Course: 105: Data Structures and Design and Analysis of Algorithms

Course Title	Course Code 105: Data Structures and Design and Analysis of Algorithms										
Teaching per Week											
Teaching per Week Minimum weeks per Semester 15 (Including Class work, examination, preparation, holidaysetc.) Purpose of Course This course introduces the various data structures and algorithms involving these data structures and their logical implementation. Students also will be able to understand complex data structures like arrays, stacks, lists. 2. To learn fundamental data structures like trees and their use in variou. Applications  Course Objective 1. To learn fundamental data structures like trees. 3. To learn complex data structures like trees. 3. To learn analysis of algorithms  Course Outcome Course Explain the futures of various non-primitive data structure and the applications. Train students for algorithms to create, insert, delete a traversing various data structure. Course Explain the basics of computational complexity analysis and big of the anotation that help for fundamental research in algorithm analysis. Students can select the efficient algorithm after analyzin problem and identifying it's the computing requirements. Course Course Course Students will be able to work with various searching and sort techniques and compare its computational complexity. Course Cour											
Minimum weeks per Semester   Review / Revision   June 2020											
Review/Revision Purpose of Course This course introduces the various data structures and algorithms involving these data structures and their logical implementation. Students also will be able to understand complex data structures like trees and their use in variou. Applications  Course Objective  1. To learn fundamental data structures like trees. 3. To learn complex data structures like trees. 4. To learn analysis of algorithms  Course Outcome  Course Outc											
Purpose of Course  This course introduces the various data structures and algorithms involving these data structures and their logical implementation. Students also will be able to understand complex data structures like trees and their use in various Applications  1. To learn fundamental data structures like trees. 2. To learn and compare various sorting techniques. 4. To learn and compare various sorting techniques. 4. To learn analysis of algorithms  CO1. Explain the futures of various non-primitive data structure and the applications. Train students for algorithms to create, insert, delete a traversing various data structure.  CO2. Explain the basics of computational complexity analysis and big of the condition of the computational complexity analysis. Students can select the efficient algorithm after an alyzing problem and identifying it's the computing requirements.  CO3. Explain in sight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO1 CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO3 PSO4 PSO5 PSO6 PSO7 PSO  CO3 PSO6 PSO7 PS											
these data structures and their logical implementation. Students also will be able to understand complex data structures like trees and their use in variou. Applications  Course Objective  1. To learn fundamental data structures like arrays, stacks, lists. 2. To learn and compare various sorting techniques. 4. To learn analysis of algorithms  Course Outcome  Course Outcome  Course Outcome  Course Outcome  Course Outcome  Course Outcome  Course Coutcome  Course Course Outcome  Course Cour	•										
able to understand complex data structures like trees and their use in various Applications  Course Objective  1. To learn fundamental data structures like arrays, stacks, lists. 2. To learn complex data structures like trees. 3. To learn and compare various sorting techniques. 4. To learn analysis of algorithms  Course Outcome  Course Course Outcome  Course Course Outcome  Course Course outcome, Creedy method, Backtracking & Branch & Bound Outcome, Greedy met	poss or ocaros	· · · · · · · · · · · · · · · · · · ·						_			
Course Objective  1. To learn fundamental data structures like arrays, stacks, lists. 2. To learn complex data structures like trees. 3. To learn and compare various sorting techniques. 4. To learn analysis of algorithms  Course Outcome  CO1. Explain the futures of various non-primitive data structure and the applications. Train students for algorithms to create, insert, delete a traversing various data structure.  CO2. Explain the basics of computational complexity analysis and big of theta notation that help for fundamental research in algorithm analysis. Students can select the efficient algorithm after analyzin problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO3 PSO7 PSO6  CO4 PSO7 PSO7 PSO7 PSO7  CO5 STarrays-its storage structures.  1.1 Arrays-its storage structures & operations 1.2 Stacks-operations and its applications. 1.3 Queues-Types of fueues, operations and its applications. 1.4 Linked lists-Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded stora representation of Binary Trees.  1.6 Applications of Trees - The manipulation of Arithmetic expressions,					_	•					
2. To learn complex data structures like trees. 3. To learn and compare various sorting techniques. 4. To learn and compare various sorting techniques. 4. To learn analysis of algorithms  CO1. Explain the futures of various non-primitive data structure and the applications. Train students for algorithms to create, insert, delete a traversing various data structure.  CO2. Explain the basics of computational complexity analysis and big (some theta notation that help for fundamental research in algorithms analysis. Students can select the efficient algorithm after a nalyzing problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO1 PSO7 PSO6 PSO7 PSO6  CO3 PSO7 PSO6 PSO7 PSO6  CO4 PSO7 PSO6 PSO7 PSO6  CO5 PSO7 PSO6 PSO7 PSO6 PSO7 PSO6  CO5 PSO7 PSO6 PSO7 PSO		· ·									
3. To learn and compare various sorting techniques. 4. To learn analysis of algorithms  CO1. Explain the futures of various non-primitive data structure and the applications. Train students for algorithms to create, insert, delete a traversing various data structure.  CO2. Explain the basics of computational complexity analysis and big of the tan notation that help for fundamental research in algorithms in algorithms. CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO1 PSO7 PSO6 PSO7 PSO6  CO3 PSO7 PSO6 PSO7 PSO6  CO4 PSO5 PSO6 PSO7 PSO6  CO4 PSO5 PSO6 PSO7 PSO6  CO5 PSO7 PSO6 PSO7 PSO6  CO5 PSO7 PSO6 PSO7 PSO6  CO6 PSO7 PSO6  CO7 PSO6 PSO7 PSO6  CO8 PSO7 PSO6 PSO7 PSO6  CO9 PSO7 PSO6  CO9 PSO7 PSO6 PSO7 PSO6  CO9 PSO7 PSO6  CO9 PSO7 PSO6	Course Objective	1. To	learn fun	damenta	l data str	uctures l	ike array:	s, stacks, l	ists.		
Course Outcome  Course Course Outcome  Course		2. To	learn con	nplex dat	a structu	res like t	rees.				
CO1. Explain the futures of various non-primitive data structure and the applications. Train students for algorithms to create, insert, delete a traversing various data structure.  CO2. Explain the basics of computational complexity analysis and big of theta notation that help for fundamental research in algorithmation that help for fundamental research in algorithmal dentifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO3 PSO7 PSO6  CO4 PSO7 PSO7  CO5 PSO7 PSO6  CO6 PSO7 PSO7  CO7 PSO7 PSO7  CO7 PSO7 PSO7  CO7 PSO7 PSO7  CO8 PSO7 PSO7  CO9 PSO7		3. To	learn and	l compar	e various	sorting t	echnique	es.			
applications. Train students for algorithms to create, insert, delete a traversing various data structure.  CO2. Explain the basics of computational complexity analysis and big of the the computation that help for fundamental research in algority analysis. Students can select the efficient algorithm after analyzing problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO3 PSO6 PSO7 PSO6  CO4 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO6 PSO7 PSO6  CO7 PSO6 PSO7 PSO6  CO8 PSO7 PSO6  CO9 PSO7 PSO6  CO1 PSO6 PSO7 PSO6  CO1 PSO7 PSO6  CO1 PSO6 PSO7 PSO6  CO1 PSO6 PSO7 PSO6  CO1 PSO7 PSO6  CO1 PSO7 PSO6  CO2 PSO7 PSO6 PSO7 PSO6  CO1 PSO7 PSO6  CO1 PSO7 PSO6  CO2 PSO7 PSO6  CO3 PSO7 PSO6  CO4 PSO7 PSO6  CO5 PSO7 PSO6  CO5 PSO7 PSO6  CO6 PSO7 PSO6  CO6 PSO7 PSO6  CO6 PSO7 PSO6  CO6 PSO7 PSO6  CO7 PSO6  CO1 PSO7 PSO6  CO1 PSO7 PSO6  CO1 PSO7 PSO6  CO2 PSO7 PSO6  CO2 PSO7 PSO6  CO3 PSO7 PSO6  CO3 PSO7 PSO6  CO4 PSO7 PSO6  CO5 PSO7 PSO6  CO6		4. To	learn ana	lysis of a	lgorithm:	S					
traversing various data structure.  CO2. Explain the basics of computational complexity analysis and big O theta notation that help for fundamental research in algorit analysis. Students can select the efficient algorithm after a nalyzing problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO4 PSO5 PSO6 PSO7 PSO7  CO4 PSO7 PSO7  CO5 PSO7 PSO7  CO6 PSO7 PSO7  CO7 PSO7  CO8 PSO7 PSO8  Pre-requisite  Curse Content  Unit 1: Non-Primitive Data structures.  1.1 Arrays - its storage structures & operations 1.2 Stacks - operations and its applications. In Recursion, Polish expression etc.  1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded storal representation of Binary Trees.  1.6 Applications of Trees - The manipulation of Arithmetic expressions,	Course Outcome	CO1.	Explain t	he futur	es of vari	ous non-	primitive	data stru	cture a	nd their	
CO2. Explain the basics of computational complexity analysis and big of theta notation that help for fundamental research in algorit analysis. Students can select the efficient algorithm after analyzin, problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO4 PSO5 PSO6 PSO7 PSO6  CO4 PSO5 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO6 PSO7 PSO6  CO7 PSO6 PSO7 PSO6  CO8 PSO6 PSO7 PSO6  CO9 PSO6 PSO6 PSO7 PSO6  CO9 PSO6 PSO6 PSO7 PSO6  CO9 PSO6 PSO7 PSO6  CO9 PSO6 PSO6 PSO7 PSO6  CO9 PSO6 PSO7 PSO6  CO9 PSO6 PSO6 PSO7 PSO6  CO9 PSO6 PSO6 PSO7 PSO6  CO9 PSO6 PSO6 PSO6 PSO6  CO9 PSO6 PSO7 PSO6  CO9 PSO6 PSO6 PSO7 PSO6  CO9 PSO6 PSO6 PSO6 PSO6  CO9 PSO6 PSO6 PSO6 PSO6  CO9 PSO6 PSO6			applicati	ons. Trai	n studen	ts for alg	orithms t	o create,	insert, de	elete an d	
theta notation that help for fundamental research in algorit analysis. Students can select the efficient algorithm after analyzin problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedymethod, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO3 PSO6 PSO7 PSO6  CO4 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO6 PSO7 PSO6  CO7 PSO6 PSO7 PSO6  CO8 PSO7 PSO6  CO9 PSO7											
analysis. Students can select the efficient algorithm after analyzin, problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedymethod, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash methor and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO4 PSO5 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO6 PSO7 PSO6  CO7 PSO6 PSO7 PSO6  CO8 PSO7 PSO6  CO9 PSO7		CO2. Explain the basics of computational complexity analysis and big O &									
problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOS  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO3 PSO6 PSO7 PSO6 PSO7 PSO6  CO4 PSO6 PSO7 PSO		theta notation that help for fundamental research in algorithm									
CO3. Explain insight of basic technique for algorithm such as Divide Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO3 PSO6 PSO7 PSO6  CO4 PSO7 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO6 PSO7 PSO6  CO7 PSO6 PSO7 PSO6  CO8 PSO7 PSO6  CO9 PSO7 PSO6  CO1 PSO7 PSO6  CO9 PSO7 PSO6		problem and identifying it's the computing requirements.  CO3. Explain insight of basic technique for algorithm such as Divide & Conquer, Greedy method, Backtracking & Branch & Bound  CO4. Students will be able to work with various searching and sorting techniques and compare its computational complexity.						lyzing a			
Conquer, Greedy method, Backtracking & Branch & Bound CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity. CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6  CO3 PSO6 PSO7 PSO6  CO4 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO5 PSO6 PSO7 PSO6  CO6 PSO7 PSO6  CO7 PSO6 PSO7 PSO6  CO8 PSO7 PSO6  CO9 PSO6  CO9 PSO7 PSO6  CO9 PSO6  CO9 PSO7 PS								D::.d = 0			
CO4. Students will be able to work with various searching and sort techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash methor and collision detection techniques.  Mapping between COs with PSOs  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO4 PSO5 PSO6 PSO7 PSO7  CO5 PSO6 PSO7 PSO7  CO5 PSO6 PSO7								Divide &			
techniques and compare its computational complexity.  CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOS  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO7  CO3 PSO6 PSO7 PSO7  CO3 PSO6 PSO7 PSO7  CO4 PSO5 PSO6 PSO7 PSO7  CO5 PSO6 PSO7 PSO7  CO6 PSO7 PSO7  CO7 PSO6 PSO7 PSO7  CO8 PSO7 PSO7  CO9 PSO7 PSO6  CO9 PSO7 PSO6 PSO7 PSO7  CO9 PSO7 PSO6  CO1 PSO7 PSO6  CO9 P								d carting			
CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.  Mapping between COs with PSOs  PSO1  PSO2  PSO3  PSO4  PSO5  PSO6  PSO7  PSO6  CO1  CO2  CO3  CO4  CO5  Pre-requisite  Corpogramming Language  Course Content  Unit 1: Non-Primitive Data structures.  1.1 Arrays - its storage structures & operations 1.2 Stacks - operations and its applications in Recursion, Polish expression etc.  1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded storal representation of Binary Trees. 1.6 Applications of Trees - The manipulation of Arithmetic expressions,								a sorting			
Mapping between COs with PSOS  PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO6 CO1 CO2 CO3 CO4 CO5								method			
Mapping between COs with PSOS PSO1 PSO2 PSO3 PSO4 PSO5 PSO6 PSO7 PSO5 CO1 CO2 CO3 CO3 CO4 CO5									nethou		
CO2 CO3 CO4 CO5 Pre-requisite C programming Language  Course Content Unit 1: Non-Primitive Data structures. 1.1 Arrays - its storage structures & operations 1.2 Stacks - operations and its applications in Recursion, Polish expression etc. 1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded storage representation of Binary Trees. 1.6 Applications of Trees - The manipulation of Arithmetic expressions,	Mapping between COs with PSOs					1	PSO5	PSO6	PSO7	PSO8	
CO2 CO3 CO4 CO5 Pre-requisite C programming Language  Course Content Unit 1: Non-Primitive Data structures. 1.1 Arrays - its storage structures & operations 1.2 Stacks - operations and its applications in Recursion, Polish expression etc. 1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded storage representation of Binary Trees. 1.6 Applications of Trees - The manipulation of Arithmetic expressions,		CO1									
CO3 CO4 CO5 Pre-requisite Course Content Unit 1: Non-Primitive Data structures.  1.1 Arrays - its storage structures & operations 1.2 Stacks - operations and its applications in Recursion, Polish expression etc. 1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded storal representation of Binary Trees. 1.6 Applications of Trees - The manipulation of Arithmetic expressions,											
Pre-requisite  Course Content  Unit 1: Non-Primitive Data structures.  1.1 Arrays - its storage structures & operations 1.2 Stacks - operations and its applications in Recursion, Polish expression etc.  1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded storage representation of Binary Trees.  1.6 Applications of Trees - The manipulation of Arithmetic expressions,											
Pre-requisite  Course Content  Unit 1: Non-Primitive Data structures.  1.1 Arrays - its storage structures & operations 1.2 Stacks - operations and its applications in Recursion, Polish expression etc.  1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded storage representation of Binary Trees. 1.6 Applications of Trees - The manipulation of Arithmetic expressions,		CO4									
Course Content  Unit 1: Non-Primitive Data structures.  1.1 Arrays - its storage structures & operations 1.2 Stacks - operations and its applications in Recursion, Polish expression etc. 1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded storage representation of Binary Trees. 1.6 Applications of Trees - The manipulation of Arithmetic expressions,		CO5									
<ul> <li>1.1 Arrays - its storage structures &amp; operations</li> <li>1.2 Stacks - operations and its applications in Recursion, Polish expression etc.</li> <li>1.3 Queues - Types of queues, operations and its applications.</li> <li>1.4 Linked lists - Types of linked list, operations and its applications.</li> <li>1.5 Trees - Concept and Definitions, Operations, linked &amp; threaded storage representation of Binary Trees.</li> <li>1.6 Applications of Trees - The manipulation of Arithmetic expressions,</li> </ul>	Pre-requisite	Cprog	ramming	Languag	e	l					
<ol> <li>Stacks - operations and its applications in Recursion, Polish expression etc.</li> <li>Queues - Types of queues, operations and its applications.</li> <li>Linked lists - Types of linked list, operations and its applications.</li> <li>Trees - Concept and Definitions, Operations, linked &amp; threaded storal representation of Binary Trees.</li> <li>Applications of Trees - The manipulation of Arithmetic expressions,</li> </ol>	Course Content										
etc. 1.3 Queues - Types of queues, operations and its applications. 1.4 Linked lists - Types of linked list, operations and its applications. 1.5 Trees - Concept and Definitions, Operations, linked & threaded stora representation of Binary Trees. 1.6 Applications of Trees - The manipulation of Arithmetic expressions,		1.1 Arrays - its storage structures & operations									
<ul> <li>1.3 Queues - Types of queues, operations and its applications.</li> <li>1.4 Linked lists - Types of linked list, operations and its applications.</li> <li>1.5 Trees - Concept and Definitions, Operations, linked &amp; threaded stora representation of Binary Trees.</li> <li>1.6 Applications of Trees - The manipulation of Arithmetic expressions,</li> </ul>		1.2 Stacks - operations and its applications in Recursion, Polish expressions									
<ul> <li>1.4 Linked lists - Types of linked list, operations and its applications.</li> <li>1.5 Trees - Concept and Definitions, Operations, linked &amp; threaded stora representation of Binary Trees.</li> <li>1.6 Applications of Trees - The manipulation of Arithmetic expressions,</li> </ul>											
<ul> <li>1.5 Trees - Concept and Definitions, Operations, linked &amp; threaded stora representation of Binary Trees.</li> <li>1.6 Applications of Trees - The manipulation of Arithmetic expressions,</li> </ul>		· · · · · · · · · · · · · · · · · · ·									
representation of Binary Trees.  1.6 Applications of Trees - The manipulation of Arithmetic expressions,		· · · · · · · · · · · · · · · · · · ·									
1.6 Applications of Trees - The manipulation of Arithmetic expressions,		· · · · · · · · · · · · · · · · · · ·									
·											
Jynnbol-table constituction, Jyntax Analysis etc.						-			exhiessi	U115,	
		Symbol table collab action, Symanymaysis etc.									
Unit 2: Analysis of Algorithms		Unit 2: Analysis of Algorithms									
2.1 Asymptotic: Big-O and Theta											
Unit 3: Basic techniques & example algorithms for		Unit 3: Basic techniques & example algorithms for									
3.1 Divide & Conquer method				-	-						
3.2 Greedy method		·									

	3.3 Backtracking 3.4 Branch & Bound
	Unit 4: Searching and Sorting  4.1 Sequential, Binary  4.2 Search Trees:- Height, Balanced tree, 2-3, tree, red-black trees weight-balanced trees  4.3 Sorting  4.3.1 Internal sorting - Insertion, Selection, Quick, 2-way merge and Heap  4.3.2 External sorting - k-way merging, Balanced merge and polyphase Merge
	Unit 5: Hashing 5.1 Hash Tables 5.2 Hash functions 5.2.1 Division method 5.2.2 Multiplication method
	[Self Study] Graphs – Creation and Traversal
Reference Books	<ol> <li>1. An Introduction to Data Structures with applications - Trembley – McGraw Hill</li> <li>2. Theory and Problems of Data Structure – Lipschutz Semour – McGraw Hill</li> <li>3. Algorithms + Data Structure Programs - Wirth, Niclaus - PHI.</li> <li>4. Fundamentals of Data Structures, Horwitz, E. and Sahni S Computer Science Press.</li> <li>5. The Art of Computer Programming, Vols. 1-2, Knuth D Addison Wesley.</li> <li>6. Data Structures and Algorithms - Aho A.V., Hopcroft and Ullman - Addison Wesley</li> <li>7. Data Structure &amp; "C" Programming - Vanwyte C J - Addison Wesley.</li> <li>8. Data Structures, Algorithms And Object Oriented Programming – Tata McGraw Hill edition Geogory L. Heileman.</li> <li>9. Data Structures and the Standard Template Library - William J. Collins, Tata McGraw Hill edition.</li> <li>10. Programming with C++ and Data Structures - Maria Litvin &amp; Gary Litvin, Vikas Publishing House Pvt. Itd.</li> <li>11. Data Structures using C &amp; C++ - Y. Langsam Moshe J. Angensterin &amp; A.M. Tenenbaum</li> <li>12. Data Structures and Algorithms in C++ - Adam Drozdek, Thomson Learning</li> <li>13. Data Structures &amp; Program Design in C - Robert Kruse, C.L. Tondo, Brnceleing PHI Pvt Ltd.</li> <li>14. Data Structures and Algorithms in Java, Lafore, Pearson</li> <li>15. Data Structures and Algorithm Analysis in Java, Mark Allen Weiss, Pearson</li> <li>16. Data Structures and Algorithms in Java, Micheal T Goodrich, Roberto</li> </ol>
Teaching Mothodology	Tamassia, Wiley Class Work, Discussion, Self Study, Seminars and /or Assignment
Teaching Methodology Evaluation Method	30% Internal assessment 70% External Assessment