Fakultät für Ingenieurwesen
Facoltà di Ingegneria
Faculty of Engineering

SYLLABUS
COURSE DESCRIPTION YEAR 2024/2025

COURSE TITLE	**Data Structures and Algorithms**
COURSE CODE	76243
SCIENTIFIC SECTOR	INF/01
DEGREE	Bachelor in Computer Science
SEMESTER	1st
YEAR	2nd
CREDITS	9

TOTAL LECTURING HOURS	60
TOTAL LAB HOURS	30
ATTENDANCE	Attendance is not compulsory, but strongly recommended. The lectures consist of presentations on the black board, interspersed by small exercises, and discussions with the students. The goal of the course is to enable students to develop and analyze algorithms, which is a skill that can only be acquired by training. All the material used in the lectures and labs as well as the assignments will be published on the OLE pages of the course. Students should note that slides and hand-written lecture notes are supporting material, but their study is not sufficient to reach the goal of the course. Experience shows that some students are able to acquire these intended skills without attending all lectures or all labs, but attendance and success in studies are strongly correlated. Students who are unable to follow all lectures and labs are encouraged to attend at least some of them. They are also encouraged to work out all the exercises given during the lectures and the labs and to submit the coursework, for which they will receive feedback and marks.
PREREQUISITES	• Java programming skills at an introductory level • Basic mathematical knowledge about sets, functions, and elementary calculus
COURSE PAGE	https://ole.unibz.it/

SPECIFIC EDUCATIONAL OBJECTIVES	• Type of course: "caratterizzante" • Scientific area: "discipline informatiche" By following this course, students will be able to formulate algorithmic problems and to recognize algorithmic problems underlying an application.

	They will also acquire an in-depth understanding of the standard data structures and the corresponding algorithmic techniques to solve such problems. They will recognize how certain algorithmic approaches depend on the choice of a suitable data structure and vice versa. Moreover, students will learn how to analyze whether an algorithm is correct and which time and space resources it needs. Finally, students will learn how to compare different algorithms with respect to their suitability for a given application.

LECTURER	Werner Nutt
SCIENTIFIC SECTOR OF THE LECTURER	INF/01
TEACHING LANGUAGE	English
OFFICE HOURS	Friday, 15:30-17:00, by previous appointment Office No: 2.09, Faculty of Computer Science, Piazza Domenicani 3 E-mail: Werner.Nutt@unibz.it
TEACHING ASSISTANTS	Dalmonte Tiziano
OFFICE HOURS	Wednesday, 14:00-15:00, by previous appointment Office No: 2.02, Faculty of Computer Science, Piazza Domenicani 3 E-mail: Tiziano.Dalmonte@unibz.it
LIST OF TOPICS COVERED	• Design Principles: Problem reduction via recursion • Searching and Sorting • Correctness: Loop invariants, termination • Complexity: Asymptotic analysis • Divide and Conquer • Pointers, dynamic data structures, linked lists • Abstract data types: stacks, queues, priority queues, maps • Binary trees, red-black trees • Graph Search
TEACHING FORMAT	• Frontal lectures • Lab groups supported by teaching assistants (TAs) • Biweekly coursework assignments that are corrected and commented by the TAs. In the lectures, new concepts and techniques are introduced, both by way of presentation on the blackboard and by small exercises. In the assignments, students refine these in order to apply them to selected problems. They also measure the actual performance of their implementations and compare it with the theoretical predictions. In the lab groups, students discuss possible approaches to the tasks of the assignments with the TAs and compare different solutions. In addition, students also solve problems that are independent of the assignments to deepen the understanding of the material presented in the lectures.

LEARNING OUTCOMES	**Knowledge and understanding** • Know the concepts of complexity of algorithms and data structures

	• Have a solid knowledge of the most important data structures and programming techniques • Have a solid knowledge of the most important algorithms for sorting and searching and their complexity **Applying knowledge and understanding** • Be able to analyze and measure size, complexity and critical aspects of algorithms and data structures **Ability to make judgments** • Be able to collect useful data about the performance of algorithms and to judge which algorithm is most suitable for a given task **Communication skills** • Be able to structure and write scientific documentation **Ability to learn** • Be able to learn cutting edge IT technologies and their strengths and limitations

ASSESSMENT	The assessment is based on a written final exam, a mock exam, and coursework assignments. The written exam consists of questions to verify knowledge, questions that assess the ability to apply knowledge acquired in the course, and small exercises. The mock exam allows students to familiarize themselves with the exam situation and to understand the kind of skills they are expected to acquire during the course. The assignments consist of exercises to apply knowledge acquired in the lectures and experiments, on which the students have to report. Passing the written exam is mandatory. The coursework and the mock exam are optional (for the weighting and calculation of final mark, see below). The marks are valid during the three exam sessions following the teaching of the course.
ASSESSMENT LANGUAGE	English
EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS	The assessment is based on • coursework assignments (45%) • a mock exam (5%) • a written final exam (50%). To pass the course, the written exam has to be passed. In the written exam, students have to apply techniques taught in the course in a defined setting and have to develop algorithms for new problems. The algorithms developed have to be analyzed with respect to correctness and efficiency. The answers are marked according to their correctness, the suitability of the algorithms developed, and the validity and clarity of the analysis. In the coursework exercises students have to develop solutions for algorithmic problems and analyze their solutions with respect to correctness and running time. The exercises are assessed according to correctness and efficiency and validity of the analysis. In experiments, students have to implement variants of algorithms and identify under which conditions which variant performs best. The experiments are assessed according to the suitability of the design of the experiment, the appropriateness of the measurements taken, and the validity of the conclusions drawn.

	Students who do not submit all assignments or do not take part in the mock exam will be assessed on the written exam taken and the submitted parts of the coursework. For students who take the mock exam and submit all assignments, the final mark will be a weighted average of the exam mark (50%), the mock exam mark (5%) and the assignment mark (45%). If students do not submit all assignments or do not take the mock exam, the percentage for assignments and mock exam will be lower. Also, assignments for which the mark is lower than the mark of the written exam will not be considered. The same holds for the mock exam.

REQUIRED READINGS	Textbook: *Introduction to Algorithms,* Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein (CLRS), 2nd or 3rd edition University Library: ST 134 C811
SUPPLEMENTARY READINGS	*Algorithms and Data Structures - The Basic Toolbox*, K. Mehlhorn and P. Sanders, free download from http://www.mpi-inf.mpg.de/~mehlhorn/ftp/Mehlhorn-Sanders-Toolbox.pdf
SOFTWARE USED	Java