unibz
Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science

# SYLLABUS
# COURSE DESCRIPTION

| | |
|---|---|
| **COURSE TITLE** | Data Structures and Algorithms |
| **COURSE CODE** | 75003 |
| **SCIENTIFIC SECTOR** | INF/01 |
| **DEGREE** | Bachelor in Computer Science and Engineering |
| **SEMESTER** | 2nd Semester |
| **YEAR** | 1st |
| **CREDITS** | 8 |

| | |
|---|---|
| **TOTAL LECTURING HOURS** | 48 |
| **TOTAL LAB HOURS** | 24 |
| **PREREQUISITES** | • Java programming skills at an introductory level<br>• Basic mathematical knowledge about sets, functions, and elementary calculus |
| **COURSE PAGE** | http://www.inf.unibz.it/~nutt/Teaching/DSA1415/ |

| | |
|---|---|
| **SPECIFIC EDUCATIONAL OBJECTIVES** | • Type of course: "di base" for L-31 and L-08<br>• Scientific area: "formazione informatica di base" for L-31 and "matematica, informatica e statistica" for L-8<br><br>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, office POS 2.09 Faculty of CS, POS Building, piazza Domenicani 3<br>werner.nutt@unibz.it<br>+39 0471 016126<br>Lecture's page:<br>http://www.inf.unibz.it/~nutt/ |
| **SCIENTIFIC SECTOR** | INF/01 |

| | |
|---|---|
| **OF THE LECTURER** | |
| **TEACHING LANGUAGE** | English |
| **OFFICE HOURS** | Will be announced |
| **TEACHING ASSISTANT** | TBD |
| **OFFICE HOURS** | |
| **LIST OF TOPICS COVERED** | • Sorting algorithms<br>• Correctness and complexity of algorithms<br>• Divide and conquer algorithms<br>• Pointers and dynamic data structures<br>• Abstract data types<br>• Algorithms for linked lists and trees<br>• Hashing<br>• Graph implementations and graph algorithms |
| **TEACHING FORMAT** | Frontal lectures, exercise groups supported by teaching assistants (TAs), and weekly coursework assignments that are corrected and commented by the TAs. In the lectures, new concepts and techniques are presented. 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 exercise groups, students discuss possible approaches to the tasks of the assignments with the TAs and compare different approaches taken. In the exercise groups, students also solve small problems that are independent of the assignments to deepen the understanding of the material presented in the lectures |

| | |
|---|---|
| **LEARNING OUTCOMES** | **Knowledge and understanding:**<br>• Know the concepts of complexity of algorithms and data structures<br>• Have a solid knowledge of the most important data structures and programming techniques<br>• Have a solid knowledge of the most important algorithms for sorting and searching and their complexity<br>**Applying knowledge and understanding:**<br>• Be able to analyze and measure size, complexity and critical aspects of algorithms and data structures<br>**Ability to make judgments**<br>• Be able to collect useful data and to judge information systems and their applicability<br>**Communication skills**<br>• Be able to structure and write scientific documentation<br>**Ability to learn**<br>• Be able to learn cutting edge IT technologies and their strengths and limitations |

| | |
|---|---|
| **ASSESSMENT** | The assessment is based on a final written exam, the coursework |

| | |
|---|---|
| | assignments, and a midterm exam. Students who do not submit assignments will be assessed on the exams alone.<br>For students who submit all assignments, the final mark will be a weighted average of the exam mark (50%), the midterm mark (10%) and the assignment mark (40%). If students do not submit all assignments or do not take the midterm, the percentage for assignments and midterm will be lower. Also, assignments for which the mark is lower than the mark of the written exam will not be considered.<br>The assignment marks are valid during the three exam sessions following the teaching of the course. |
| **ASSESSMENT LANGUAGE** | English |
| **EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS** | In the exam, students have to show the ability to apply concepts and skills learned in the course to small sample problems. They have to develop algorithmic solutions for new algorithmic problems and analyze their solutions with respect to correctness and running time. They also have to explain their choice of data structure and algorithmic technique |

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