INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad -500 043

INFORMATION TECHNOLOGY

COURSE DESCRIPTOR

Course Title	**DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY**				
Course Code	ACSB07				
Programme	B. Tech				
Semester	IV	CSE | IT			
Course Type	Core				
Regulation	IARE - R18				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	-	-	-	3	1.5
Chief Coordinator	Ms. E Uma Shankari, Assistant Professor				
Course Faculty	Dr.M Purushotham , Associate Professor Mrs. G Geetha, 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	ACSB03	III	Data Structures	3

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Design and analysis of algorithm Laboratory	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:

Each laboratory will be evaluated for a total of 100 marks consisting of 30 marks for internal assessment and 70 marks for semester end lab examination. Out of 30 marks of internal assessment, continuous lab assessment will be done for 20 marks for the day to day performance and 10 marks for the final internal lab assessment.

Semester End Examination (SEE): The semester end lab examination for 70 marks shall be conducted by two examiners, one of them being Internal Examiner and the other being External Examiner, both nominated by the Principal from the panel of experts recommended by Chairman, BOS.

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

20 %	To test the preparedness for the experiment.
20 %	To test the performance in the laboratory.
20 %	To test the calculations and graphs related to the concern experiment.
20 %	To test the results and the error analysis of the experiment.
20 %	To test the subject knowledge through viva – voce.

Continuous Internal Assessment (CIA):
CIA is conducted for a total of 30 marks (Table 1), with 20 marks for continuous lab assessment during day to day performance, 10 marks for final internal lab assessment.

Table 1: Assessment pattern for CIA

Component	**Laboratory**		**Total Marks**
Type of Assessment	Day to day performance	Final internal lab assessment	
CIA Marks	20	10	30

Continuous Internal Examination (CIE):
One CIE exams shall be conducted at the end of the 16th week of the semester. The CIE exam is conducted for 10 marks of 3 hours duration.

Preparation	**Performance**	**Calculations and Graph**	**Results and Error Analysis**	**Viva**	**Total**
2	2	2	2	2	10

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	Videos/ Student Viva
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	Lab Exercises/ StudentViva
PO 3	**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.	2	Videos/ StudentViva
PO 5	**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.	2	Lab Exercises

3 = High; 2 = Medium; 1 = Low

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Strength	Proficiency assessed by
PSO1	**Professional Skills:** Able to utilize the knowledge of aeronautical/aerospace engineering in innovative, dynamic and challenging environment for design and development of new products	3	Videos
PSO2	**Practical implementation and testing skills:** Providing different types of in house and training and industry practice to fabricate and test and develop the products with more innovative technologies	3	Lab Exercises
PSO3	**Successful Career and Entrepreneurship:** To prepare the students with broad aerospace knowledge to design and develop systems and subsystems of aerospace and allied systems and become technocrats	-	-

3 = High; 2 = Medium; 1 = Low

VIII. COURSE OBJECTIVES :

The course should enable the students to:	
I	Learn how to analyze a problem and design the solution for the problem.
II	Design and implement efficient python programming for a specified application.
III	Identify and apply the suitable algorithm for the given real world problem.

IX. COURSE OUTCOMES (COs):

Cos	Course Outcome	CLOs	Course Learning Outcome
CO 1	Implement Quick sort, Merge sort and Warshall's algorithm.	CLO 1	Understand the basic concepts of python.
		CLO 2	Understand the different sorting techniques to organize the data in ascending or descending order using quick sort and merge sort.
		CLO 3	Computing the transitive closure of a given directed graph using Warshall's algorithm.
CO 2	Implement Dynamic Programming algorithm for the 0/1 Knapsack problem and greedy algorithm for job sequencing with deadlines.	CLO 4	Implementation of dynamic programming for knapsack problem.
CO 3	Implement Dijkstra's , Prim's, Kruskal's algorithm on spanning tree.	CLO 5	Identify the shortest paths to other vertices using Dijkstra's algorithm.
		CLO 6	Analyze the concept of minimum cost spanning trees using Kruskal's algorithm
		CLO 11	Analyze the concept of minimum cost spanning trees using Prim's algorithm
CO 4	Implement Tree Traversal and Graph Traversals techniques using BFS and DFS.	CLO 7	Implementation of tree traversal algorithms for given graphs.
		CLO 8	Understand graphs and graph traversal techniques like Depth first search and Breadth first search.
CO 5	Implement Floyd's algorithm for the all pair's shortest path problem and N-queens problem..	CLO 9	Understand and implement the sum of subsets problem
		CLO 10	Implement the travelling salesperson problem.
		CLO 11	Implementation of All-Pairs Shortest Paths Problem using Floyd's algorithm and N-Queens problem

X. 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
ACSB07.01	CLO 1	Understand the basic concepts of python.	PO 1	3
ACSB07.02	CLO 2	Understand the different sorting techniques to organize the data in ascending or descending order using quick sort and merge sort.	PO 3,	3
ACSB07.03	CLO 3	Computing the transitive closure of a given directed graph using Warshall's algorithm.	PO 3,PO 5	3
ACSB07.04	CLO 4	Implementation of dynamic programming for knapsack problem.	PO 1, PO 5	3
ACSB07.05	CLO 5	Identify the shortest paths to other vertices using Dijkstra's algorithm.	PO 1,PO 5	3
ACSB07.06	CLO 6	Analyze the concept of minimum cost spanning trees using Kruskal's algorithm	PO 3,PO 5	3
ACSB07.07	CLO 7	Implementation of tree traversal algorithms for given graphs.	PO 3,PO 5	3
ACSB07.08	CLO 8	Understand graphs and graph traversal techniques like Depth first search and Breadth first search.	PO 3,PO 5	3
ACSB07.09	CLO 9	Understand and implement the sum of subsets problem.	PO 1,PO 5	3
ACSB07.10	CLO 10	Implement the travelling salesperson problem.	PO 2,PO 5	3
ACSB07.11	CLO 11	Analyze the concept of minimum cost spanning trees using Prim's algorithm	PO 2,PO 5	3
ACSB07.12	CLO 12	Implementation of All-Pairs Shortest Paths Problem using Floyd's algorithm and N-Queens problem	PO 2, PO 3	3

3= High; 2 = Medium; 1 = Low

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

Course Outcomes (COs)	Program Outcomes (POs)				Program Specific Outcomes(PSOs)		
	PO1	PO2	PO3	PO5	PSO1	PSO2	PSO3
CO 1	3		3	2	3		
CO 2	3			2		3	
CO 3	3		3	2		3	
CO 4		3		2			
CO 5		3	3	2		3	

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

Course0 Learning 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														
CLO 2			3										3		
CLO 3			3		2								3		
CLO 4	3				2										
CLO 5	3				2									3	
CLO 6	3				2								3		
CLO 7			3		2								3		
CLO 8			3		2								3		
CLO 9	3				2								3		
CLO 10		3			2								3		
CLO 11		3			2										
CLO 12		3	3		2									3	

3 = High; 2 = Medium; 1 = Low

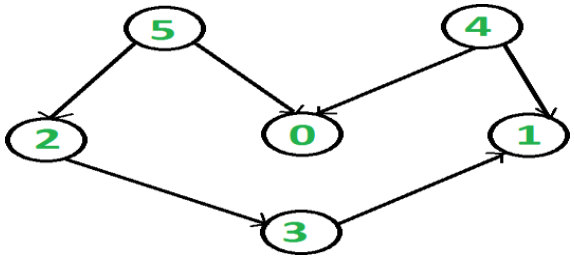
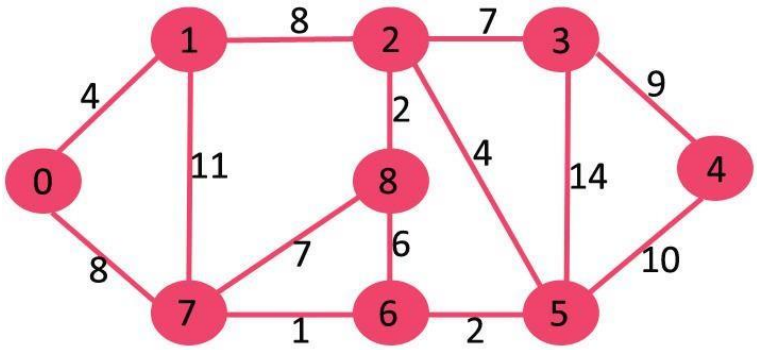
XIII. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO 1,PO2, PO3, PO5	SEE Exams	PO 1, PO 2 PO 3, PO 5	Lab Exercises	PO 1, PO2, PO 3, PO 5	Seminars	PO 1, PO 2, PO3, PO5
Laboratory Practices	PO 1, PO 2, PO 3, PO 5	Student Viva	PO 1, PO 2 PO 3, PO 5	Mini Project	-	Certification	-

XIV. ASSESSMENT METHODOLOGIES - INDIRECT

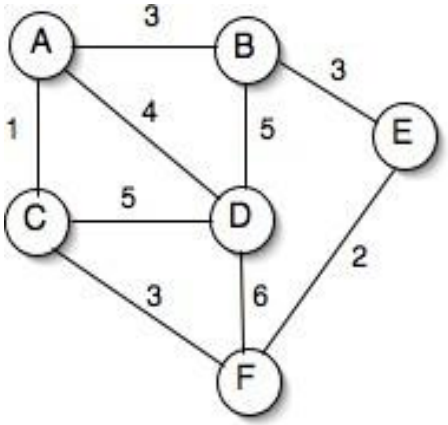
✔	Early Semester Feedback	✔	End Semester OBE Feedback
✘	Assessment of Mini Projects by Experts		

XV. SYLLABUS

LIST OF EXPERIMENTS	
WEEK-1	**QUICK SORT**
Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the 1st to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.	
WEEK-2	**MERGE SORT**
Implement merge sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.	
WEEK-3	**WARSHALL'S ALGORITHM**
a) Obtain the Topological ordering of vertices in a given digraph.  b) Compute the transitive closure of a given directed graph using Warshall's algorithm.	
WEEK-4	**KNAPSACK PROBLEM**
Implement 0/1 Knapsack problem using Dynamic Programming.	
WEEK-5	**SHORTEST PATHS ALGORITHM**
From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.  	

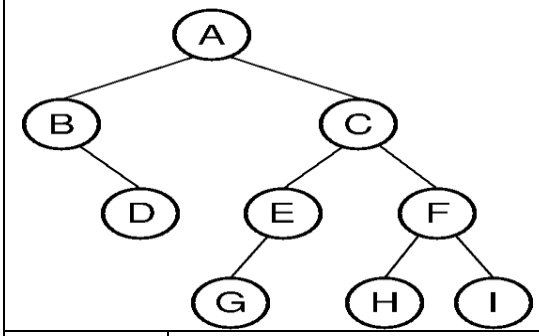
WEEK-6	MINIMUM COST SPANNING TREE

Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.



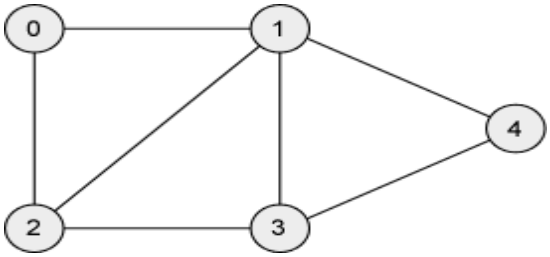
WEEK-7	TREE TRAVESRSALS

Perform various tree traversal algorithms for a given tree.

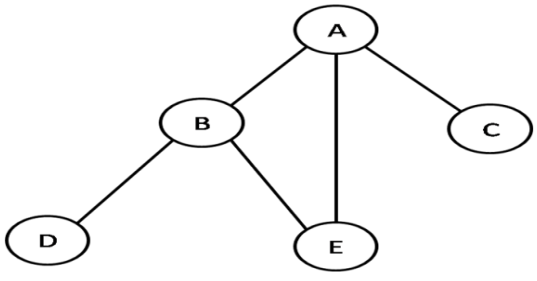


WEEK-8	GRAPH TRAVERSALS

a. Print all the nodes reachable from a given starting node in a digraph using BFS method.



b. Check whether a given graph is connected or not using DFS method.



WEEK-9	SUM OF SUB SETS PROBLEM

Find a subset of a given set S = {sl, s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1, 2, 6} and

{1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.

WEEK-10	TRAVELLING SALES PERSON PROBLEM

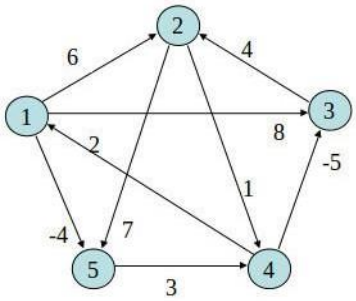
Implement any scheme to find the optimal solution for the Traveling Sales Person problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.

WEEK-11	MINIMUM COST SPANNING TREE

Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.

WEEK-12	ALL PAIRS SHORTEST PATHS

Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.



	1	2	3	4	5
1	0	6	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	3	0

WEEK-13	N QUEENS PROBLEM

Implement N Queen's problem using Back Tracking.

Reference Books:

1. Levitin A, "Introduction to the Design And Analysis of Algorithms", Pearson Education, 2008.
2. Goodrich M.T.,R Tomassia, "Algorithm Design foundations Analysis and Internet Examples", John Wileyn and Sons, 2006.
3. Base Sara, Allen Van Gelder ," Computer Algorithms Introduction to Design and Analysis", Pearson, 3rd Edition, 1999.

Web References:

1. http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html
2. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms
3. http://www.facweb.iitkgp.ernet.in/~sourav/daa.html

XVI. COURSE PLAN:

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

Week No	Topics to be covered	Course Learning Outcomes (CLOs)	Reference
1	Basics of Python	CLO 1	R1:15.1
2	Quick Sort, Merge Sort	CLO 2	T1:5.1
3	Warshall's Algorithm	CLO 3	T1:5.2 R2 : 10.2
4	Knapsack Problem	CLO 4	T1:7.1 T1:8.1
5	Shortest Paths using Dijkstra's Algorithm	CLO 5	T2:26.8
6	Minimum Cost Spanning Tree using Kruskal's algorithm	CLO 6	T1:9.2
7	Tree Traversals	CLO 7	T2:26.14 R2:21.55
8	Graph Traversals	CLO 8	T1:7.2
9	Sum of Sub Sets Problem	CLO 9	T1:7.2 R2:21.61
10	Travelling Sales Person Problem	CLO 10	T2:25.12 R2:21.24
11	Minimum Cost Spanning Tree using Prim's algorithm	CLO 11	T2:25.16 R2:21.29
12	Floyd's algorithm and N-Queens problem	CLO 12	T1:8.1

XVII. 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	PO 5	PSO 1
2	Substitution Method	Extra Lab Sessions.	PO 2	PSO 2

Prepared by:
E Uma Shankari, Assistant Professor

HOD, CSE