HOLY SPIRIT UNIVERSITY OF KASLIK

Course Syllabus

School of Engineering

I. Course Information (based on catalogue)

Code: GIN321	Type: ☒ C ☐ CTP ☐ TP ☐ P ☐ TD Category: ☐ Math and Basic Sciences ☑ Engineering (☑ Contains Significant Design) ☐ General Education ☐ Other
Title: Algorithmics	
Number of credits: 3	Number of contact hours per week: 3
Pre-requisites: GIN231	Co-requisites:
Delivery Language: ☐ Fr. ☒ Eng. ☐ Ar.	

II. Course Core Curriculum

Course Description *(Slight introduction)*: Asymptotic notation, time and space complexities; Solving recurrences; Trees: traversing methods, balanced trees (AVL and red-black trees), heaps; Advanced sorting algorithms, methods of linear sort; Hashing : open and closed hashing ; Graphs: traversal in depth-first and breadth-first, finding of spanning trees and shortest paths; Huffman coding.

Learning Outcomes:

At the end of this course, students will be able to:

1. Recognize the associated algorithm's operations and complexity
2. Design and apply appropriate data structures for solving computing problems
3. Develop computer programs to implement different data structures and related algorithms
4. Demonstrate the ability to design, develop, run, test, debug and implement complex C++ programs that demonstrate understanding and mastery of concepts and techniques learned in the course

Description of Delivery Mode: All lessons conducted in a classroom environment exposed on slide show and/or on blackboard with mandatory attendance.
Student is provided with print based or electronic course materials which cover all theory subjects and practical applications.

Design activities (if exists):

Design and apply appropriate data structures for solving computing problems

Course Timetable ☐ **per Session** ☒ **per Week**
(Provide course topics)

1. Recall the asymptotic analysis: the time complexity (resp. space) of iterative and recursive algorithms. Introduction to templates in C++

2. Recursion. Divide and conquer. Dynamic programming.
Recall trees: representation, implementation, traversal methods. , Binary search trees (BST).

3. Heaps : min-heap and max-heap
4. The traditional sorting algorithms. Advanced sorting algorithms.
5. Advanced sorting algorithms (continued). Linear sorting algorithms. **<test-1>**
6. Hashing, Collision resolution. Open hashing.
7. Closed Hashing (Open Addressing): linear probing, quadratic probing, double hashing.
8. Huffman coding **<test-2>**
9. Balanced BST: AVL trees **<**Implementation project assignment**>**
10. Balanced BST: RBT (red-black trees)
11. Indexing methods: B-tree and B+-tree.
12. Graphs: representation, directed and undirected graphs, without circuit, weighted graph. Implementation by adjacency matrix and adjacency list.
13. Graphs: depth-first and the breadth-first traversals, shortest path finding algorithms: Dijkstra and Floyd.
14. The minimum spanning tree: Kruskal's algorithm and Prim's algorithm. **<**Implementation project presentation**>**
15. Final examination

III. Course Grading

Grading Criteria (Total = 100%)	
10%	Attendance and participation
10%	Homework, project, research paper
40%	Quizzes, Tests, Midterm
40%	End of semester evaluation (e.g. Final exam)

IV. Course Material

Required Texts	- Practical Introduction to Data Structures and Algorithm Analysis (C++ Edition), *2nd Edition , Clifford Shaffer , Prentice Hall.* *-Teacher materials*
Supplemental References	-Introduction to algorithms, *T. Cormen, C. Leiserson, R. Revest and C. Stein, 3rd edition,* The MIT Press -Algorithm Design: Foundations, Analysis, and Internet Examples, *M. Goodrich and R. Tamassia, Wiley*
Required Materials	*Microsoft Visual C++*