



COURSE SYLLABUS

I- Course Information :

Title of the course	Data Structures
Department	Data science
Specialization	Data science
Academic Year	2020-2021
Semester Duration	4 months
Credit Number	
Number of Hours	55 hours
Course Instructor	Mira Abboud
Email	mira.abboud@ul.edu.lb
Office Hours	2 hours / week

II- Course Description

This course covers the **design, complexity, implementation, and applications** of data structures and algorithms to solve programming problems using an object oriented programming language (**JAVA**).

Mainly, and not limited to, the course includes the following subjects:

- **Part 1** (elementary data structures): arrays, linked lists, Doubly Linked List, Circular Linked List, Stacks, Queues + main operations complexity
- **Part 2** (advanced data structures): Recursion, trees (AVL, heap, BST) + hash tables + main operations complexity
- **Part 3**: Sorting algorithms + algorithms complexity

III- Course Objectives

This course aims to provide students with/allows students to:

- **Foundational Knowledge Outcomes**
Students who successfully complete this course will be able to:
 - Apply problem to design well-structured algorithms from problem statements.
 - Demonstrate ability to design, implement, and use ADT (Abstract Data Type): arrays, stacks, queues, linked lists, hash tables, trees.
 - Demonstrate ability to apply data structures to solve well-known engineering problems like palindrome, syntax error detection, convert infix, postfix and prefix.
 - Make use of recursive algorithm in problem solving and programming.



- Choose appropriate data structures and algorithms design techniques to implement programs for problems of substantial complexity.

➤ **Application Outcomes**

This course forms a basis of almost all other courses. A good understanding of data structures and algorithms is essential to every other computer sciences/data science subjects.

➤ **Integration Outcomes**

Learn how and when to use different data structures and search and sorting algorithms.

➤ **Human Dimension Outcomes**

Drive and passion to be creative, ambitious, and the strive to succeed.

➤ **Caring Outcomes**

Every student is a potential scientist and thinker, and one.

➤ **Learning-How-to-learn Outcomes**

The primary objective will be to inspire students to love the subject matter. Once this happens, students will strive on their own for the rest of their career to learn on their own.

In order to achieve the objectives of the course, the following requirements must be met by the student:

- Scientific research
- Academic regulations
- Academic work and perseverance
- Attendance and participation

IV- Course Outline and Calendar

Learning Objective	Session(s)	Learning Content	Pedagogical Strategy & Tools
Introduction to data structures	Session 1	Course syllabus + Introduction to Data Structures	Interactive course (chapter0)
Part 1 : Linear Data structures			
Singly Linked Lists	Sessions 2-5	Introduction to Singly Linked List (Node, reference, head) + operations (insert/delete/edit elements in different positions, find, print, etc.) + operations complexity	Interactive course (chapter 1) + JAVA implementation on Eclipse
	Assign graded homework on the concept of the Circular LL. The students have to implement different operations. Deadline: to be delivered before session 9		
Doubly Linked Lists	Sessions 6-8	Introduction to Doubly Linked List (Node, previous and next references, head) + operations	Interactive course (chapter 2) + JAVA implementation on Eclipse



		(insert/delete/edit elements in different positions, find, print, etc.) + operations complexity	
Circular Linked Lists	Sessions 9-10	Introduction to Circular Linked Lists + operations (insert/delete/edit elements in different positions, find, print, etc. + operations complexity	Correct the homework + flipped classroom + discussion of their implementations.
	Assign projects on different topics. Deadline: to be delivered before session 34		
Stacks	Sessions 11-12	Introduction to Stacks (LIFO) + main operations (push, pop, top, is Empty) + operations complexity	Interactive course (chapter 3) + 2 different JAVA implementations using arrays and linked lists
	Session 13	Stack applications (checking syntax errors like symbols balancing + postfix evaluations)	Graded Lab work
Queues	Sessions 14-15	Introduction to Queues (FIFO) + main operations (enqueue, dequeue, isEmpty, front) + operations complexity + Queues application	Interactive course (chapter 4) + JAVA Implementation
30 mns quiz	Session 16	part 1	Quiz + correction
Algorithms complexity	Sessions 17-18	Types of analysis, big-O, Omega-Q, Theta, summations	Interactive course (chapter 5) + exercises
Part 2: non-linear Data structures			
Recursion	Sessions 19-21	Recursion + recursive functions complexity	Interactive course (chapter 6) + JAVA Implementation + exercises on recursive functions complexity
Trees	Sessions 22-25	Trees concept + Introduction to Binary Search Tree (BST) + operations (insert/delete(3 cases)/edit elements in different positions, find, print, etc.) + operations complexity	Interactive course (chapter 7) + exercises on insert/delete operations + JAVA implementation
	Sessions 26-28	Introduction to AVL tree + operations (insert (4cases) /delete/edit elements in different positions, find, print, etc.) + operations complexity	Interactive course (chapter 7) + exercises on insert/delete operations + JAVA implementation
	Assign graded homework on AVL and BST. The students have to implement other operations. Deadline: to be delivered before session 34		
	Sessions 29-30	Introduction to Heaps + operations (insert /delete/edit elements in different positions,	Interactive course (chapter 8) + JAVA Implementation using arrays and pseudo code on how



		find, print, percolate up/down, etc.) + operations complexity	it could be implemented using nodes
Hash Map	Sessions 31-33	HashMap (Keys, values, hash function, collisions, linear probing, quadratic probing, double hashing)	Interactive Course (chapter 9) + exercises on probing + JAVA implementation using linear probing
Part 3: Sorting algorithms			
Sorting Algorithms	Sessions 34	Introduction to Sorting algorithms + studying the complexity of classic methods (best, average, worst cases) + Selection sort	Interactive course (chapter 10) + JAVA implementation of selection sort
	Session 35	Insertion sort + Bubble Sort + studying their complexities	Interactive course (chapter 10) + JAVA implementation of Insertion and Bubble sorts
	Session 36	Quick sort + Merge Sort + studying their complexities	Interactive course (chapter 10) + JAVA implementation of Quick and Merge sorts
Course Review	Session 37	Course review	Questions + correction of last year final, midterm, quiz
Final Exam			

Note: After a week of a project submission, 10 mns will be added on some sessions for project presentation/questions.

V- Teaching approach & methods

- Most of the sessions will follow the Interactive approach
- Some sessions will follow the flipped classroom approach
- Practice (Implementations) will be a key factor
- Homework (as code implementations, problems solving, research) are included

VI- Assessment Methods

The assessment of the student is carried out according to the following grid:

Assessment Type	Final Grade Weighting	%	Dates/Deadlines
Participation + class interaction	/10	40% of the final grade	Continuous + session 13
30 mns test	/20		Session 16
Assignments	Assignment 1 /15 Assignment 2 /15		Session 5 – session 9 Session 29 – session 34
Project	/40		Session 10 – session 34
Final exam	/100	60% of the final grade	Date: End of the course
Total		100%	



VII- Specific Course Requirements

1- Regulations concerning deadlines for written and applied work submissions

The teacher has the right to:

- Refuse to receive written applied work outside the deadlines.
- Refuse homework/project copies with high percentage of similarities.
- Refuse homework/project not submitted on **MS teams**.
- Give the student additional time after omitting up to 25% of the mark awarded to the concerned activity.

2- Academic measures in case of plagiarism

The student shall be liable to the following penalties in case of any attempt of illicit scientific citation:

- Cancellation of the student's contribution, activity or scientific article and reduction of the final mark.
- If the same offense is reported in other courses, cancellation of the final exam and failure in the relevant course(s).

VIII- Reference Book

- Karumanchi, Narasimha. *Data Structures and Algorithms Made Easy: 700 Data Structure and Algorithmic Puzzles*. CreateSpace, 2011.

IX- Bibliography & Additional Sources

- Weiss, Mark Allen. *Data structures and algorithm analysis in Java*. Vol. 2. Reading, MA, USA: Addison-Wesley, 1999.