INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

MECHANICAL ENGINEERING

COURSE DESCRIPTOR

Course Title	**DATA STRUCTURES**				
Course Code	ACSB03				
Program	B.Tech				
Semester	THIRD				
Course Type	Core				
Regulation	IARE - R18				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3	0	3	3	1.5
Course Coordinator	Ms. K Laxminarayanamma, Assistant Professor				

I. COURSE OVERVIEW:

The course covers some of the general-purpose data structures and algorithms, and software development. Topics covered include managing complexity, analysis, static data structures, dynamic data structures 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 reaches to student by power point presentations, lecture notes, and lab which involve the problem solving in mathematical and engineering areas.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites
B.Tech	ACSB01	II	Programming for Problem Solving

III. MARKS DISTRIBUTION:

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

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✔	PPT	✔	Chalk & Talk	✔	Assignments	✗	MOOCs
✔	Open Ended Experiments	✔	Seminars	✔	Mini Project	✔	Videos
✔	Others: Quiz						

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 modules and each module 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 expected percentage of cognitive level of the questions is broadly based on the criteria given in Table: 1.

Table 1: The expected percentage of cognitive level of questions in SEE

Percentage of Cognitive Level	Blooms Taxonomy Level
0 %	Remember
50 %	Understand
40 %	Apply
10 %	Analyze
0 %	Evaluate
0 %	Create

Continuous Internal Assessment (CIA):

CIA is conducted for a total of 30 marks (Table 2), with 20 marks for Continuous Internal Examination (CIE), 05 marks for Quiz and 05 marks for Alternative Assessment Tool (Table 3).

Table 2: 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 center. 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 table 3**.**

Table 3: Assessment pattern for AAT

5 Minutes Video	Assignment	Tech-talk	Seminar	Open Ended Experiment
20%	30%	30%	10%	10%

VI. COURSE OBJECTIVES:

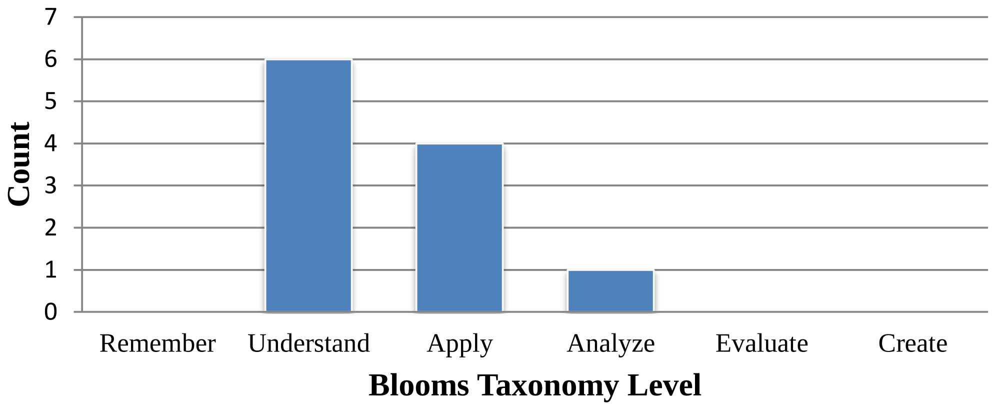
The students will try to learn:	
I	To provide students with skills needed to understand and analyze performance trade-offs of different algorithms / implementations and asymptotic analysis of their running time and memory usage.
II	To provide knowledge of basic abstract data types (ADT) and associated algorithms: stacks, queues, lists, tree, graphs, hashing and sorting, selection and searching.
III	The fundamentals of how to store, retrieve, and process data efficiently.
IV	To provide practice by specifying and implementing these data structures and algorithms in Python.
V	Understand essential for future programming and software engineering courses.

VII. COURSE OUTCOMES:

After successful completion of the course, students will be able to:		
Course Outcomes		**Knowledge Level (Bloom's Taxonomy)**
CO 1	**Carryout** the analysis of a range of algorithms in terms of algorithm analysis and express algorithm complexity using the O notation.	Understand
CO 2	**Make use** of recursive algorithm design technique in appropriate contexts.	Apply

CO 3	**Represent** standard ADTs by means of appropriate data structures.	Understand
CO 4	**Select** appropriate sorting technique for given problem.	Understand
CO 5	**Select** appropriate searching technique for given problem.	Understand
CO 6	**Implement** standard searching and sorting algorithms; including binary search; merge sort and quick sort; and their complexities.	Apply
CO 7	Design and **implement** linked lists, stacks and queues in Python.	Apply
CO 8	**Explain** the use of basic data structures such as arrays, stacks, queues and linked lists in program design.	Understand
CO 9	Extend their knowledge of data structures to more sophisticated data structures to solve problems involving balanced binary search trees, AVL Trees, B-trees and B+ trees, hashing, and basic graphs.	Apply
CO 10	**Design** and implement tree structures in Python.	Apply
CO 11	**Compare** and contrast the benefits of dynamic and static data structures implementations and choose appropriate data structure for specified problem domain.	Understand
CO 12	Quickly **determine and explain** how efficient an algorithm or data structure will be, apply appropriate data structures for solving computing problems with respect to performance.	Analyze

COURSE KNOWLEDGE COMPETENCY LEVELS



VIII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		Strength	Proficiency Assessed by
PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineeringfundamentals, and an engineering specialization to the solution of complex engineering problems.	3	CIE/Quiz/AAT
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complexengineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences	2	CIE/Quiz/AAT

Program Outcomes		Strength	Proficiency Assessed by
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	Seminar/ Conferences/ Workshops
PO 5	**Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modelling to complex Engineering activities with an understanding of the limitations.	2	Assignments/ Discussion
PO 9	**Individual and Teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings	2	Class group/ Multi-disciplinary group
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.	2	Discussion on Innovations/ Presentation
PO 12	**Life-Long Learning:** Recognize the need for and having the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	1	Seminars/ Workshops/ Short term courses

3 = High; 2 = Medium; 1 = Low

IX. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Strength	Proficiency assessed by
PSO 1	Formulate and evaluate engineering concepts of design, thermal and production to provide solutions for technology aspects in digital manufacturing.	1	Mini Projects/ Seminar/ Assignments
PSO 3	Make use of computational and experimental tools for creating innovative career paths, to be an entrepreneur and desire for higher studies.	2	Discussion on Innovations/ Industry exposure

3 = High; 2 = Medium; 1 = Low

X. MAPPING OF EACH CO WITH PO(s), PSO(s):

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO 1	√	-	-	-	-	-	-	-	-	-	-	-	√	-	-
CO 2	√	-	-	-	-	-	-	-	-	-	-	-	√	-	-
CO 3	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 4	√	-	-	-	-	-	-	-	-	√	-	-	√	-	-

CO 5	√	-	-	-	-	-	-	-	-	√	-	-	√	-	-
CO 6	√	-	-	-	-	-	-	-	-	√	-	-	√	-	-
CO 7	√	√	√	-	-	-	-	-	-	√	-	-	√	-	√
CO 8	√	√	√	-	-	-	-	-	-	√	-	-	√	-	√
CO 9	√	√	√	-	-	-	-	-	-	√	-	-	√	-	√
CO 10	√	√	√	-	-	-	-	-	-	√	-	-	√	-	√
CO 11	√	√	√	-	-	-	-	-	-	√	-	-	√	-	√
CO 12	√	√	√	-	-	-	-	-	-	√	-	-	√	-	√

XI. JUSTIFICATIONS FOR CO – (PO, PSO) MAPPING –DIRECT

Course Outcomes	POs / PSOs	Justification for mapping (Students will be able to)	No. of key competencies
CO 1	**PO 1**	Understand (knowledge) the basic concept of algorithm analysis which provides theoretical estimates for the resources needed by any algorithm for a given computational problem. These estimates provide an insight into reasonable directions of search for efficient algorithms by applying the principles of **mathematics and science.**	3
	PSO 1	Understand, design and analyze computer programs in the areas related to Algorithms and System Software.	1
CO 2	**PO 1**	Describe (knowledge) to simplify an algorithm easily understood by most people. To solve a problem using recursion, the solution depends on solutions to smaller instances of the same problem **using principles of mathematics, science, and engineering fundamentals.**	3
	PSO 1	Make use of recursive approaches is crucial to the process of creating efficient and well-structured solutions for **design and development of real-time computational problems.**	2
CO 3	**PO 1**	Focus on (knowledge) data abstraction or abstract data type as a logical description of viewing the data and the operations without regard to how they will be implemented using the **fundamentals of mathematics, science, and engineering.**	2
CO 4	**PO 1**	Describe (knowledge) the use sorting techniques as a basic building block in algorithm design and problem solving **using principles of mathematics, science, and engineering fundamentals.**	3
	PO 10	Recognize the importance of efficient sorting techniques for optimizing the efficiency of other algorithms that require input data to be in sorted by **communicating effectively with engineering community.**	4
	PSO 1	Focus on simple sorting algorithms which are at the heart of solving important problems in computational geometry, machine learning, Big data and AI by **understanding and applying the engineering principles.**	2

CO 5	**PO 1**	Outline the importance of searching algorithms to retrieve an element from any data structure where it is stored by **understanding and applying the fundamentals of mathematics, science and engineering.**	3
	PO 10	Understand the use of searching techniques that retrieve information stored within some data structure by **communicating effectively with engineering community.**	3
	PSO 1	Extend the focus to understand searching techniques in solving many Web design, Machine Learning and Artificial Intelligence (AI) problems by **understanding and applying the engineering principles.**	2
CO 6	**PO 1**	Make use of broad knowledge of searching and sorting techniques to efficiently search an item from a data structure and optimize the efficiency of other algorithms by **applying the knowledge of mathematics, science, Engineering fundamentals.**	3
	PO 10	Build strong foundation of Asymptotic analysis of algorithms and searching and sorting applications for career building by **communicating effectively with engineering community.**	3
	PSO 1	Acquire sufficient knowledge to understand the importance of algorithmic analysis in real-time problem solving in various domains like Big-data, Cloud computing, Web design etc. by **understanding and applying the engineering principles.**	2
CO 7	**PO 1**	Make use of linear data structures to organize the data in a particular way so to use them in the most effective way by applying the **basic knowledge of mathematics, science, engineering fundamentals**.	3
	PO 2	Build strong foundation of data Structures which tells the program how to store data in memory and forming some relations among the data and use them in **design and development of new products.**	7
	PO3	Recognize the need of linear data structures such as linked list, array, stack and queue by **designing solutions for complex Engineering problems** in real-time.	7
	PSO 1	Extend the focus to understand the linear data structures as the key building blocks of many important algorithms used in the areas related to **System Software, Web design, Big data, Artificial Intelligence, Machine Learning and Networking.**	2
	PSO 3	Acquire sufficient knowledge to develop real-time applications by making use of linear data structures in **career building and higher studies**.	2
CO 8	**PO 1**	Describe (knowledge) the usage of data structures in organizing, managing, and storing different data formats that enables efficient access and modification by applying the **fundamentals of mathematics, science, and engineering.**	3
	PO 2	Recognizing (knowledge) the importance of basic data structures that are suitable for different kinds of applications by **analyzing and formulating solutions of complex engineering problems.**	7
	PO 3	Understand the applications of basic data structures such as stacks, queues, linked lists in **designing and developing solutions of complex engineering applications.**	6

	PSO 1	Make use of basic data structures to **understand, design and analyze computer programs** in reducing time and space complexities of various applications.	2
	PSO 3	**Make use of modern computer tools for** applying the basic data structure concepts in building real-time applications for a successful career.	1
CO 9	**PO 1**	Apply the sophisticated hierarchical data structures to organize keys in form of a tree to use in many real-life applications by **using the principles of mathematics and engineering fundamentals.**	2
	PO 2	Make use of non-linear data structures such as balanced trees in **by identifying, formulating and analyzing complex engineering problems** such as databases, syntax tree in compilers and domain name servers etc. **with the help of basic mathematics and engineering sciences.**	7
	PO 3	Extend the concept of tree data structures to **design and develop solutions for complex engineering problems**.	6
	PSO 1	Extend the focus to understand data structures and algorithms that play a major role in implementing software in **Big data, Machine Learning, Networking and Operating Systems**.	2
	PSO 3	Make use of modern computer tools in implementing non-linear data structures for various applications to become a **successful professional in the domain**.	1
CO 10	**PO 1**	Demonstrate different tree structures in Python to implement real-time problems by **applying basic knowledge of science and engineering fundamentals.**	3
	PO 2	Illustrate the importance of tree data structures used for various applications **by identifying, formulating and analyzing complex engineering problems** such as operating systems and compiler design**.**	7
	PO 3	Make use of tree data structures to **design and develop solutions for complex engineering problems** and which is the key organizing factor in software design. Data structures can be used to organize the storage and retrieval of information stored in both main memory and secondary memory.	6
	PSO 1	**Build strong foundation** of Trees which reflect structural relationships in the data and provides flexibility in allowing data to move sub-trees around with minimum effort in various computer science applications.	2
	PSO 3	Acquire sufficient knowledge in field of data structures and its applications by using modern computer tools so that new product development can take place, which leads to become **successful entrepreneur and or to obtain higher education.**	2
CO 11	**PO 1**	Understand (knowledge) the benefits of dynamic and static data structures implementations and choose appropriate data structure for specified problem domain **using knowledge of mathematics, science and engineering fundamentals.**	3
	PO 2	Recognize the need of dynamic and static data structures **in identifying, formulating and analyzing complex engineering problems.**	6

	PO 3	Describe (knowledge) the usage of static and dynamic data structures in **designing solutions for complex Engineering problems.**	7
	PSO 1	**Understand, design and implement** computer programs using the benefits of static and dynamic data structures in the areas related to Algorithms, System Software, Web design, Big data, Artificial Intelligence, Machine Learning and Networking.	2
	PSO 3	Build sufficient knowledge of dynamic data structures by using modern tools so that new product can be developed, which leads to become **successful entrepreneur in the present market.**	2
CO 12	**PO 1**	Build strong foundation of quickly determining the efficiency of an algorithm or data structure for solving computing problems with respect to performance by **using knowledge of mathematics, science and engineering fundamentals.**	3
	PO 2	Recognize the importance of suitable data structures in checking the efficiency of algorithms used for **complex engineering problems.**	8
	PO 3	Make use of broad usage of data structures in **designing and developing of complex engineering applications.**	7
	PSO 1	Make use of data structures to **understand, and implement computer programs** in reducing time and space complexities of various applications.	2
	PSO 3	Extend the concept of data structures in solving complex engineering problems using modern engineering tools to become a **successful professional in the domain**.	2

XII. TOTAL COUNT OF KEY COMPETENCIES FOR CO – (PO, PSO) MAPPING

Course Outcomes	**Program Outcomes) / No. of Key Competencies Matched**												**PSO / No. of key competencies**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
	3	**10**	**10**	**11**	**1**	**5**	**3**	**3**	**12**	**5**	**12**	**12**	**2**	**2**	**2**
CO 1	3	-	-	-	-	-	-	-	-	-	-	-	1	-	-
CO 2	3	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO 3	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 4	3	-	-	-	-	-	-	-	-	4	-	-	2	-	-
CO 5	3	-	-	-	-	-	-	-	-	3	-	-	2	-	-
CO 6	3	-	-	-	-	-	-	-	-	3	-	-	2	-	-
CO 7	3	7	7	-	-	-	-	-	-	-	-	-	2	-	2
CO 8	3	7	6	-	-	-	-	-	-	-	-	-	2	-	1
CO 9	2	7	6	-	-	-	-	-	-	-	-	-	2	-	1
CO 10	3	7	6	-	-	-	-	-	-	-	-	-	2	-	2

CO 11	3	6	7	-	-	-	-	-	-	-	-	-	2	-	2
CO 12	3	8	7	-	-	-	-	-	-	-	-	-	2	-	2

XIII. PERCENTAGE OF KEY COMPETENCIES FOR CO – (PO, PSO):

Course Outcomes	Program Outcomes / No. of key competencies												PSOs / No. of key competencies		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
	3	10	10	11	1	5	3	3	12	5	12	12	2	1	2
CO 1	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	0.0
CO 2	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
CO 3	66.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 4	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	80.0	0.0	0.0	100.0	0.0	0.0
CO 5	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	60.0	0.0	0.0	100.0	0.0	0.0
CO 6	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	60.0	0.0	0.0	100.0	0.0	0.0
CO 7	100.0	70.0	70.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0
CO 8	100.0	70.0	60.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	50.0
CO 9	66.7	70.0	60.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	50.0
CO 10	100.0	70.0	60.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0
CO 11	100.0	60.0	70.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0
CO 12	100.0	80.0	70.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0

XIV. COURSE ARTICULATION MATRIX (PO – PSO MAPPING)

COs and POs and COs and PSOs on the scale of 0 to 3, **0** being **no correlation**, **1** being the **low correlation**, **2** being **medium correlation** and **3** being **high correlation**.

0 – $0 \leq C \leq 5\%$ –Nocorrelation;

1 – $5 < C \leq 40\%$ – Low/ Slight;

2 – $40\% < C < 60\%$ –Moderate.

3 – $60\% \leq C < 100\%$ – Substantial /High

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO 1	3	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO 2	3	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO 3	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-

CO 4	3	-	-	-	-	-	-	-	-	3	-	-	3	-	-
CO 5	3	-	-	-	-	-	-	-	-	3	-	-	3	-	-
CO 6	3	-	-	-	-	-	-	-	-	3	-	-	3	-	-
CO 7	3	3	3	-	-	-	-	-	-	-	-	-	3	-	3
CO 8	3	3	3	-	-	-	-	-	-	-	-	-	3	-	2
CO 9	3	3	3	-	-	-	-	-	-	-	-	-	3	-	2
CO 10	3	3	3	-	-	-	-	-	-	-	-	-	3	-	3
CO 11	3	3	3	-	-	-	-	-	-	-	-	-	3	-	3
CO 12	3	3	3	-	-	-	-	-	-	-	-	-	3	-	3
TOTAL	36	18	18	-	-	-	-	-	-	9	-	-	32	-	16
AVERAGE	**3.0**	**3.0**	**3.0**	**-**	**-**	**-**	**-**	**-**	**-**	**3.0**	**-**	**-**	**2.9**	**-**	**2.6**

XV. ASSESSMENT METHODOLOGY - DIRECT

CIE Exams	PO 1,PO 2, PO 3,PO 5, PO 12, PSO 1,PSO 3	SEE Exams	PO 1, PO 2, PO 3, PO 5, PO 12, PSO 1,PSO 3	Assignment	PO 1,PO 2, PO 3,PO 5, PO 12, PSO 1,PSO 3	Seminars	PO 1,PO 2, PO 3,PO 5, PO 12, PSO 1,PSO 3
Laboratory Practices	PSO 1, PSO 3	Student Viva	-	Mini Project	-	Certification	-
Term Paper							

XVI. ASSESSMENT METHODOLOGY - INDIRECT

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

XVII. SYLLABUS

MODULE-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.	
MODULE- 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).	

MODULE-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	
MODULE-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	
MODULE-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.	
Textbooks:	
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.	

XVIII. COURSE PLAN:

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

Lecture No	Topics to be Covered	CO	Reference
1 – 2	Basic concepts: Introduction to Data Structures	CO 3	T1:1.1.3 R2 : 1.2
3 – 4	Classification of data structures	CO 3	T1:1.1.3 R2 : 1.4
5 – 6	Operations on data Structures	CO 3	T1:1.2
7 – 8	Searching techniques: Linear search and binary search	CO 5, CO 6, CO 12	T1:5.1
9 – 10	Sorting techniques: Bubble sort, selection sort	CO 4, CO 6, CO 12	R1:14.5
11 – 14	Insertion sort and comparison of sorting algorithms.	CO 4, CO 6, CO 12	T1:5.2 R2 : 10.2
15 – 16	Stacks: Primitive operations, implementation of stacks using Arrays	CO 7, CO 8, CO 11, CO 12	T1:7.1
17 – 20	Applications of stacks arithmetic expression conversion and evaluation	CO 7, CO 8, CO 11, CO 12	T1:7.2
21 – 22	Queues: Primitive operations; Implementation of queues using Array	CO 7, CO 8, CO 11, CO 12	T1:8.1
23 – 24	Applications of linear queue, circular queue	CO 7, CO 8, CO 11, CO 12	T1:8.4
25 – 26	Double ended queue (deque)	CO 7, CO 8, CO 11, CO 12	R2 : 5.4
27 – 28	Linked lists: Introduction, singly linked list, representation of a linked list in memory	CO 7, CO 8, CO 11, CO 12	T1:9.1

Lecture No	Topics to be Covered	CO	Reference
29– 30	Operations on a single linked list, Applications of linked lists: Polynomial representation, Circular linked lists, doubly linked lists	CO 7, CO 8, CO 11, CO 12	T1:9.2
31 - 32	Sparse matrix manipulation	CO 7, CO 8, CO 11, CO 12	T2:9.2
33 – 35	Linked list representation and operations of Stack, Linked list representation and operations of queue	CO 7, CO 8, CO 11, CO 12	T1:9
36 – 38	Trees: Basic concept, binary tree, binary tree representation, array and linked representations	CO 8, CO 9, CO 10, CO 11, CO 12	T1:13. 1-13.2
39 – 40	Binary tree traversal, binary tree variants, application of trees	CO 8, CO 9, CO 10, CO 11, CO 12	T1:13.2.3
41 – 43	Graphs: Basic concept, graph terminology, graph Implementation	CO 8, CO 9, CO 10, CO 11, CO 12	R2 : 8.2
44 – 46	Graph traversals, Application of graphs	CO 8, CO 9, CO 10, CO 11, CO 12	T2:6.2
47 – 50	Priority Queue	CO 8, CO 9, CO 10, CO 11, CO 12	T1:6.1 T2:5.6
51 – 52	Binary search trees, properties and operations	CO 8, CO 9, CO 10, CO 11, CO 12	T1:14.1
53 – 55	Balanced search trees: AVL trees, Introduction to M-Way search trees, B trees	CO 8, CO 9, CO 10, CO 11, CO 12	T1:14.3
56 – 58	Hashing and collision: Introduction, hash tables, hash functions	CO 9, CO 12	R2 : 6.4
59 - 60	Collisions, applications of hashing	CO 9, CO 12	R2 : 6.4

Prepared by:
Ms. K Laxminarayanamma, Assistant Professor **HOD, ME**