



# FOUNDATIONS OF COMPUTER SCIENCE A - Co

ING-INF/05 - 9 CFU - Annual Tuition

## Teaching Staff

### VINCENZA CARCHIOLO

**Email:** vincenza.carchiolo@unict.it

**Office:** Edificio 3 - piano V - stanza 11

**Phone:** 095 7382359

**Office Hours:** martedì e venerdì dalle ore 11 alle ore 12

---

## LEARNING OBJECTIVES

### Knowledge and understanding abilities

- Basic knowledge of computer architecture
- Knowledge of the principles of computer science and procedural programming
- Knowledge of the main data structures and understanding of the operating mechanisms
- Knowledge of search and sorting techniques
- Knowledge of the rudiments of computational complexity

### Applying knowledge and understanding abilities

- Ability to develop programs in ANSI C to solve problems using the most important data structures (stacks, lists, queues)
- Ability to analyze the code and correct errors during the development phases
- Ability to solve problems and defining "problem solving" algorithms
- Ability to use a development environment (IDE)

### Ability of making judgements

- The student is able to evaluate the most suitable algorithms to solve a given problem

### Communication skills

- The student acquires knowledge of computer languages and technical terminology

## **COURSE STRUCTURE**

The course is organized into elementary didactic units based on the contents and skills to be developed.

The course includes lectures as the main teaching method. Lectures are accompanied by exercise sessions are designed with the aim of acquiring the basic theoretical knowledge and all the syntactic elements, acquiring the ability to solve problems, applying the knowledge, using development environments and methodologies.

The teacher also proposes individual exercises that consist in solving a problem that the student is asked to face with autonomously and subsequently solved and discussed in class.

Should teaching be carried out in mixed mode or remotely, it may be necessary to introduce changes with respect to previous statements, in line with the programme planned and outlined in the syllabus.

Learning assessment may also be carried out on line, should the conditions require it.

---

## **DETAILED COURSE CONTENT**

### **Module 1**

- Information processing and algorithms - Algorithms and programs - A graphic notation for algorithms representation - Programming languages - Programs design.
- Information representation: Numerical systems - Numerical systems conversion - Binary number system - Binary number operations - overflow and underflow - Integers representation - Numbers with sign representations - Fixed-point and floating-point representation - Characters representation - Boole's Algebra, Logical Functions, Logical Expression, Applications of Boolean algebra.
- Computer organization overview: computer organization: main memory, central unit, instruction set architecture.
- Basics on computer systems: basic software: Translation and program execution - programming environment - Programming languages: imperative languages - Compilers, Linkers, Implementation briefs: Preprocessor - Comments - Libraries and header files.

### **Module 2**

- C language fundamental elements: C syntax - Structure of a C program - Compiling a program - Types of data and representations - Main data types - Identifiers - Variables - Access modifiers - Constants - Operators - Control structures - Selection instructions - Iteration instructions - Jump instructions - Expression instructions - Block statement.
- Arrays, strings and pointers: One-dimensional arrays - Array pointers - Arrays as a function argument - Strings - String arrays - Multidimensional arrays - Pointer variables - Operators and expressions with pointers - Array pointers - Function pointers.

- Functions: Function visibility rules - Function arguments - Main arguments - Return statement - Return values from a function - Recursion - Identifiers declarations and field of action - Parameter binding techniques - side effects and function implementation.
- User-defined structures, unions and types: Structures - Arrays of structures - Structures as arguments of functions - Pointers to structures - Arrays and structures within other structures - Union - Enumerations - Sizeof - Typedef.
- I/O from file and console: Read and write characters and console strings - Console-formatted I/O - Channels - File.

### **Module 3**

- Dynamic memory allocation.
- Computational Complexity: Program Efficiency, Notations O and W, Program complexity evaluation.
- Sorting algorithms: Sorting algorithm classes - sorting by selection (selection sort) - insertion algorithms (insertion sort) - bubble sort exchange algorithms, quick sort, merge sort.
- Abstract data types: Lists, Queues, Stacks, Binary Trees.

---

### **TEXTBOOK INFORMATION**

[Be] Belagurusamy. Programming in ANSI C McGraw-Hill

---