UNIVERSITY OF ZAKHO
DIRECTORATE OF QUALITY ASSURANCE

Data Structures
Course Book

Faculty/ College: Faculty of Science

Department of: Computer Science

Year: 2

Module Name: Data Structures

Academic Year: 2019-2020

1. Introductory information:

Module Name	Data Structures
Module Code	Click or tap here to enter text.
Module Type	Requisite
Hours per week	Theory (2 hours), practical (2 hours)
No. of ECTS	7
Semester	4
Lecturer(s) in charge	Dr. Abdulhakeem Othman Mohammed
Academic Title	Lecturer
Department	Computer Science
Faculty /College	Faculty of Science
Contact details	E-mail address: a.mohammed@uoz.edu.krd
	Mobile Number: Click or tap here to enter text.

2. Module Overview:

Well organised data structures allow for quick and efficient retrieval of information and are essential for modern computing. Organised data can be easily sorted, ordered, and searched to retrieve information that meets certain requirements. Additionally, a good algorithm usually comes together with a set of good data structures that allow the algorithm to manipulate the data efficiently. In this course we will look at the core of data structures used in everyday applications. This is an introductory level course on data structures and algorithms, as well as how to program them in C++. Topics included are fundamental data structures such as linked lists, queues, stacks, trees, heaps, hash tables. Sorting and searching algorithms: selection, heap, quick, merge sort, binary, tree-based, hash-based search. Graphs: storage, search, traversals.

3. Module Objective:

Students will explore the importance and impact of well organised data and learn how to build a program from small pieces and understand why organisational approaches make such a difference to some very common approaches to solutions. We use C++ as the language of implementation, but the emphasis of the course itself remains on uniformly accepted topics such as pointers, data structures, algorithm analysis, and increasingly programming projects. Additionally, Students will

- Become familiar with some of the fundamental data structures in computer science.
- Learn both data structures and algorithm design from the viewpoint of abstract thinking and problem solving.
- Develop programming language and OOP skills.
- Develop algorithms for manipulating stacks, queues, linked lists, hash tables, trees, and graphs.
- Know the trade-offs of each studied data structure so as to employ the appropriate one for a given situation.
- Be able to analyze data structures and their algorithms for asymptotic behaviour.

4. Assessment Strategy:

Task(s)	Marks
Homework	8%
Quizzes	8%
Lab Assignments	12%
Projects	12%
Mid-term examination	20%
Final examination	Practical 10 %, Theory 30%, Total 40%

5. Learning OUTCOMES:

Having successfully completed this course, students will be able to apply and compare the basic abstract data structures used in computer science. These include: stacks, queues, linked lists, heap, priority queue and binary trees. Students will understand and demonstrate their ability to construct abstract data types and solve problems using an object oriented programming language. This will include demonstrated understanding of recursion, dynamic memory management. The student will be able to analyse, evaluate and choose appropriate abstract data types and algorithms to solve particular problems.

6. Teaching and Learning Methods:

This course is two hours lecture. We have Labs to program the data structures discussed in the class. We will a have some assignments and quizzes which will be a good preparation for students for the exam. Additionally, there will be two projects.

7. Module Reading List and References:

Textbooks:
1) Algorithm Design: Foundations, Analysis, and Internet Examples, Wiley, 2002, ISBN 0-471-38365-1 (by Michael T. Goodrich and Roberto Tamassia)
2) *Introduction to Algorithms*, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, 3rd edition, MIT Press, 2009

8. Syllabus:

Weeks	Topic(s)/ Theoretical	Topic(s)/ Practical	Date
1	Syllabus + Introduction + Algorithm Analysis	Review Week for Programming Fundamentals and OOP	Click or tap here to enter text.
2	Stack	Review Week for Programming Fundamentals and OOP	
3	Queue + Circular Queue	Stack Implementation	Click or tap here to enter text.
4	Linked List	Queue Implementation	Click or tap here to enter text.
5	Double Linked List + **Homework1 Deadline**	Linked List Implementation	Click or tap here to enter text.
6	Recursion + **Quiz1**	Double Linked List Implementation	Click or tap here to enter text.
7	Introduction to trees and Tree Traversals	**Stack and Queue implementations With Linked List Project Deadline**	[
8	Binary Trees and Binary search Trees + **Homework2 Deadline**	Tree Implementation	
9	**Midterm Exam**		
10	Priority Queue and Heap	Binary Tree Implementation	Click or tap here to enter text.
11	Dictionaries and Hash Tables	Tree Traversals Implementation	Click or tap here to enter text.

12	Sorting Algorithms	Hash Tables Implementation	Click or tap here to enter text.
13	Sorting Algorithms	Sorting Algorithms	Click or tap here to enter text.
14	Introduction into Graph	Sorting Algorithms	Click or tap here to enter text.
15	Introduction into Graph		Click or tap here to enter text.

9. Examinations:

1. ***Compositional:***

   Q) Insert into an initially empty binary search tree items with the following keys (in this order): 30, 40, 23, 58, 48, 26, 11, 13. Draw the tree after all insertions. Include a few intermediate stages.

2. ***Multiple choices:***

   Q) A queue follows ___________

   a) FIFO (First In First Out) principle
   b) LIFO (Last In First Out) principle
   c) Ordered array
   d) Linear tree

10. Peer Review:

Name	Click or tap here to enter text.
Academic Title	Choose an item.
E-mail	Click or tap here to enter text.
Signature	Click or tap here to enter text.

Does the course book meet the requirements of the module? Choose an item.
If (NO), please specify what needs to be changed in the table below