Course Title:	**Algorithm and data structure**
Course Code:	CSE131
Program:	Master Degree In Computer Engineering
Department:	Computer Engineering
Course coordinator:	Dr. Amina GHARSALLAH
Institution:	Private Higher School of Engineers of Gafsa (ESIP)

A. Course identification

1. Credit hours: **4.5 (3-1.5-0)**

2. Course type

a. College ☐ Department ■ Others ☐

b. Fundamental ■ Transversal ☐ Optional ☐

3. Level/year at which this course is offered: 1.1/3

4. Pre-requisites for this course (if any): Logical Mathematics, Simple data structures, Basic Programming Knowledge, Fundamental Mathematics, Basic computer organization

1. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Self-study	Total workload
1	**Traditional classroom**	…..	43	110.5
2	**Blended**	67.5		
3	**E-learning**	.....		
4	**Distance learning**	.....		
5	**Other ()**	.....		

2. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	**Lecture**	45
2	**Laboratory/Studio**	……
3	**Tutorial**	22.5
4	**Others** (specify)	…..
	Total	67.5

B. Course Objectives and Learning Outcomes

Course Description

This course covers fundamental concepts in algorithm design and data structures, including basic logic, control structures, arrays, functions, recursion, and object-oriented programming. Students will learn advanced data structures such as linked lists, stacks, queues, trees, and graphs, as well as searching, sorting, and algorithm complexity analysis.
Through hands-on practice in CSE132 (Programming Workshop) using C++, students will apply these concepts to real-world problems. By the end of the course, they will be able to design efficient algorithms, implement data structures, and develop structured programs to solve complex computing challenges.

This course aims to:

- ✓ Introduce basic algorithm concepts, including problem-solving steps, control structures, and data representation.
- ✓ Build a strong understanding of data structures, such as arrays, lists, stacks, queues, trees, and graphs, for efficient data handling.
- ✓ Teach how to improve algorithm efficiency, covering searching, sorting, recursion, and complexity analysis.
- ✓ Strengthen problem-solving and coding skills by applying algorithms and data structures to real-world problems.
- ✓ Improve teamwork and communication through structured algorithm design, documentation, and coding projects, with hands-on practice in CSE132 (Programming Workshop).

1. Course Learning Outcomes

CLOs		Aligned PLOs
1	**Knowledge and understanding**	
1.1	✓ Understand fundamental algorithmic concepts, control structures, and data representation.	PLO.K1
1.2	✓ In-depth knowledge of advanced algorithms, including sorting algorithms, search algorithms, graph traversal algorithms, and the ability to design efficient algorithms to solve complex problems.	
1.3	✓ The ability to design and implement algorithmic solutions to solve complex computational problems, taking into account time constraint.	
2	**Skills**	
2.1	✓ Communicate technical concepts effectively through structured code documentation and algorithmic problem-solving discussions.	PLO.S2
	✓ Work collaboratively in programming environments, demonstrating problem-solving and teamwork skills in software development projects.	
2.2	✓ The ability to present research results related to algorithms and data structures in a professional manner, whether at academic conferences, industry meetings, or other research forums.	

CLOs		Aligned PLOs
	✓ Apply data structures such as arrays, linked lists, stacks, queues, and trees to solve computational problems. ✓ Implement and analyze searching and sorting algorithms to improve program efficiency. ✓ Design recursive algorithms and understand their application in problem-solving. ✓ Develop structured and modular programs using functions and parameter passing techniques	PLO.S7

C. Course Content

No	List of Topics	Contact Hours
1	**Part 1:** Introduction to Algorithms and Basic Concepts 1. Introduction to Algorithms ▪ Definitions, Characteristics, and Problem-Solving Steps ▪ Constants, Variables, and Algorithm Structure 2. Data Types and Expressions ▪ Simple Data Types: Logical, Character, Integer, and Real ▪ Operators, Expressions, and Operator Precedence 3. Control Structures ▪ Sequential Operations (Assignment, Input/Output) ▪ Conditional Structures (If-Else, Nested Conditions, Multiple Choice) ▪ Iterative Structures (For, While, Repeat-Until Loops)	12
2	**Part 2:** Data Structures and Function-Based Programming 1. **Arrays** ▪ Declaration, Access, and Basic Operations ▪ Searching (Sequential and Dichotomous) ▪ Multidimensional Arrays 2. **Sub-Programs** ▪ Functions and Procedures ▪ Calling Mechanisms and Scope of Variables ▪ Parameter Passing Modes	12.5
3	**Part 3:** Advanced Data Structures 1. Pointers & Dynamic Memory 2. Sorting & Searching Algorithms 3. Recursion 4. Abstract Data Types (ADTs: Lists, Stacks, Queues)	15
4	**Part 4:** Algorithm Optimization and Problem-Solving Strategies 1. Summary of Key Concepts	5.5

No	List of Topics	Contact Hours
	2. Problem-Solving and Algorithm Optimization	

No	List of Topics	Contact Hours
1	Tutorial 1: Conditional and Iterative Structures	5
2	Tutorial 2: List, Array, Dictionary, Tuple, Set	5
3	Tutorial 3: Recursion, Stacks, Queues Tutorial 4: Tree Data Structure Tutorial 5: Priority Queues and Heap Trees	8
4	Tutorial 6: Sorting and Graphs	4.5
Total		22.5

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	**Knowledge and understanding**		
PLO.K1	✓ Understand fundamental algorithmic concepts, control structures, and data representation. ✓ In-depth knowledge of advanced algorithms, including sorting algorithms, search algorithms, graph traversal algorithms, and the ability to design efficient algorithms to solve complex problems. ✓ The ability to design and implement algorithmic solutions to solve complex computational problems, taking into account time constraints.	• Lecturing • Tutorial • Problem-Based Learning • Project-Based Learning	Assignments, Homework Quizzes, Exams,
2.0	**Skills**		
PLO.S2	✓ Communicate technical concepts effectively through structured code documentation and algorithmic problem-solving discussions. ✓ Work collaboratively in programming environments, demonstrating problem-solving and teamwork skills in software development projects. ✓ The ability to present research results related to algorithms and data structures	• Lecturing • Tutorial • Problem-Based Learning • Project-Based Learning	Assignments, Homework Quizzes, Exams, Report,

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
	in a professional manner, whether at academic conferences, industry meetings, or other research forums.		
PLO.K1, PLO.S7	✓ Apply data structures such as arrays, linked lists, stacks, queues, and trees to solve computational problems. ✓ Implement and analyze searching and sorting algorithms to improve program efficiency. ✓ Design recursive algorithms and understand their application in problem-solving. ✓ Develop structured and modular programs using functions and parameter passing techniques		

2. Assessment Tasks for Students

	Assessment task*	Week Due	Percentage of Total Assessment Score
1	Practical Work (written or oral)	Weekly	00%
2	Quizzes, Homework assignments	Random	10%
3	First mid Term	8	25%
4	Final Exam	16	65%

E. Student Academic Counselling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice:
1- Office hours 2- Blackboard interface

F. Learning Resources and Facilities

1. Learning Resources

Required Textbooks	1. Goldschlager, Lister. Informatique et Algorithmique. Prentice-Hall International, *Interéditions, 1986. ISBN: 2-7296-0127-9.* 2. Wirth, Niklaus. Algorithms & Data Structures. Prentice-Hall International Editions, *1986. ISBN: 0-13-021999-1.*

	3. Aho, Alfred, John Hopcroft, & Jeffrey Ullman. *Structures de Données et Algorithmes. Addison-Wesley Europe, Interéditions, 1987. ISBN: 2-7296-0194-5.* 4. Michael T. Goodrich, Roberto Tamassia, & Michael H. Goldwasser – *Data Structures and Algorithms in Python*, Wiley, 2013 5. Lecture Notes for Data Structures and Algorithms – School of Computer Science, University of Birmingham, UK, 2019
Essential References Materials	NA
Electronic Materials	1. Runestone Academy – Problem Solving with Algorithms and Data Structures 2. MIT OpenCourseWare (OCW) – Introduction to Algorithms 3. Khan Academy – Algorithms and Data Structures Tutorials
Other Learning Materials	NA

2. Facilities Required

Item	**Resources**
Accommodation	**Classroom board**
Technology Resources	**Data projector**

G. Course Quality Evaluation

Evaluation Areas/Issues	**Evaluators**	**Evaluation Methods**
Effectiveness of teaching and assessment.	Students, course coordinator, Alumni, Employers	Direct/Indirect
Extent of achievement of course learning outcomes.	Faculty, Program Leaders, quality department	Direct
Quality of Learning resources	Faculty, Program Leaders	Direct, Indirect
Teaching and learning quality and effectiveness.	Students, Faculty Program Leaders	Direct, Indirect

H. Specification Approval Data

Council / Committee	Computer Engineering Council
Date	11/09/2023