and its resolving methodsin applications.	PO1,PO 2, PO5	2
ACS002.17	CLO 17	Strengthen the knowledge of data structures and algorithms for employability.	PO1,PO 2, PO5	2

3 = High; 2 = Medium; 1 = Low

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

CLOs	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO 1	3													2	
CLO 2		3	2												
CLO 3		3			2									3	
CLO 4	3												2		
CLO 5		3	3		2									2	
CLO 6	3	3													
CLO 7	3				2										
CLO 8	3	3	2											3	
CLO 9	2	3			2										
CLO 10	3	3												2	
CLO 11	3	3												3	
CLO 12	3														2
CLO 13	3	3			2								1		
CLO 14	2	3			2									3	
CLO 15	3	3			2								1		
CLO 16	2	3			2							2		3	
CLO 17	2	3			2							3		3	

3 = High; 2 = Medium; 1 = Low

XI. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO1, PO2,PO3, PO5	SEE Exams	PO 1	Assignments	PO 1	Seminars	PO 2
Laboratory Practices	PO 1	Student Viva	PO 1	Mini Project	PO 1	Certification	-
Term Paper	PO 4						

XII. ASSESSMENT METHODOLOGIES - INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE Feedback
✘	Assessment of Mini Projects by Experts		

XIII. SYLLABUS

UNIT-I	INTRODUCTION TO DATA STRUCTURES, SEARCHING AND SORTING
Basic concepts: Introduction to data structures, classification of data structures, operations on data structures, abstract data type, algorithms, different approaches to design an algorithm, recursive algorithms; Searching techniques: Linear search, binary search and Fibonacci search; Sorting techniques: Bubble sort, selection sort, insertion sort, quick sort, merge sort, and comparison of sorting algorithms.	
UNIT-II	**LINEAR DATA STRUCTURES**
Stacks: Primitive operations, implementation of stacks using Arrays, applications of stacks arithmetic expression conversion and evaluation; Queues: Primitive operations; Implementation of queues using Arrays, applications of linear queue, circular queue and double ended queue (deque).	
UNIT-III	**LINKED LISTS**
Linked lists: Introduction, singly linked list, representation of a linked list in memory, operations on a single linked list; Applications of linked lists: Polynomial representation and sparse matrix manipulation. Types of linked lists: Circular linked lists, doubly linked lists; Linked list representation and operations of Stack, linked list representation and operations of queue.	
UNIT-IV	**NON LINEAR DATA STRUCTURES**
Trees: Basic concept, binary tree, binary tree representation, array and linked representations, binary tree traversal, binary tree variants, application of trees; Graphs: Basic concept, graph terminology, graph implementation, graph traversals, Application of graphs, Priority Queue.	
UNIT-V	**BINARY TREES AND HASHING**
Binary search trees: Binary search trees, properties and operations; Balanced search trees: AVL trees; Introduction to M-Way search trees, B trees; Hashing and collision: Introduction, hash tables, hash functions, collisions, applications of hashing.	
Text Books:	
1. Rance D. Necaise, "Data Structures and Algorithms using Python", Wiley Student Edition. 2. Benjamin Baka, David Julian, "Python Data Structures and Algorithms", Packt Publishers, 2017.	
Reference Books:	
1. S. Lipschutz, "Data Structures", Tata McGraw Hill Education, 1st Edition, 2008. 2. D. Samanta, "Classic Data Structures", PHI Learning, 2nd Edition, 2004. 3. Y Daniel Liang, "Introduction to Programming using Python", Pearson. 4. Martin Jones, "Python for Complete Beginners", 2015. 5. Zed A. Shaw, "Learn Python the Hard Way: a very simple introduction to the terrifyingly beautiful world of computers and code", 3e, Addison-Wesley, 2014. 6. Hemant Jain, "Problem Solving in Data Structures and Algorithms using Python: programming interview guide", 2016.	

XIV. COURSE PLAN:

The course plan is meant as a guideline. Probably there may be changes.

Lecture No	Topics to be covered	Course Learning Outcomes (CLOs)	Reference
1 – 2	Basic concepts: Introduction to Data Structures.	CLO 1	T1:1.1.3 R2 : 1.2
3 – 4	Classification of data structures, operations on data Structures.	CLO 2	T1:1.1.3 R2 : 1.4
5 – 6	Abstract data type, Algorithms, Different approaches to design an Algorithm.	CLO 2	T1:1.2
7 – 8	Recursive algorithms ,Searching techniques: Linear search, binary search	CLO 4	T1:5.1
9 – 10	Fibonacci search	CLO 4	R1:14.5

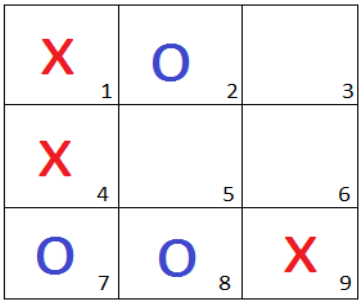
Lecture No	Topics to be covered	Course Learning Outcomes (CLOs)	Reference
11 – 14	Sorting techniques: Bubble sort, selection sort, insertion sort, and quick sort; merge sort, and comparison of sorting algorithms.	CLO 7	T1:5.2 R2 : 10.2
15 – 16	Stacks: Primitive operations, implementation of stacks using Arrays.	CLO 9	T1:7.1
17 – 20	Applications of stacks arithmetic expression conversion and evaluation.	CLO 9	T1:7.2
21 – 22	Queues: Primitive operations; Implementation of queues using Array.	CLO 11	T1:8.1
23 – 24	Applications of linear queue circular queue.	CLO 11	T1:8.4
25 – 26	Double ended queue (deque).	CLO 13	R2 : 5.4
27 – 28	Linked lists: Introduction, singly linked list, representation of a linked list in memory.	CLO 11	T1:9.1
29– 30	Operations on a single linked list, Applications of linked lists: Polynomial representation, Circular linked lists, doubly linked lists;	CLO 9	T1:9.2
31 - 32	Sparse matrix manipulation.	CLO 14	T2:9.2
33 – 35	Linked list representation and operations of Stack, Linked list representation and operations of queue.	CLO 14	T1:9
36 – 38	Trees: Basic concept, binary tree, binary tree representation, array and linked representations	CLO 14	T1:13.1-13.2
39 – 40	Binary tree traversal, binary tree variants, application of trees.	CLO 14	T1:13.2.3
41 – 43	Graphs: Basic concept, graph terminology, graph implementation.	CLO 14	R2 : 8.2
44 – 46	Graph traversals, Application of graphs,	CLO 17	T2:6.2
47 – 50	Priority Queue.	CLO 17	T1:6.1 T2:5.6
51 – 52	Binary search trees, properties and operations.	CLO 19	T1:14.1
53 – 55	Balanced search trees: AVL trees, Introduction to M-Way search trees, B trees.	CLO 19	T1:14.3
56 – 58	Hashing and collision: Introduction, hash tables, hash functions,	CLO 20	R2 : 6.4
59 - 60	Collisions, applications of hashing.	CLO 20	R2 : 6.4

XV. GAPS IN THE SYLLABUS - TO MEET INDUSTRY / PROFESSION REQUIREMENTS:

S No	Description	Proposed actions	Relevance with POs	Relevance with PSOs
1	Updating latest version and new features of the Python language	Laboratory Sessions	PO5	PSO2
2	Familiarizing the role of Python language in developing application level programs.	Assignments/ Industrial visits	PO1,PO2	PSO2
3	Familiarizing different areas where Python language can be used.	Seminars	PO12	PSO3
4	Solving different problems and Practicing various debugging strategies to become a good programmer	Extra Lab Sessions, Participating in Coding contests.	PO2	PSO3

XVI. DESIGN BASED PROBLEMS (DP) / OPEN ENDED PROBLEM:

1. **Check a Single Linked List is Palindrome or not**: Given a singly linked list of integers, the task is to complete the function **is Palindrome** which returns true if the given list is palindrome, else returns false. The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains 2 line the first line of each test case contains an integer N denoting the size of the linked list. The next line contains N space separated integers denoting the values of the nodes of the linked list.
2. **Tic-Tac-Toe Game**: The game of Tic-Tac-Toe is being played between two players and it is in below state after six moves.



Answer the following questions?

a. Who will win the game, O or X?
b. Which was the sixth mark and at which position?

Assume that both the players are intelligent enough.

3. **Lowest Common Ancestor in a Binary Tree:** Given a Binary Tree and 2 nodes value n1 and n2. The task is to find the lowest common ancestor of the two nodes. You are required to complete the function LCA. You should not read any input from stdin/console. There are multiple test cases. For each test case, this method will be called individually.
   The task is to complete the method LCA which takes 3 arguments, root of the Tree and two nodes value n1 and n2. The struct node has a data part which stores the data, pointer to left child and pointer to right child. There are multiple test cases. For each test case, this method will be called individually. The function should return the node which is the least common ancestor of the two nodes n1 and n2.

Prepared by:
B Padmaja, Associate Professor, CSE

HOD, FRESHMAN ENGINEERING