INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE INFORMATION SHEET

Course Title	**DATA STRUCTURES AND PROBLEM SOLVING**			
Course Code	**BCS002**			
Programme	**M.Tech**			
Semester	**I**			
Course Type	**Core**			
Regulation	**IARE - R16**			
Course Structure	**Lectures**	**Tutorials**	**Practicals**	**Credits**
	4	**-**	**-**	**4**
Course Coordinator	**Mr. Rajasekhar Nennuri , Assistant Professor** **Department of Computer Science and Engineering**			

I. COURSE OVERVIEW:

The course covers the concepts of programming and demonstrates fundamental programming techniques, customs and terms including the library functions and the usage of the preprocessor. This course helps the students in gaining the knowledge to write C language applications, mathematical and engineering problems. This course helps to undertake future courses that assume this programming language as a background in C and Data Structures. Topics include variables, data types, functions, control structures, pointers, strings, arrays and dynamic allocation principles. 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
PG	BCS002	I	C Programming	4

III. MARKS DISTRIBUTION

Subject	SEE Examination	CIA Examination	Total Marks
Data Structures and Problem Solving	70 Marks	30 Marks	100

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 weight age 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.

Continuous Internal Assessment (CIA):

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

Continuous Internal Examination (CIE):

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 20 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, micro projects, five minutes video and MOOCs.

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

√	CHALK & TALK	√	QUIZ	√	ASSIGNMENTS	√	MOOCs
√	LCD / PPT	√	SEMINARS	√	MINI PROJECT	√	VIDEOS
√	OPEN ENDED EXPERIMENTS						

V. ASSESSMENT METHODOLOGIES – DIRECT

√	CIE EXAMS	√	SEE EXAMS	√	ASSIGNEMNTS	√	SEMINARS
√	STUDENT VIVA	√	MINI PROJECT	X	CERTIFICATION	√	TERM PAPER

VI. ASSESSMENT METHODOLOGIES – INDIRECT

√	ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	√	STUDENT FEEDBACK ON FACULTY (TWICE)
√	ASSESSMENT OF MINI PROJECTS BY EXPERTS		

VII. COURSE OBJECTIVES:

The course should enable the students to:

I. Learn the basic techniques of algorithm analysis.
II. Demonstrate several searching and sorting algorithms.
III. Implementation of linear data structure mechanisms.
IV. Demonstrate various tree and graph traversal algorithms.
V. Analyze and choose appropriate data structure to solve problems in real world.

VIII . COURSE LEARNING OUTCOMES:

Students, who complete the course, will have demonstrated the ability to do the following:

BCS002.01	Analyze time and space complexity of an algorithm for their performance analysis
BCS002.02	Understand arrays, single and doubly linked lists in linear data structure and tress , graphs in non-linear data structure
BCS002.03	Master a variety of advanced abstract data type (ADT) and their implementations
BCS002.04	Understand dynamic data structures and relevant standard algorithms
BCS002.05	Design and analyze and Concepts of heap, priority queue
BCS002.06	Design more complex variants in linear linked list representation
BCS002.07	Understand and implement hash table representation
BCS002.08	Understand the properties of binary tress and implement recursive and non-recursive traversals
BCS002.09	Understand graphs terminology, representations and traversals in Graphs
BCS002.10	Implement Depth First Search and Breath First Searching methods of non –linear data structures
BCS002.11	Analyze dijkstra's algorithm for single source shortest path problem for minimum cost spanning trees
BCS002.12	Implement binary search ADT for finding parent node, smallest and largest values in binary search

BCS002.13	Understand and implement operations and applications of red-Black and splay Trees
BCS002.14	Implement Huffman Coding and decoding for text compression

IX. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		Level	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.	H	Assignments
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.	H	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.	S	Mini Project
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.	S	Open ended experiments /
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.	S	Mini Project
PO 6	**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	---
PO 7	**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	---.
PO 8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	N	---
PO 9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	N	---
PO 10	**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.	S	Seminars / Term Paper / 5 minutes video
PO 11	**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	---
PO 12	**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	----

N= None **S= Supportive** **H = Highly Related**

X. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Level	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.	H	Lectures, Assignments
PSO 2	**Problem-Solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.	H	Projects
PSO 3	**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

XI. SYLLABUS:

UNIT I
Algorithm analysis: Algorithms; Performance analysis: Time complexity and space complexity, asymptotic notation: Big Oh, omega and theta notations, complexity analysis examples; Data structures: Linear and non linear data structures, ADT concept, linear list ADT, stack and queue ADTs, array and linked list representations; Circular queue: Insertion and deletion, de queue ADT, priority queue ADT, implementation using heaps, insertion into a max heap, deletion from a max heap, singly linked lists, doubly linked lists, circular linked list.
UNIT II
Dictionaries: Linear list representation, operations insertion, deletion and searching, hash table representation, hash functions, collision resolution, separate chaining, open addressing, linear probing, quadratic probing, double hashing, rehashing, extendible hashing .
UNIT III
Trees: Ordinary and binary trees terminology, properties of binary trees, binary tree ADT, representations, recursive and non recursive traversals, threaded binary trees. Graphs: Graphs terminology, graph ADT, representations, graph traversals; Search methods: DFS and BFS; Applications of Graphs: Minimum cost spanning tree using Kruskal's algorithm, Dijkstra's algorithm for single source shortest path problem. .
UNIT IV
Binary search tree: Binary search tree ADT, insertion, deletion and searching operations, finding the parent of a given node, attaining a reference to a node, finding the smallest and largest values in the binary search tree; Balanced search trees: AVL trees, definition, height of an AVL tree; Operations : Insertion, deletion and searching.
UNIT V
Red-Black and Splay Trees; B trees: Definition, operations and applications; R trees: Nearest neighbor query, join and range queries; Comparison of search trees; Text compression: Huffman coding and decoding; Pattern matching: KMP algorithm.

TEXT BOOKS:

1.	Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Universities Press Private Limited, India, 2nd Edition, 2008.
2.	G. A. V. Pai, "Data Structures and Algorithms", Tata Mc Graw Hill, New Delhi, 1st Edition, 2008.
3	M. A. Weiss, Addison Wesley, "Data Structures and Algorithm Analysis in Java", Pearson Education, 2nd Edition, 2005.

REFERENCES:

1.	D. Samanta, "Classic Data Structures", Prentice Hall of India Private Limited, 2nd Edition, 2003.
2.	Aho, Hopcraft, Ullman, "Design and Analysis of Computer Algorithms", Pearson Education India, 1st Edition, 1998.

XII. COURSE PLAN:

The course plan is meant as a guideline. There may probably be changes.

Lecture No	Topic Outcomes	Topic/s to be covered	Reference
1-2	Analyze time and space complexity of an algorithm for their performance analysis	Algorithm analysis: Algorithms; Performance analysis: Time complexity and space complexity.	T1: 1.1-1.5
2-5	Analyze time and space complexity of an algorithm for their performance analysis	asymptotic notation: Big Oh, omega and theta notations, complexity analysis examples.	T1: 2.1-2.8
6-10	Understand arrays, single and doubly linked lists in linear data structure and tress , graphs in non-linear data structure	Data structures: Linear and non linear data structures, ADT concept, linear list ADT, stack and queue ADTs, array and linked list representations; Circular queue: Insertion and deletion, de queue ADT, priority queue ADT.	T1: 3.1-3.6
11-12	Master a variety of advanced abstract data type (ADT) and their implementations	Implementation using heaps, insertion into a max heap, deletion from a max heap, singly linked lists, doubly linked lists, circular linked list.	T1: 5.1-5.3
13-15	Understand dynamic data structures and relevant standard algorithms	Dictionaries: Linear list representation, operations insertion, deletion and searching, hash table representation.	T1: 5.4-5.7
16-19	Understand dynamic data structures and relevant standard algorithms	hash functions, collision resolution, separate chaining, open addressing, linear probing, quadratic probing, double hashing, rehashing, extendible hashing.	T2:5.1-5.5
20-22	Understand the properties of binary tress and implement recursive and non-recursive traversals	Trees: Ordinary and binary trees terminology, properties of binary trees, binary tree ADT.	T2:10.2.3
23-26	Understand the properties of binary tress and implement recursive and non-recursive traversals	Recursive and non recursive traversals, threaded binary trees.	T1:8.1-8.4
24-28	Understand graphs terminology, representations and traversals in Graphs	Graphs: Graphs terminology, graph ADT, representations, graph traversals; Search methods: DFS and BFS; Applications of Graphs.	T2:9.1-9.6
29-35	Understand graphs terminology, representations and traversals in Graphs	Applications of Graphs: Minimum cost spanning tree using Kruskal‟s algorithm, Dijkstra‟s algorithm for single source shortest path problem.	T1:11.1-11.4

Lecture No	Topic Outcomes	Topic/s to be covered	Reference
33-40	Understand the properties of binary tress and implement recursive and non-recursive traversals	Binary search tree: Binary search tree ADT, insertion, deletion and searching operations, finding the parent of a given node, attaining a reference to a node, finding the smallest and largest values in the binary search tree.	T1:20.1-20.7
41-45	Understand the properties of binary tress and implement recursive and non-recursive traversals	Balanced search trees: AVL trees, definition, height of an AVL tree; Operations : Insertion, deletion and searching.	T1:20.8-20.9
47-55	Understand and implement operations and applications of red-Black and splay Trees	Red-Black and Splay Trees; B trees: Definition, operations and applications; R trees: Nearest neighbor query, join and range queries; Comparison of search trees.	T3:25.1-20.3
56-60	Implement Huffman Coding and decoding for text compression	Text compression: Huffman coding and decoding; Pattern matching: KMP algorithm.	T3:28.1-28.7

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

S NO	DESCRIPTION	PROPOSED ACTIONS	RELEVANCE WITH POs	RELEVANCE WITH PSOs
1	Sorting Algorithms	Seminars / Guest Lectures / NPTEL	PO 1, PO 2, PO 3	PSO 1, PSO 2
2	Binary Tree Traversals	Seminars / Guest Lectures / NPTEL	PO 2, PO 3	PSO 1
3	Hash Function	Assignments / Laboratory Practices	PO 1, PO 3, PO 4	PSO 2

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

Course Objectives (COs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
I	H	S	S	S									S	H	
II	S	S	S	H	S								H		S
III	H	H	S	S							S	S		H	
IV	H	S	H	S	S						S	S		H	
V	S											H	H		S

S= Supportive **H = Highly Related**

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

Course Learning Outcomes (CLOs)	Program Outcomes (COs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CACS004.01	H	S												S	
CACS004.02	S	H	S											S	
CACS004.03	S	H		S	S								S		
CACS004.04	H	S	H	S								S		H	
CACS004.05	S	H	S	S								S		H	
CACS004.06		H	S									S	S		
CACS004.07				H								S	S		
CACS004.08	S	H			S									S	
CACS004.09		S	H											S	
CACS004.10		H										S		S	
CACS004.11	S	H	S									S	S	S	S
CACS004.12		S		H									S	S	
CACS004.13	S	S	S										S	S	
CACS004.14		H		S	S							S		H	
CACS004.15	S	S											S	S	
CACS004.16	S	H		S	S							S		H	
CACS004.17	H			S								S		S	
CACS004.18	H			S								S	S	H	
CACS004.19	H	S			S									H	
CACS004.20									S		S	H	S		S

S= Supportive **H = Highly Related**

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

I. Rearrange the following numbers 42, 12, 18, 98, 67,83,8,10, 71 using sorting algorithms.

II. Given In order traversal of a binary tree is D,G,B,E,A,H,F,I,C and pre order traversal is A,B,D,G,E,C,F,H,I construct binary tree.

III. Analyze input (371, 323, 173, 199, 344, 679, 989) and hash function h(x)=x mod 10, Show the result Separate Chaining, linear probing.

Prepared by
Mr. Rajasekhar N, Assistant Professor, C.S.E

HOD, CSE