structures to have knowledge and skills on the subject.Course ContentData concept and data types, Lists, Queue, Stack , binary tree and application areas compression algorithms, Sorting algorithms, search algorithms and hash tables Classification of data structures, introduction to algorithm and complexity analysis lists, basic data structures and their derivatives, such as stacks , queues, trees, graphs and stacks, analysis of key sorting algorithms, program using dynamic memory allocation emphasis on writing, practical exercises in laboratories and programming homework.Course Learning OutcomesStudents who successfully complete this course; 1. Choosing the appropriate data structure 2. Analyzing algorithms 3. Data abstraction capability 4. More efficient programming 5. Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms.	Course title			Code	semester	T+U	credit	ECTS			
Language of the Course Course Level English Undergraduate Type of Course Course Coordinator Compalsory Course Coordinator English The aim of lesson computer memory. design, analysis and development of students' basic data structures to have knowledge and skills on the subject. Course Content Data concept and data types, Lists, Queue, Stack. binary tree and application areas compression algorithms, Sorting algorithms, search algorithms and hash tables Classification of data structures, introduction to algorithm and complexity analysis lists, basic data structures, and their derivatives, such as stack, queues, trees, graphs and stacks, analysis of key sorting algorithms, program using dynamic memory allocation emphasis on writing, practical exercises in laboratorics and programming homework. Course Learning Outcomes Students who successfully complete this course; 1. Choosing the appropriate data structure 2. Analyzing algorithms 3. Data abstraction capability 4. More efficient programming 5. Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. 6. lists, stacks, queues, trees, graphs and stacks; the ability to use basic data structures for problem solving. 7. Ability to use and compare data structure and application areas. 8. Ability to use and compare totals for applications 2 linked list, bidirectional lists, unidirectional linked lists, Stack data structure and applications 3 Queue data structure and applications, Priority queue and Heap tree <	Data Structur	es			3	3+0	3	5			
Language of the Course Course Level Undergraduate Type of Course Course Coordinator Compalyory Course Coordinator Instructors Course Assistants Presenting basic data structures designed to store and access information in computer memory. design, analysis and development of students' basic data structures to have knowledge and skills on the subject. Course Content Data concept and data types, Lists, Queue, Stack. binary tree and application areas compression algorithms, Sorting algorithms, search algorithms and hash tables Classification of data structures, introduction to algorithm and complexity analysis lists, basic data structures and their derivatives, such as stack, queues, trees, graphs and stacks, analysis of key sorting algorithms, sporgam using dynamic memory allocation emphasis on writing, practical exercises in laboratories and programming homework. Outcomes 1. Choosing the appropriate data structure 1. Choosing the appropriate data structures 2. Analyzing algorithms. 3. Data abstraction capability 4. More efficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. 6. lists, stacks, queues, trees, graphs and stacks; the ability to use basic data structures for problem solving. 7. Ability to use and compare data structure and application areas. 8. Ability to use and compare data structure and applications 6. lists, stacks, queues, trees, graphs and stacks; the ability to use basic data structures for problem solving. 9 Inked list, bidirect	Prerequisite	Courses	None					1			
Type of Course Compulsory Course Coordinator Instructors Course Assistants Presenting basic data structures designed to store and access information in computer memory. design, analysis and development of students' basic data structures to have knowledge and skills on the subject. Course Content Data concept and data types, Lists, Queue, Stack, binary tree and application areas compression algorithms, Sorting algorithms, search algorithms and complexity analysis lists, basic data structures, introduction to algorithm and complexity analysis lists, basic data structures and their derivatives, such as stacks, queues, trees, graphs and stacks, analysis of key sorting algorithms, program using dynamic memory allocation emphasis on writing. practical exercises in laboratories and programming homework. Course Learning Students who successfully complete this course; Outcomes 1. Choosing the appropriate data structure 2. Analyzing algorithms. 3. Data abstraction capability 4. More efficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. 5. Sufficient Knowledge of sequential algorithm analysis; ability to use basic data structures for problem solving. 7. Ability to use and compare data structures and application areas. 8. Ability to use and compare data structure and applications 2 Inked list, bidirectional lists, unidirectional linked lists , Stack data structure and applications 3 Qu											
Course Coordinator Instructors Instructors Course Assistants The aim of lesson Presenting basic data structures designed to store and access information in computer memory. design, analysis and development of students' basic data structures to have knowledge and skills on the subject. Course Content Data concept and data types, Lists, Queue, Stack, binary tree and application areas and stacks and a stacks, analysis of key sorting algorithms, search algorithms, sorting and porthums, sorting and porthums, program using dynamic memor allocation emphasis on writing, practical exercises in laboratories and programmin homework. Course Learning Students who successfully complete this course; Outcomes 1. Choosing the appropriate data structure 2. Analyzing algorithms 3. Data abstraction capability 4. More efficient programming 5. Sufficient knowledge of sequential algorithm analysis; ability to use basic data structures for problem solving. 7. Ability to use and compare data structures in different application areas. 8. Ability to use and compare data structure and applications 2 Inkely to use and compare data structure and applications 3 3 Queue data structure and applications. 5 4 Tree data structure and applications. 5 5 Data comperesion algorithms. Sorting methods (Insertion , Selection , Bubble)	Course Level										
Instructors Course Assistants Course Assistants Presenting basic data structures designed to store and access information in computer memory. design, analysis and development of students' basic data structures to have knowledge and skills on the subject. Course Content Data concept and data types, Lists, Queue, Stack , binary tree and application areas compression algorithms, Sorting algorithms, search algorithm and complexity analysis lists, basic data structures and their derivatives, such as stacks , queues, trees, graph and stacks, analysis of key sorting algorithms, program using dynamic memory allocation emphasis on writing, practical exercises in laboratories and programming homework. Course Learning Students who successfully complete this course; Outcomes 1. Choosing the appropriate data structure 2. Analyzing algorithms 5. Bata abstraction capability 4. More efficient programming 5. Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. 7. Ability to use and compare data structures in different application areas. 8. Ability to use and compare data structure and applications 2 Iinked list, bidirectional lists, unidirectional linked lists , Mack data structure and applications. 3 Queue data structure and applications. Priority queue and Heap tree 4 Tree data structure and applications. Priority queue and Heap tree 5 Data Compression algorithms , Sortin			Compulsory								
Course Assistants Presenting basic data structures designed to store and access information in computer memory. design, analysis and development of students' basic data structures to have knowledge and skills on the subject. Course Content Data concept and data types, Lists, Queue, Stack, binary tree and application areas compression algorithms, Sorting algorithms, search algorithms and that hables Classification of data structures and their derivatives, such as stacks, queues, trees, graphs and stacks, analysis of key sorting algorithms, program using dynamic memory allocation emphasis on writing, practical exercises in laboratories and programming homework. Course Learning Students who successfully complete this course; Outcomes I. Choosing the appropriate data structure Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. Sufficient knowledge of sequential algorithm analysis; ability to use basic data structures for problem solving. One Basic data tructure and compare data structures in different application areas. Ability to use not complete this, stack data structure and applications Queue data structure and applications. Priority queue and Heap tree Meeks Topics One Basic data structures Binary search trees and applications S Data Compression algorithms. Sorting methods (Insertion , Selection , Bubble) G Sorting methods (Merge, Quick, Heap), Searc		dinator									
The aim of lesson Presenting basic data structures designed to store and access information in computer memory. design, analysis and development of students' basic data structures to have knowledge and skills on the subject. Course Content Data concept and data types, Lists, Queue, Stack , binary tree and application areas compression algorithms, sorting algorithms, segred algorithms and hash tables Classification of data structures, introduction to algorithm and complexity analysis in and stacks, analysis of key sorting algorithms, prearent algorithms and thash tables Course Learning Outcomes Students who successfully complete this course; A more efficient programming Students who successfully complete this course; A more efficient programming Students who successfully complete this course; A bility to use and compare data structures in different application areas. 8. Ability to use and compare data structures in different applications areas. One Basic data types and data concept . List data structure and appli	11001 00010										
computer memory. design, analysis and development of students' basic data structures to have knowledge and skills on the subject. Course Content Data concept and data types, Lists, Queue, Stack, binary tree and application areas compression algorithms, Sorting algorithms, search algorithms and hash tables. Classification of data structures and their derivatives, such as stacks, queues, trees, graphs and stacks, analysis of key sorting algorithms, program using dynamic memory allocation emphasis on writing, practical exercises in laboratories and programming homework. Course Learning Outcomes Students who successfully complete this course; 1. Choosing the appropriate data structure 2. Analyzing algorithms 3. Data abstraction capability 4. More efficient programming 5. Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. 6. lists, stacks, queues, trees, graphs and stacks; the ability to use basic data structures for problem solving. 7. Ability to use and compare data structures in different application areas. 8. Ability to use necessary tools for application design and development Weeks One Basic data types and data concept , List data structure and applications 2 Inked list, unidirectional linked lists , Stack data structure and applications 3 Queue data structure and applications , Priority queue and Heap tree 4 Tree data structure and applications , Binary search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 9 9 Algorithm Complexity 9											
compression algorithms, Sorting algorithms, search algorithms and hash tables Classification of data structures, introduction to algorithm and complexity analysis lists, basic data structures and their derivatives, such as stacks, queues, trees, graphs and stacks, analysis of key sorting algorithms, program using dynamic memory allocation emphasis on writing, practical exercises in laboratories and programming homework. Course Learning Outcomes Students who successfully complete this course; 1. Choosing the appropriate data structure 2. Analyzing algorithms. 3. Data abstraction capability 4. More efficient programming 5. Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. 6. lists, stacks, queues, trees, graphs and stacks; the ability to use basic data structures for problem solving. 7. Ability to use and compare data structure in different application areas. 8. Ability to use necessary tools for application design and development Weeks Topics one Basic data types and data concept , List data structure and applications 3 Queue data structure and applications , Priority queue and Heap tree 4 Tree data structure and applications , Stack data structure and applications 5 Data Compression algorithms a	The aim of lesson		computer memory. design, analysis and development of students' basic data								
Outcomes 1. Choosing the appropriate data structure 2. Analyzing algorithms 3. Data abstraction capability 4. More efficient programming 5. Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. 6. lists, stacks, queues, trees, graphs and stacks; the ability to use basic data structures for problem solving. 7. Ability to use and compare data structures in different application areas. 8. Ability to use necessary tools for application design and development Topics One Basic data types and data concept. List data structure and applications 2 linked list, bidirectional lists, unidirectional linked lists , Stack data structure and applications 3 Queue data structure and applications , Priority queue and Heap tree 4 Tree data structure and applications , Binary search trees and applications 5 Data Compression algorithms , Sorting methods (Insertion , Selection , Bubble) 6 Sorting methods (Merge , Quick , Heap) , Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13	Course Content		compression algorithms, Sorting algorithms, search algorithms and hash tables, Classification of data structures, introduction to algorithm and complexity analysis, lists, basic data structures and their derivatives, such as stacks, queues, trees, graphs, and stacks, analysis of key sorting algorithms, program using dynamic memory allocation emphasis on writing, practical exercises in laboratories and programming								
2. Analyzing algorithms 3. Data abstraction capability 4. More efficient programming 5. Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. 6. lists, stacks, queues, trees, graphs and stacks; the ability to use basic data structures for problem solving. 7. Ability to use and compare data structures in different application areas. 8. Ability to use necessary tools for application design and development Weeks One Basic data types and data concept . List data structure and applications 2 linked list, bidirectional lists, unidirectional linked lists , Stack data structure and applications 3 Queue data structure and applications , Priority queue and Heap tree 4 Tree data structure and applications , Binary search trees and applications 5 Data Compression algorithms , Sorting methods (Insertion , Selection , Bubble) 6 Sorting methods (Merge , Quick , Heap) , Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions	Course Lear	ning	Students w	ho successfu	lly complete the	nis course;					
One Basic data types and data concept , List data structure and applications 2 linked list, bidirectional lists, unidirectional linked lists , Stack data structure and applications 3 Queue data structure and applications , Priority queue and Heap tree 4 Tree data structure and applications , Binary search trees and applications 5 Data Compression algorithms , Sorting methods (Insertion , Selection , Bubble) 6 Sorting methods (Merge , Quick , Heap) , Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure			 Analyzing algorithms Data abstraction capability More efficient programming Sufficient knowledge of sequential algorithm analysis; ability to analyze sequential algorithms. lists, stacks , queues, trees, graphs and stacks; the ability to use basic data structures for problem solving. Ability to use and compare data structures in different application areas. 								
2 Iinked list, bidirectional lists, unidirectional linked lists, Stack data structure and applications 3 Queue data structure and applications, Priority queue and Heap tree 4 Tree data structure and applications, Binary search trees and applications 5 Data Compression algorithms, Sorting methods (Insertion, Selection, Bubble) 6 Sorting methods (Merge, Quick, Heap), Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity, Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies	Weeks	Topics									
3 Queue data structure and applications , Priority queue and Heap tree 4 Tree data structure and applications , Binary search trees and applications 5 Data Compression algorithms , Sorting methods (Insertion , Selection , Bubble) 6 Sorting methods (Merge , Quick , Heap) , Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies	one	Basic data ty	pes and data con	cept , List da	ta structure an	d application	18				
4 Tree data structure and applications, Binary search trees and applications 5 Data Compression algorithms, Sorting methods (Insertion, Selection, Bubble) 6 Sorting methods (Merge, Quick, Heap), Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity, Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies	2	linked list, bi	directional lists,	unidirectiona	al linked lists ,	Stack data s	tructure and	applications			
4 Tree data structure and applications , Binary search trees and applications 5 Data Compression algorithms , Sorting methods (Insertion , Selection , Bubble) 6 Sorting methods (Merge , Quick , Heap) , Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure	3	Queue data st	ructure and app	lications . Pri	ority queue ar	d Heap tree					
5 Data Compression algorithms , Sorting methods (Insertion , Selection , Bubble) 6 Sorting methods (Merge , Quick , Heap) , Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure	4						tions				
6 Sorting methods (Merge, Quick, Heap), Search Methods 7 Hash tables 8 Graph data structures 9 Algorithm Complexity, Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure	5										
7 Hash tables 8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure	6						,				
8 Graph data structures 9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies			Jus (wierge , Ql	лек, пеар <i>)</i>	, Scaren Melli	ous					
9 Algorithm Complexity , Recursion, Abstract Data Structures 10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure											
10 Stacks and Queues 11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure											
11th Linked Lists and Trees 12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure		Algorithm Complexity, Recursion, Abstract Data Structures									
12 Binary Search Trees and Snap Functions 13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure		Stacks and Queues									
13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure		Linked Lists and Trees									
13 heaps 14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure	12	Binary Search Trees and Snap Functions									
14 charts 15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure	13										
15 Algorithm Complexity General Competencies Designing and choosing the appropriate data structure	14										
General Competencies Designing and choosing the appropriate data structure	15										
Designing and choosing the appropriate data structure				General Co	mpetencies						
resources	Designing and	d choosing the									
				resou	rces						

R. ÇÖLKESEN, "Data structures and algorithms", Papatya publishing, 2002. Allen M., (1999). data structures and problem solving using C++, Addison Wesley . Ford W. & Topp W., (2002). data structures with C++, Prentice Hall

Evaluation System

The dates, days and hours of the Midterm Exam, Quiz, Final Exam and Evaluations will be announced later, according to the decision of the Faculty Administrative Board.

				WITH	PROGRAM	M LEARN	ING OUT	COMES				
	COURSE LEARNING OUTCOMES RELATIONSHIP TABLE											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	
L01	5	5	5	5	4	4	4	5	5	4	4	
LO2	4	4	4	4	4	3	3	3	5	3	4	
ÖK3	5	5	5	4	5	3	3	3	3	3	3	
ÖK4	5	4	5	3	5	4	3	3	3	3	3	
ÖK5	5	5	5	4	5	3	3	3	3	3	3	
OK6	5	5	5	5	4	4	4	5	5	4	4	
OK7	5	4	4	4	4	3	3	3	5	3	5	
OK8	5	4	5	4	5	3	3	3	3	3	3	
			LO	Learning	Outcome	s OP: Prog	ram Outc	omes				
Contri bution Level	1 Very Low		2 Low		3 Medium 4 High			1	5 Very High			

Relation of Program Outcomes and Related Course

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
Data Structures 1	5	5	5	4	5	3	4	4	3	4	3