QF01/0408-4.0E	Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Cyber Security Department

Study plan No.	**2022/2021**		**University Specialization**		**Cyber security**		
Course No.	**0125312**		**Course name**		**Data Structure and Algorithms**		
Credit Hours	**3**		**Prerequisite/ Co-requisite**		Object Oriented Programming		
Course type	☐ MANDATORY UNIVERSITY REQUIREMENT	☐ UNIVERSITY ELECTIVE REQUIREMENTS	☐ FACULTY MANDATORY REQUIREMENT	☐ Support course family requirements	√Mandatory requirements	☐ Elective requirements	
Teaching style	☐ **Full online learning**		√**Blended learning**		☐ **Traditional learning**		
Teaching model	☐ **1 Synchronous: 1 asynchronous**		√**1 face to face : 1 asynchronous**		☐ **2 Traditional**		

Faculty member and study divisions' information (to be filled in each semester by the subject instructor)

Name	Academic rank	Office No.	Phone No.	E-mail	
Division number	**Time**	**Place**	**Number of students**	**Teaching style**	**Approved model**
				Blended learning	**1 face to face : 1 asynchronous**

Brief description

The Data Structure and Algorithm course sets out the structuring principles, Abstract Data Types (ADT) and Implementations: Lists, Stacks, Queues, Priority Queues, Recursion. Introduction to algorithm analysis. Introduction of search and sort algorithms including Trees and Binary Search Trees, Hashing, and Heaps. In a high-level language (usually Java) the student should implement the user-defined data structures. Student can compare performance-related alternative implementations of data structures. Write programs that use the arrays, records, strings, linked lists, stacks and queues of each of the following data structures.

Learning resources

Course book information (Title, author, date of issue, publisher ... etc)	F. M. Carrano and T. M. Henry: Data Structures and Abstractions with Java, 5th edition, Pearson, 2019.			
Supportive learning resources (Books, databases, periodicals, software, applications, others)	Nell Dale, Daniel T. Joyce, Chip Weems ,Object-oriented data structures using Java,2016 Data Structures and Algorithms in Java 6th Edition by Michael T. Goodrich, Roberto Tamassia			
Supporting websites				
The physical environment for teaching	☐ √Class room	labs	☐ Virtual educational platform	☐ Others

Necessary equipment and software	
Supporting people with special needs	
For technical support	

Course learning outcomes (S= *Skills, C= Competences K= Knowledge,)*

No.	Course learning outcomes	The associated program learning output code
	Knowledge	
K1	Recognize the concept of an Abstract Data Type (ADT).	MK4
K2	Determine how an ADT is designed and implemented as a class of some object-oriented programming language.	MK4
K3	Understanding the concepts of time and space complexity, worst case, average case and best case complexities and the big-O notation	MK4
K4	Understanding a wide range of searching and sorting algorithms	MK4
	Skills	
S1	Apply some basic complexity analysis methods.	MS4
S2	Improve the programming skills of students, especially in Java.	MS4
S3	Enable students to design and implement some user-defined data structures (lists, stacks, queues, linked lists, binary trees, etc.) as Java generic classes.	MS4
	Competences	
C1	Give students some practice in the application of new user-defined data structures	MC2

Mechanisms for direct evaluation of learning outcomes

Type of assessment / learning style	Fully electronic learning	Blended learning	Traditional Learning (Theory Learning)	Traditional Learning (Practical Learning)
Midterm exam	30%	**30%**	40%	30%
Participation / practical applications	0	**0**	10%	30%
Asynchronous interactive activities	30%	**20%**	0	0
Final exam	40%	**50%**	50%	40%

Note 1: Asynchronous interactive activities are activities, tasks, projects, assignments, research, studies, projects, work within student groups ... etc, which the student carries out on his own, through the virtual platform without a direct encounter with the subject teacher.

Note 2: According to the Regulations of granting Master's degree at Al-Zaytoonah University of Jordan, 40% of final evaluation goes for the final exam, and 60% for the semester work (examinations, reports, research or any scientific activity assigned to the student).

Schedule of simultaneous / face-to-face encounters and their topics

Week	Subject	learning style*	Reference **
1	**Introduction to data structures:** What is a data structure? References, arrays, Big-O Analysis. **Concept of an ADT**: Definition of an Abstract Data Type (ADT), representation of objects, implementation of operations.	**Lecture**	Ref.1: 28-68
2	**Introduction to Linked Lists**: Array vs. Linked Lists, operations on Linked Lists. **Stack ADT**: Stack ADT definition and its array implementation.	**Lecture**	Ref.1: 102-112 Ref.1: 160-162,185-193
3	**Stack ADT**: Linked stack implementation, applications of stacks (Evaluating Postfix Expressions). **Recursion**: recursive definitions, how recursion works, classic examples.	**Lecture**	Ref.1: 194-229 Ref.1: 243-253
4	**Recursion**: Recursive processing of linked lists, deciding when to use recursion. **Queue ADT**: Queue ADT definition and its linear array implementation.	**Lecture**	Ref.1: 269-285 Ref.1: 297-314
5	**Queue ADT**: Circular Queue implementation, Queue implementation as a linked structure, applications of queues.	**Lecture**	Ref.1: 331-339
6	Review of previous chapters + solutions of problems. Analysis of exam results.	**Lecture**	
7	**List ADT**: Comparing Objects, varieties of lists, List ADT specifications, array implementation of sorted and unsorted lists.	**Lecture**	Ref.1: 383-413
8	**List ADT**: Binary Search algorithm, recursive binary search, implementing List ADT as a linked structure.	**lecture**	Ref.1: 425-444
9	**List ADT**: Circular linked lists, doubly linked linear and circular lists, linked lists with headers and trailers, linked list as an array of nodes.	**lecture**	Ref.1: 474 - 496
10	**Review of Previous Chapters MID Exam: 30%**	**Lecture**	

11	– **Binary Search Trees**: Binary search tree specification, binary search tree implementation.	**Lecture**	Ref.1: 536-554
12	-Algorithms for sorting: insertion sort and merge sort. Sorting in Linear Time	**Lecture**	Ref.1: 555-577
13	-Divide and conquer in the context of merge sort	**Lecture**	Ref.1: 584-598
14	Minimum Spanning Trees, Shortest Paths	**Lecture**	Ref.1: 600-608
15	**Review of Previous Chapters** – **Discussions of Reports and Home Works:10%**		
16	**Final Exam 50%**		

*** Learning styles: Lecture, flipped learning, learning through projects, learning through problem solving, participatory learning ... etc.**
**** Reference: Pages in a book, database, recorded lecture, content on the e-learning platform, video, website ... etc.**

Schedule of asynchronous interactive activities (in the case of e-learning and blended learning)

Week	Task / activity	Reference	Expected results
1	Assignment	Lectures 1 and 2	Understanding Abstract Data Type (ADT),
2	Assignment	Lectures 3 and 4	
3	Fill in blanks, drag the words	Lectures 5 and 6	Understanding **Stack ADT**
4	Fill in blanks, drag the words	Lectures 7 and 8	Understanding **Linked Lists**
5	Assignment	Lecture 9 and 10	**Recursion**
6	Assignment	Lecture 11 and 12	**Queue ADT**
7	Assignment	Lecture 13 and 14	
8	Fill in blanks, drag the words	Lecture 15 and 16	Binary Search algorithm
9	Assignment	Lecture 17 and 18	Understanding circular lists
10	Fill in blanks, drag the words	Lecture 19 and 20	Understanding
11	Assignment	Lecture 21 and 22	
12	Assignment	Lecture 23 and 24	insertion sort
13	Assignment	Lecture 25 and 25	Understanding merge sort
14	Assignment	Lecture 25 and 25	Minimum Spanning Trees, Shortest Paths
15	Discussion forum	Review lectures	Review final exam materials
16	Final exam		