# COMPUTATION II

SYLLABUS
2023-2024

| | |
|---|---|
| INSTRUCTOR INFORMATION | JOÃO FONSECA<br>jpfonseca@novaims.unl.pt<br>joaopfonseca.github.io |
| SCHEDULE | TP1:<br>• Wednesday 8:30h – 10:00h<br>P1:<br>• Wednesday 11:30h – 13:00h<br>• Thursday 8:30h – 10:00h<br>P2:<br>• Wednesday 10:00h – 11:30h<br>• Friday 8:30h – 10:00h<br>P3:<br>• Thursday 14:00h – 15:30h<br>• Friday 10:00h – 11:30h<br>P4:<br>• Wednesday 14:00h – 15:30h<br>• Thursday 11:30h – 13:00h |
| OFFICE HOURS | Friday 14:00h – 15:00h (schedule appointment by email)<br>Room 137 (CAN) |
| CONTACT | Students should use the Moodle platform for any communication with the course instructor. Homework and project submissions should also be done through Moodle. |
| DESCRIPTION | The objective of this course is to provide students with fundamental concepts, methods and tools available to write more efficient and effective computer programs. Computation II focuses on the foundational concepts required for a more complete understanding of software development. It is assumed that the students have attended the curricular unit Computation I. This course aims to bridge practical and theoretical knowledge with in-class examples and a group project. The course is mostly practical and is taught in Python using Jupyter notebooks. |
| OBJECTIVES | At the end of the course, students should be able to:<br>• Analyze and implement algorithms and data structures efficiently, considering their time and space complexities.<br>• Apply mathematical foundations to analyze algorithm efficiency and solve computational problems effectively.<br>• Design and implement recursive algorithms, understanding their advantages and limitations compared to iterative approaches.<br>• Evaluate and utilize various searching and sorting algorithms, comprehending their complexities and suitability for different scenarios.<br>• Construct and manipulate abstract data structures and trees. |
| COURSE SUCCESS | In this course, success depends on a number of factors:<br>• A priori knowledge of Python programming; |

| | |
|---|---|
| | • Class attendance;<br>• Work during the semester and not only when exams are about to start;<br>• Develop the course project during the semester, making the most of the practical classes;<br>• Complete exercises provided in class and consistent practice;<br>• Read the suggested references and materials made available by the lecturer. |
| CONTENTS | 1. Introduction:<br>a. Fundamental concepts: problem, algorithm, program<br>b. The need for effective and efficient data structures: cost-benefit analysis<br>2. Mathematical foundations for understanding computation:<br>a. Logarithms, summations, and product<br>b. Recurrences and quantifiers<br>c. Free and bound variables<br>3. Introduction to recursion and time complexity analysis:<br>a. Recursive functions and algorithms<br>b. Recursion vs iteration<br>c. Time complexity and memory occupation: best, worst and average cases<br>d. Algorithm complexity<br>4. Searching algorithms<br>5. Sorting algorithms<br>6. Data structures and dedicated algorithms:<br>a. Simple and complex data type<br>b. Abstract data structures<br>c. Dynamic data structures<br>d. Linear dynamic data structures<br>e. Trees |
| BIBLIOGRAPHY | References:<br>❑ Data Structures and Algorithms with Python (2015), by Kent D. Lee and Steve Hubbard, in Springer Publishing<br>❑ Python Data Structures and Algorithms (2017), by Benjamin Baka, in Packt Publishing Ltd<br>❑ Problem Solving with Algorithms and Data Structures Using Python (2011, 2nd edition), by Bradley N. Miller, David L. Ranum, in Franklin, Beedle & Associates<br><br>**Note:** all references are available at NOVA IMS library or are provided by the teacher. |
| STUDENT EVALUATION | 1st and 2nd Call – 65% written exam, 35% practical group-project (including project discussion)<br>**Minimum grade of 9.5 is required for both evaluation components.** |

## Coursework

**The practical group project** will account for 35% of the student's final grade. It consists of the application of all contents discussed in the course. The students will develop an analysis of a fictional dataset using the data structures and algorithms discussed throughout the course. The full description of the project guidelines, goals, requirements and deadlines will be announced on lecture 4. The end product of the project should be your code, fully documented. With this project, students are expected to develop their proficiency in Python programming, as well as understand software development best practices.

**Project discussion**: after submitting the projects, students will be called to discuss them with the instructors. This will be one of the main sources of evaluation.

**Groups' composition**: the project can be done individually or in groups (the latter is a better option). The groups cannot exceed 3 students.

**Project Deadline:** June 11$^{th}$ (tentative)

**In-class behavior:** Active and positive participation is essential to ensure personal and group learning. As such, the teaching faculty reserves the right to apply penalties to the individual students' final grade for poor in-class behavior, at their discretion.

**Tasks:** In both, practical and theoretical classes, students will be frequently assigned homework, which will consist of simple tasks related with the course material. It is expected that the students complete these tasks.

**Final Exam (1$^{st}$ and 2$^{nd}$ call evaluation):** The written exams (both calls) will be in-class exam covering all the materials of the course.

**Note:** Any form of copying, cheating, or plagiarism, as well as usage or possession of any kind of electronic device during any of the examination epochs will result into the student failing the course in all examination epochs of the current academic year.