INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE DESCRIPTOR

Course Title	**DESIGN AND ANALYSIS OF ALGORITHMS**				
Course Code	AIT001				
Programme	B.Tech				
Semester	III	CSE			
Course Type	Core				
Regulation	IARE - R16				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3	-	3	2	1
Chief Coordinator	Dr. K Rajendra Prasad, HOD,CSE				
Course Faculty	Mr. C Raghavendra, Assistant Professor Ms. G Vasavi Assistant Professor Mr. S.Laxman Kumar, Assistant Professor Mr. Ch.Suresh, Assistant Professor				

I. COURSE OVERVIEW:

The primary objective of this course is to introduce the concept of algorithm as a precise mathematical concept, and study how to design algorithms, establish their correctness, study their efficiency and memory needs. The course consists of a strong mathematical component in addition to the design of various algorithms.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites
UG	ACS001	I	Computer Programming
UG	ACS002	II	Data Structures

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Design And Analysis Of Algorithms	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 units 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 module. 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 20 marks for Continuous Internal Examination (CIE), 05 marks for Quiz and 05 marks for Alternative Assessment Tool (AAT).

Table 1: Assessment pattern for CIA

Component	**Theory**			**Total Marks**
Type of Assessment	CIE Exam	Quiz	AAT	
CIA Marks	20	05	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 20 marks of 2 hours duration consisting of five descriptive type questions out of which four 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 - Online Examination
Two Quiz exams shall be online examination consisting of 25 multiple choice questions and are to be answered by choosing the correct answer from a given set of choices (commonly four). Such a question paper shall be useful in testing of knowledge, skills, application, analysis, evaluation and understanding of the students. Marks shall be awarded considering the average of two quiz examinations for every course.

Alternative Assessment Tool (AAT)
This AAT enables faculty to design own assessment patterns during the CIA. The AAT converts the classroom into an effective learning centre. The AAT may include tutorial hours/classes, seminars, assignments, term paper, open ended experiments, METE (Modeling and Experimental Tools in Engineering), five minutes video, MOOCs etc.
The AAT chosen for this course is given in section XI.

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.	2	Assignments, Tutorials
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	2	Assignments
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	Mini Projects
PO 4	**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.	2	Projects

3 = High; 2 = Medium; 1 = Low

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Strength	Proficiency assessed by
PSO 1	**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.	2	Lectures, Assignments
PSO 2	**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.	2	Projects
PSO 3	**Successful career and entrepreneurship:** 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	2	Guest Lectures

3 = High; 2 = Medium; 1 = Low

VIII. COURSE OBJECTIVES (COs):

The course should enable the students to:	
I	Calculate performance of algorithms with respect to time and space complexity.
II	Illustrate the graph traversals and tree traversals to solve the problems
III	Demonstrate the concepts greedy method and dynamic programming for several applications like knapsack problem, job sequencing with deadlines, and optimal binary search tree, TSP.
IV	Illustrating the methods of backtracking and branch bound techniques to solve the problems like n-queens problem, graph colouring and TSP respectively

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
AIT001.01	CLO 1	Use big O-notation formally to give asymptotic upper bounds on time and space complexity of algorithms	PO 1, PO 2	2
AIT001.02	CLO 2	Explain the use of big-Omega, big-Theta, and little-o notations to describe the amount of work done by an algorithm.	PO 1, PO 2	2
AIT001.03	CLO 3	Use recurrence relations to determine the time complexity of recursive algorithms.	PO 2	2
AIT001.04	CLO 4	Evaluate and compare different algorithms using worst, average, and best-case analysis	PO 3	2
AIT001.05	CLO 5	Solve elementary recurrence relations, e.g., using some forms of a Master Theorem. Give examples that illustrate time-space trade-offs of algorithms.	PO 2, PO 3	3
AIT001.06	CLO 6	Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and explain an implementation of the algorithm in a particular context.	PO 2, PO 12	3
AIT001.07	CLO 7	Describe and use major algorithmic techniques (brute-force, greedy, divide-and-conquer, dynamic programming, and graph explorations).	PO 3, PO 4	3
AIT001.08	CLO 8	Use a divide-and-conquer algorithm to solve an appropriate problem	PO 2, PO 4	2
AIT001.09	CLO 9	Use a greedy approach to solve an appropriate problem and determine if the greedy rule chosen leads to an optimal solution.	PO 2	3
AIT001.10	CLO 10	Use dynamic programming to develop the recurrence relations and to solve an appropriate problem.	PO 4	3
AIT001.11	CLO 11	Use recursive backtracking to solve a problem such as navigating a maze	PO 1, PO 2, PO 4	3
AIT001.12	CLO 12	Explain the major graph algorithms and their analysis and employ graphs to model application problems.	PO 5,PO 11	3
AIT001.13	CLO 13	Determine appropriate algorithmic approaches to apply to a given problem.	PO 1	3
AIT001.14	CLO 14	Describe heuristic problem-solving methods.	PO 5, PO 9, PO 11	3
AIT001.15	CLO 15	Understand the mapping of real-world problems to algorithmic solutions	PO 2, PO 3, PO 9	3
AIT001.16	CLO 16	Define the classes P and NP.	PO 2	3
AIT001.17	CLO 17	Explain the significance of NP-completeness.	PO 1, PO 2	3
AIT001.18	CLO 18	Provide examples of NP-complete problems	PO 2	2
AIT001.19	CLO 19	Explain the impact of NP-complete problems to different application domains.	PO 1	2
AIT001.20	CLO 20	Explain the difference between NP-complete and NP-hard.	PO 4	2
AIT001.21	CLO 21	Prove that a problem is NP-complete.	PO 2	2

CLO Code	CLO's	At the end of the course, the student will have the ability to:	PO's Mapped	Strength of Mapping
AIT001.22	CLO 22	Use reduction techniques between problems.	PO 1, PO 4	2
AIT001.23	CLO 23	Demonstrate the use of approximation algorithms for NP-hard problems	PO 1	2
AIT001.24	CLO 24	Explain the Halting problem and other un-decidable problems.	PO 2	3
AIT001.25	CLO 25	Possess the knowledge and skills for employability and to succeed in national and international level competitive examinations	PO 12	2

3 = High; 2 = Medium; 1 = Low

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

Course 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	2											2		
CLO 2	3	2											2		
CLO 3		3												2	
CLO 4			2										3		
CLO 5		2	2										2		
CLO 6		2										2	2		
CLO 7			2	2										2	
CLO 8		3		2									2		
CLO 9		3													
CLO 10				2										3	
CLO 11	2	2		3										2	
CLO 12					2							2	2		
CLO 13	2													2	
CLO 14					2				2			2		2	
CLO 15		2	3						2				2	3	
CLO 16		2												2	

Course 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 17	2	2												2	
CLO18		3												2	
CLO19	2												2		
CLO20				2										2	
CLO21		2													
CLO22	2			2											
CLO23	2														
CLO24		3												3	
CLO25												2			2

3 = High; 2 = Medium; 1 = Low

XI. ASSESSMENT METHODOLOGIES – DIRECT

CIE Exams	PO 3	SEE Exams	PO 2	Assignments	PO 1	Seminars	PO 6
Laboratory Practices	PO 2	Student Viva	PO 2	Mini Project	PO 11	Certification	-
Term Paper	PO 6						

XII. ASSESSMENT METHODOLOGIES - INDIRECT

✔	Assessment of course outcomes (by feedback, once)	✔	Student feedback on faculty (twice)
✔	Assessment of mini projects by experts		

XIII. SYLLABUS:

UNIT-I	INTRODUCTION
Introduction: Algorithm, Pseudo code for expressing algorithms, Performance Analysis-Space complexity, Time complexity, Asymptotic Notation-Big oh notation, Omega notation, Theta notation and Little oh notation, Probabilistic analysis, Amortized complexity Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.	
UNIT-II	**SEARCHING AND TRAVERSAL TECHNIQUES**
Searching and traversal techniques, Disjoint set operations, union and find algorithms, Efficient non-recursive binary tree traversal algorithms, spanning trees, Graph traversals-Breadth first search, Depth first search, connected components, Bi-connected components.	
UNIT-III	**GREEDY METHOD AND DYNAMIC PROGRAMMING**

General method, applications-Job sequencing with deadlines, 0/1 knapsack problem, Minimum cost spanning trees, Single source shortest path problem. Dynamic Programming: General method, applications-Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Single source shortest path problem, Travelling sales person problem.	
UNIT-IV	**BACKTRACKING AND BRANCH AND BOUND**
Backtracking: General method, applications-8-queens problem, sum of subsets problem, graph coloring, Hamiltonian cycles. Branch and Bound: General method, applications-0/1 knapsack problem-LC Branch and Bound solution, FIFO Branch and Bound solution, Travelling sales person problem.	
UNIT-V	**NP-HARD AND NP-COMPLETE PROBLEMS**
NP-Hard and NP-Complete problems: Basic concepts, Non-deterministic algorithms, the classes NP -Hard and NP, NP Hard problems, clique decision problem, chromatic number decision problem, Cook's theorem.	
Text Books:	
1. Horowitz, Satraj Sahni, Sanguthevar Rajasekharan, "Fundamentals of Computer Algorithms", Universities Press, 2nd Edition, 2015. 2. Ellis 2. Alfred V. Aho, John E. Hop croft, Jeffrey D, "The Design And Analysis Of Computer Algorithms", Pearson India, 1st Edition, 2013.	
Reference Books:	
1. Levi tin A, "Introduction to the Design and Analysis of Algorithms", Pearson Education, 3rdEdition, 2012. 2. Goodrich, M. T. R Tamassia, "Algorithm Design Foundations Analysis and Internet Examples", John Wiley and Sons, 1st Edition, 2001. 3. Base Sara Allen Vangelder, "Computer Algorithms Introduction to Design and Analysis", Pearson Education, 3rdEdition ,1999	

XIV. COURSE PLAN:

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

Lecture No	Topic's to be covered	Course Learning Outcomes (CLOs)	Reference
1	Algorithm, Pseudo code for expressing algorithm	CLO 4	T1:1.1-1.2
2-5	Space complexity, time complexity Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation	CLO 1	T1:1.3
6-7	Probabilistic analysis, Amortized complexity, Divide and conquer – general method applications, Binary search, Quick sort,Merge sort, Strassen's matrix multiplication	CLO 6	T1: 1.4
8-15	non-recursive binary tree traversal, Disjoint set operations, union and find algorithms	CLO 7	T1:3.1,3.3 T1:3.5,3.6 T1:3.8
16-17	spanning trees, Graph traversals-Breadth first search, Depth first search, AND/OR graphs, Game trees	CLO 3	T1 :6.1
18-19	Connected components, Bi-connected components	CLO	T1: 2.5
20-22	Greedy Method: applications Job sequencing with deadlines, 0/1 knapsack problem Minimum cost spanning trees, Single source shortest path problem	CLO 12	T1: 6.2
23-24	Dynamic Programming: General method, applications Optimal binary search trees, matrix chain multiplication, 0/1 knapsack problem, All pairs shortest path problem, Single source shortest path problem , Travelling sales person problem	CLO 12	T1:11.3

Lecture No	Topic's to be covered	Course Learning Outcomes (CLOs)	Reference
25-26	Backtracking: General method, applications n-queen problem, sum of subsets problem graph coloring, Hamiltonian cycles	CLO	T1:6.3,6.4
27-32	Branch and Bound: General method, applications : Travelling sales person problem0/1 knapsack problem- LC Branch and Bound solution FIFO Branch and Bound solution	CLO 9	T1: 4.1 T1:4.3,4.5 T1:4.6,4.9
33-43	NP-Hard and NP-Complete problems Basic concepts	CLO 10	T1:5.15.3 T1:5.5 T1:5.7-5.9
44-51	Distinguish non-deterministic algorithms, NP - Hard and NP Complete classes NP Hard problems, Cook's theorem	CLO 11 CLO 24	T1: 7.17.5 T1:11.2

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

S No	Description	Proposed actions	Relevance with POs	Relevance with PSOs
1	Skip lists, Problem reductions, Polynomial time and intractability	Seminars / Guest Lectures/ NPTEL	PO 1, PO 2, PO 3	PSO 1
2	String matching: Knuth-Morris-Pratt, Boyer-Moore, Edit distance, Longest increasing subsequence, Smith-Waterman algorithm	Seminars / Guest Lectures/ NPTEL	PO 2, PO 5	PSO2
3	Encourage students to write programs based on the taught algorithms to solve problems	Assignments / Laboratory Practices	PO 1,PO 3, PO 4	PSO 2

Prepared by:
Dr. K Rajendra Prasad, Professor & Head. **HOD, CSE**
Dr. R.Obulakonda Reddy, Professor