

Computer Programming 1

Introduction to the course

Welcome! ... *the university*

- University of Trento
- Department of Information Engineering and Computer Science (DISI)
- Undergraduate Courses - curriculum in English
 - Computer Sciences
 - Computer, Communications and Electronic Engineering (ICE)
- Academic Year: 2025-2026



of Information Engineering and Computer Science

egree in

Computer, Communications and Electronic Engineering

Home

Computer, Communications and Electronic Engineering

PROSPECTIVE STUDENT

Course programme

Course content - curriculum in Italian

Course content - curriculum in English

Application

The Bachelor's Degree Programme in Computer, Communications and Electronic Engineering combines three major disciplines in the Information Engineering class (L8) offered in various Italian universities (Computer Engineering, Information and Communications Engineering, Electronic Engineering) and from the 2022/2023 academic year also provides **a curriculum entirely taught in English**.

The curriculum taught in Italian "Ingegneria Informatica, delle Comunicazioni ed Elettronica" after a basic and cross-disciplinary training between the 3 areas, allows you to choose a **specific curricular path between Computer, Communications, Electronic**.

[Course content: curriculum in Italian](#)

The curriculum exclusively taught in English "Computer, Communications and Electronic Engineering" offers basic training in the 1st year

Welcome! ... *the course*

- Course: Computer Programming 1 (ENG)
 - Compulsory course
 - Schedule:
 - Period: first semester of 2025 – 2026
 - Lesson period: September – December 2025
 - Exams: Jan. & Feb. 2026, Jun. & Jul. 2026, Sept. 2026
 - Academic hours:
 - lecture (~45%)
 - laboratory (~55%)
 - (Course&Exam) Modality: in presence
 - 12 credits (ECTS)

Filter by

| Area of study | Course Language | Access mode |
|---|--------------------|-------------|
| Engineering, Computer Science, and Techni ▼ | Inglese italiano ▼ | Programmed |

× Engineering, Computer Science, and Technologies × Inglese italiano × Programmed

[2] BACHELOR'S DEGREES

BACHELOR'S DEGREE

Computer Sciences

Area: Engineering, Computer Science, and Technologies
Durata: 3 years
Crediti: 180
Lingua: English Italian
Accesso: Programmed
Posti disponibili: Local programmed (235 Places)

BACHELOR'S DEGREE

Computer, Communication and Electronic Engineering

Area: Engineering, Computer Science, and Technologies
Durata: 3 years
Crediti: 180
Lingua: English Italian
Accesso: Programmed
Posti disponibili: Local programmed (235 Places)

Welcome! ... *the course*

- Course: Computer Programming 1 (ENG)
 - Compulsory course
 - Schedule:
 - Period: first semester of 2025 – 2026
 - Lesson period: September – December 2025
 - Exams: Jan. & Feb. 2026, Jun. & Jul. 2026, Sept. 2026
 - Academic hours:
 - lecture (~45%)
 - laboratory (~55%)
 - (Course&Exam) Modality: in presence
 - 12 credits (ECTS)

Filter by

| Area of study | Course Language | Access mode |
|---|--------------------|-------------|
| Engineering, Computer Science, and Techni ▼ | Inglese italiano ▼ | Programmed |

× Engineering, Computer Science, and Technologies × Inglese italiano × Programmed

[2] BACHELOR'S DEGREES

BACHELOR'S DEGREE

Computer Sciences

Area: Engineering, Computer Science, and Technologies
Durata: 3 years
Crediti: 180
Lingua: English Italian
Accesso: Programmed
Posti disponibili: Local programmed (235 Places)

BACHELOR'S DEGREE

Computer, Communication and Electronic Engineering

Area: Engineering, Computer Science, and Technologies
Durata: 3 years
Crediti: 180
Lingua: English Italian
Accesso: Programmed
Posti disponibili: Local programmed (235 Places)

Credits: 180

- ➔ Courses: 12 or 6 credits
- ➔ Credits for other activities: thesis, etc

12 CFU x 25 hours/CFU = 300 hours

- Lectures, classes, and in-class activities
- Individual study
- Other activities, if required

Welcome! ... *the course*

- Course: Computer Programming 1 (ENG)
 - Compulsory course
 - Schedule:
 - Period: first semester of 2025 – 2026
 - Lesson period: September – December 2025
 - Exams: Jan. & Feb. 2026, Jun. & Jul. 2026, Sept. 2026
 - Academic hours:
 - lecture (~45%)
 - laboratory (~55%)
 - (Course&Exam) Modality: in presence
 - 12 credits (ECTS)

[146140] - COMPUTER PROGRAMMING 1

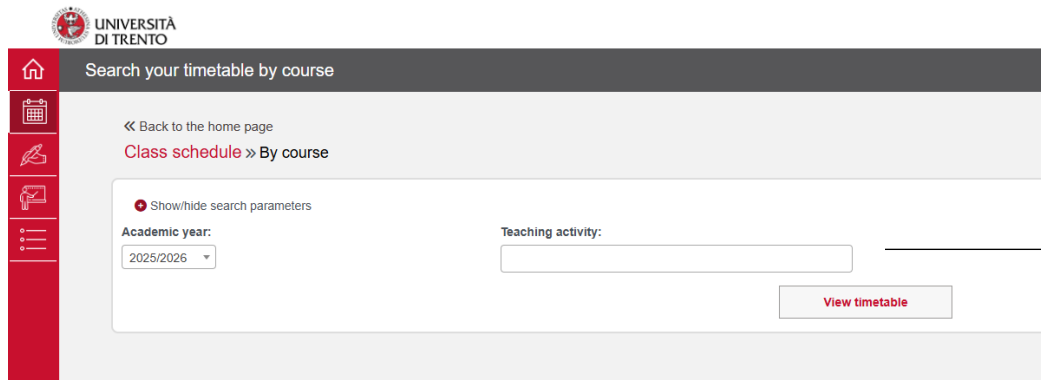
① Attività erogata in mutuaione da [146140] Computer Programming 1 del corso COMPUTER, COMMUNICATIONS AND ELECTRONIC ENGINEERING

| | |
|--|--|
| General information | |
| Course | COMPUTER SCIENCE |
| Curriculum | Computer Science |
| Course type | Bachelor's degrees |
| Academic year | 2025/2026 |
| Year | 1 |
| Training activity type | Basic |
| Scope | Formazione Informatica |
| Language | ENGLISH |
| CFU | 12 CFU |
| Didactic Activity Type | Lecture |
| Teaching period | First semester (from 10/09/2025 to 23/12/2025) |
| Holders | MARCHETTO ALESSANDRO |
| Length | 96 hours (96 hours Lecture) |
| Didactic method | Convenzionale |
| Subject area | ING-INF/05 |
| Course objectives and learning outcomes | |
| Entrance requirement | |
| Contents | |
| Teaching and learning methods and activities | |
| Test and assessment criteria | |
| Bibliography/Study materials | |

Link to the Course [Syllabus](#)

“A syllabus is a detailed course outline and roadmap, outlining the course's content, objectives, assignments, policies, grading criteria, and schedule for the academic term. The syllabus is a guide and a contract between the instructor and students, providing essential information to help students understand expectations”

Course schedule



UNIVERSITÀ DI TRENTO

Search your timetable by course

« Back to the home page

Class schedule » By course

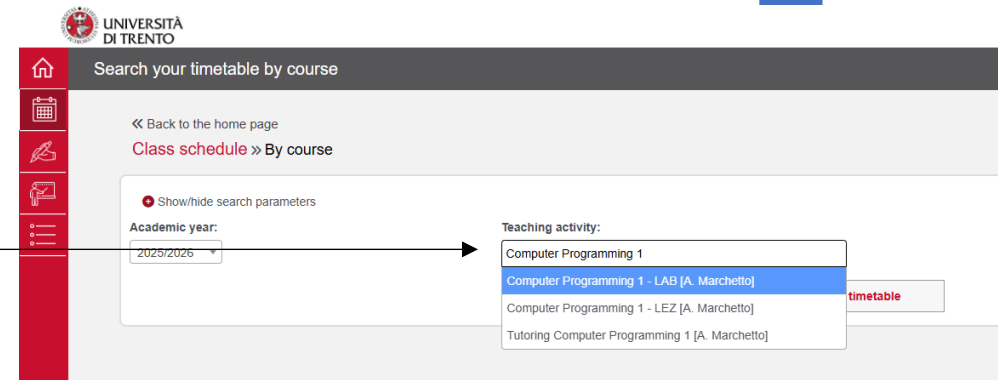
Show/hide search parameters

Academic year: 2025/2026

Teaching activity:

View timetable

Link to the Course [Schedule](#)



UNIVERSITÀ DI TRENTO

Search your timetable by course

« Back to the home page

Class schedule » By course

Show/hide search parameters






Academic year: 2025/2026

Teaching activity:

- Computer Programming 1
- Computer Programming 1 - LAB [A. Marchetto]
- Computer Programming 1 - LEZ [A. Marchetto]
- Tutoring Computer Programming 1 [A. Marchetto]

timetable

Course schedule

| Day | Time | Location and room | |
|-----------|---------------|----------------------|---|
| Monday | 10.30 – 12.30 | Povo 1, room B109 |  Lecturers  Laboratories   |
| Tuesday | 12.30 – 14.30 | Povo 2, room B106 pc | |
| Wednesday | 10.30 – 13.30 | Povo 2, room B106 pc | |
| Thursday | 15.30 – 17.30 | Povo 1, room A105 | |
| Tuesday | 8.30 – 10.30 | Povo 1, room A201 pc |  Tutoring (from Sept.23, 2025) |

Course schedule

| Day | Time | Location and room |
|-----------|--------------------------------|----------------------|
| Monday | 10.30 – 12.30 | Povo 1, room B109 |
| Tuesday | 12.30 – 14.30 | Povo 2, room B106 pc |
| Wednesday | 10.30 – 13.30 | Povo 2, room B106 pc |
| Thursday | 15.45 15.30 – 17.30 | Povo 1, room A105 |
| Tuesday | 8.30 – 10.30 | Povo 1, room A201 pc |

Lecturers

Laboratories

Tutoring
(from Sept.23, 2025)

Unitn Esse3



ESSE3
servizi *online*
per la didattica



Search exams sessions

Insert one or more search criteria and click on "Search"

| | |
|--------------------------|--------------|
| Faculties | Select |
| Course | Select |
| Teaching activity | Select |
| Professor | Select |
| | (gg/mm/aaaa) |
| Exam date | |

Search

Insert one or more search criteria and click on "Search"

| | |
|--------------------------|---|
| Faculties | [03D3] Dipartimento di Ingegneria e Scienza dell'Informazione (DIP) |
| Course | Select |
| Teaching activity | Select |
| Professor | Select |
| | (gg/mm/aaaa) |
| Exam date | |

Search

Link to the UniTN Esse3

- <https://www.esse3.unitn.it>

- *Check the information about the exams, e.g., days, rooms (when they will be published)*
- *Register for the exam session you plan to attend*

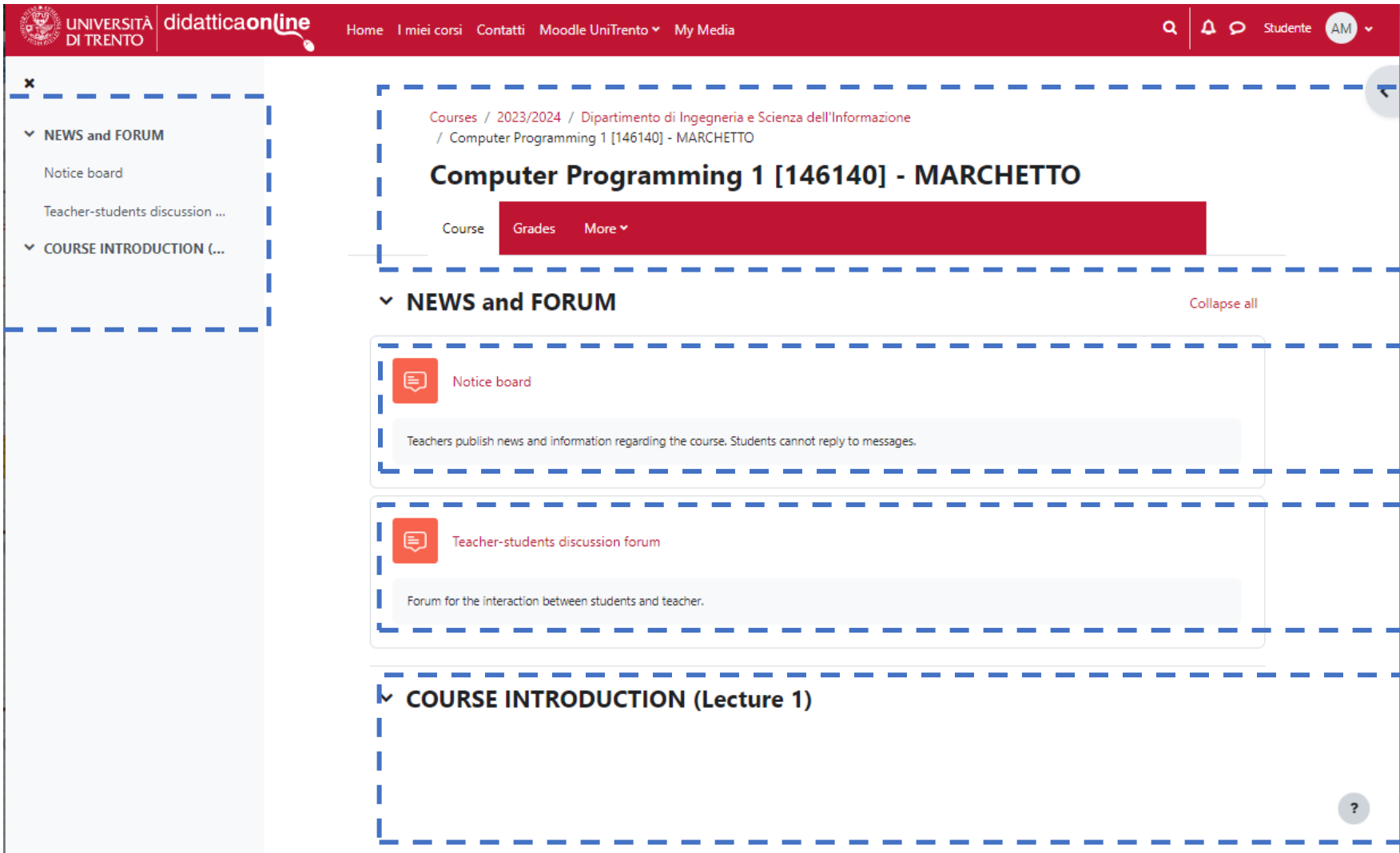
Unitn Moodle

Moodle platform of the course

- <https://didatticaonline.unitn.it/dol/course/view.php?id=41888>

- Remember **to register yourself** in the course on the Moodle platform for the specific course!!!
- News and communication about the course and lectures
- Course material (slides, exercises, source code)

Unitn Moodle



The screenshot shows the Moodle interface for the course 'Computer Programming 1 [146140] - MARCHETTO'. The page is divided into several sections, each highlighted with a dashed blue box and labeled on the right:

- Sections:** A sidebar on the left containing a list of course sections, including 'NEWS and FORUM' and 'COURSE INTRODUCTION (...)'. It is labeled 'Sections' on the left.
- The course:** The main header area of the course page, displaying the course title 'Computer Programming 1 [146140] - MARCHETTO' and navigation tabs like 'Course', 'Grades', and 'More'. It is labeled 'The course' on the right.
- News and messages from teachers:** A section titled 'NEWS and FORUM' containing a 'Notice board' where teachers publish news and information. It is labeled 'News and messages from teachers' on the right.
- Forum:** A section titled 'Teacher-students discussion forum' for interaction between students and teachers. It is labeled 'Forum' on the right.
- Material (Slides, source code, etc.):** A section titled 'COURSE INTRODUCTION (Lecture 1)' for course materials. It is labeled 'Material (Slides, source code, etc.)' on the right.

Who we are (1/2)



Alessandro Marchetto

- Associate Professor at UniTN
- ~10y experience in industry
- ~10y researcher in academia (universities & research centers)
- Ph.D. Computer science



Who we are (2/2)

Adolfo Villafiorita



- >10y Head of Research Unit at FBK
- >10y contract professor in Universities
- PhD Computer Science

Michele Bof



- Master Degree in Computer Science @UniTN
- Senior developer

Co-founder of Shair.Tech, a startup developing technologies for reducing (food) waste and helping make our distribution chain fairer, more sustainable, and circular.

Who we are (2/2)

Adolfo Villafiorita



Michele Bof



- >10y Head of Research Unit at FBK
- >10y contract professor in Universities
- PhD Computer Science

- Master Degree in Computer Science @UniTN
- Senior developer

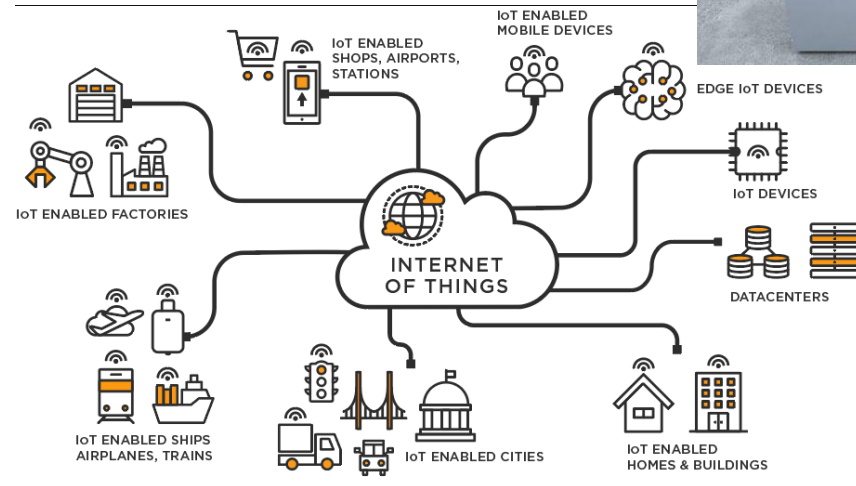
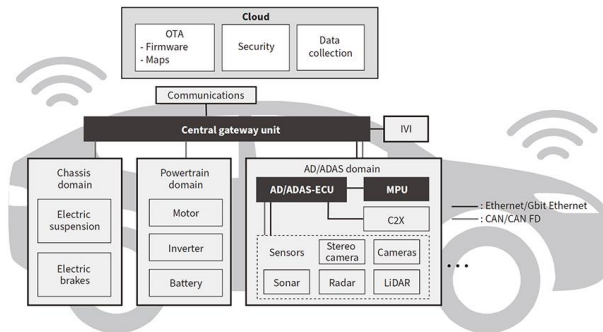
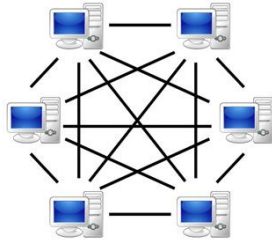
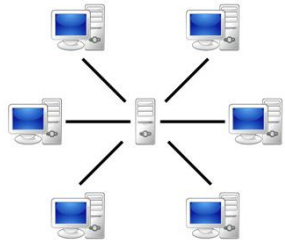
Co-founder of Shair.Tech, a startup developing technologies for reducing (food) waste and helping make our distribution chain fairer, more sustainable, and circular.

Tutor #1: ? and Tutor #2: ?

** pending but approved

** already scheduled (from Sept.23, 2025)

Context: Computer and Communications



Informatics and Computer Science (1)

In general, terms are used as synonyms even if they are not the same

Informatics (*Information processing science*)

- It is the study of computation, information, and automation
- It studies the representation, processing, and communication of information in natural and engineered systems
- It involves several scientific and technical aspects concerning the collection and processing of information in the various sectors of science, technology, economic, social, and even practical activities

Computer Science

- It mainly refers to the application to engineering, science, law, medicine, ...
- It involves theoretical disciplines: algorithms, information theory, ...
- It is applied to practical disciplines: the design and implementation of software
- It is closely related to computer programming

Informatics and Computer Science (2)

Example: software engineering

Informatics

- Design and optimization of the information flow of a system (e.g., management of a customer care), by considering different user roles and their responsibility

Computer Science

- Develop the platform underlying a system (e.g., management of a customer care)
- Definition of an efficient ordering algorithm (for reducing time and complexity)
- Optimization of language compiler for speed up the program execution

Informatics and Computer Science (3)

Example: AI

Informatics

- Develop a new application that uses AI architectures and learning algorithms to analyze pandemic data and help in making decisions (objective: understand how the data can be collected and analyzed to improve the decisions)

Computer Science

- Develop a new AI architecture or learning algorithm, to improve the capability of working with huge amount of data

Course objective



The main objective of the course is *to acquire basic knowledge on:*

- Computer
- Algorithms
- Programming language (one in particular)
- Programming

Syllabus



Introduce the course in terms of:

- Learning objectives
- Expected learning outcomes
- Content
- Method
- Material
- Assessment method and criteria

[146140] - Computer Programming 1

SAVE PDF

| General information | | ▼ |
|--------------------------|--|---|
| Course | COMPUTER, COMMUNICATION AND ELECTRONIC ENGINEERING | |
| Curriculum | Computer, Communications and Electronic Engineering | |
| Academic year | 2023/2024 | |
| Training activity type | Basic compulsory subjects | |
| CFU | 12 CFU | |
| Didactic Activity Type | Lecture | |
| Evaluation | Voto Finale | |
| Teaching period | First semester | |
| Holders | MARCHETTO ALESSANDRO | |
| Teachers | VILLAFIORITA MONTELEONE ADOLFO , BOF MICHELE | |
| Length | 108 hours (108 hours Lecture) | |
| Frequency | Not mandatory | |
| Subject area | ING-INF/05 | |
| Location | Polo di collina - Povo B - via Sommarive, 9 | |
| Goals | | ➤ |
| Required skills | | ➤ |
| Subjects | | ➤ |
| Teaching methods | | ➤ |
| Verification of learning | | ➤ |
| Books | | ➤ |

Learning Objectives and Outcome

Introductory class on programming, programming principles and techniques

Learning Objectives

- Introduce algorithms, programming principles and techniques.
- Make the student familiar with code programming and provide the basic notions, insights and tools of code programming
- Learn fundamentals of C++ imperative programming
- Learn basic/introductory notions of object-oriented programming in C++
- Learn program structures, elements, and operations
- Learn by doing: learn how to develop programs to face problems of simple/average difficulty

Expected outcome

The student:

- (1) will know fundamental notions and concepts about (C++) imperative programming;
- (2) will be able to analyze problems of simple/average difficulty;
- (3) will be able to identify and code solutions to average difficult problems by using C++ as imperative programming language.

Learning Objectives and Outcome: FAQs

Is this a course on advanced programming? NO

Is this a course to learn how to program network / mobile / web apps? NO

Can I develop the programs in my own way ? NO

Can I use my personal devices, IDEs, tools, programming language, style? NO

So what?

- This is an introductory class on programming
- The main objective is to provide all the knowledge to model, solve and develop software program to simple problems.

Coding is not programming (1)

Coding

Computers can understand machine code that is difficult to understand for humans.

Coding is the process of writing into a programming language so that the computers can understand what it needs to execute.

Coding is writing lines of code in the programming language for creating a computer program (software).

Coding is not programming (1)

Coding

Computers can understand machine code that is difficult to understand for humans.

Coding is the process of writing into a programming language so that the computers can understand what it needs to execute.

Coding is writing lines of code in the programming language for creating a computer program (software).

Programming

Coding is only a phase of a software development.

Several phases are needed: *planning, design, testing, deployment, maintenance*.

Programming is a complex activity which includes not only coding, but also other activities, e.g., to design, analyze and implement algorithms, understand data structures, and solve problems.

To write code, you need to be transparent with the schema or structure of the program: e.g., write the pseudocode to describe the logic of the algorithm to be implemented and use it for explaining the algorithm to a programmer (coder).

Coding is not programming (2)



| | Coding | Programming |
|--|--------|-------------|
|--|--------|-------------|

*If Programming is the process of writing a whole book.
Then Coding is just about writing a single chapter of the book.*

<https://hackr.io/blog/coding-vs-programming-difference-you-should-know>

Coding is not programming (2)

| | Coding | Programming |
|-------|--|--|
| Scope | It is a process in which a set of instructions are converted into a language that is comprehensible for a computer | Other than coding, it concerns the definition of the requirements, problem solving, pseudocode writing, algorithm thinking, optimizing, testing, executable code creation, maintenance |

*If Programming is the process of writing a whole book.
Then Coding is just about writing a single chapter of the book.*

<https://hackr.io/blog/coding-vs-programming-difference-you-should-know>

Coding is not programming (2)

| | Coding | Programming |
|-------|--|--|
| Scope | It is a process in which a set of instructions are converted into a language that is comprehensible for a computer | Other than coding, it concerns the definition of the requirements, problem solving, pseudocode writing, algorithm thinking , optimizing, testing, executable code creation, maintenance |
| Skill | As a coder, a good knowledge of syntax and semantics of the selected programming language is required | As a programmer, additional skills are required with respect to the ones of the coder, e.g., capability of level thinking and of problem analysis |

*If Programming is the process of writing a whole book.
Then Coding is just about writing a single chapter of the book.*

<https://hackr.io/blog/coding-vs-programming-difference-you-should-know>

Course Content (not ordered)



Content by topics

Course Content (not ordered)



Content by topics

Basic code programming concepts

- Algorithms and PC architectures
- Programming phases and languages
- Introduction to (imperative) C++

Course Content (not ordered)



Content by topics

Basic code programming concepts

- Algorithms and PC architectures
- Programming phases and languages
- Introduction to (imperative) C++

Variables and predefined types

- Representation, variables and constants
- Data type (integers, booleans, characters)

Course Content (not ordered)



Content by topics

Basic code programming concepts

- Algorithms and PC architectures
- Programming phases and languages
- Introduction to (imperative) C++

Variables and predefined types

- Representation, variables and constants
- Data type (integers, booleans, characters)

Control structure

- Basic instructions
- Expressions and assignments
- Structured instructions (sequences, conditions, cycles)

Course Content (not ordered)

Content by topics

Basic code programming concepts

- Algorithms and PC architectures
- Programming phases and languages
- Introduction to (imperative) C++

Variables and predefined types

- Representation, variables and constants
- Data type (integers, booleans, characters)

Control structure

- Basic instructions
- Expressions and assignments
- Structured instructions (sequences, conditions, cycles)

Input/Output

- I/O standard
- I/O on files (argc & argv)

Course Content (not ordered)

Content by topics

Basic code programming concepts

- Algorithms and PC architectures
- Programming phases and languages
- Introduction to (imperative) C++

Variables and predefined types

- Representation, variables and constants
- Data type (integers, booleans, characters)

Control structure

- Basic instructions
- Expressions and assignments
- Structured instructions (sequences, conditions, cycles)

Input/Output

- I/O standard
- I/O on files (argc & argv)

Advanced data types and memory management

- Array
- Ordered array
- Array multidimensional (matrices)
- Strings and texts
- Pointers and references
- Structure (struct)
- Dynamic memory allocation
- Dynamic allocation of array and struct
- Fundamental data structure (lists, stacks, and queues)
- Binary Trees and Graphs

Course Content (not ordered)

Content by topics

Basic code programming concepts

- Algorithms and PC architectures
- Programming phases and languages
- Introduction to (imperative) C++

Variables and predefined types

- Representation, variables and constants
- Data type (integers, booleans, characters)

Control structure

- Basic instructions
- Expressions and assignments
- Structured instructions (sequences, conditions, cycles)

Input/Output

- I/O standard
- I/O on files (argc & argv)

Advanced data types and memory management

- Array
- Ordered array
- Array multidimensional (matrices)
- Strings and texts
- Pointers and references
- Structure (struct)
- Dynamic memory allocation
- Dynamic allocation of array and struct
- Fundamental data structure (lists, stacks, and queues)
- Binary Trees and Graphs

Program organization and structure

- Structure and organize a program on more files
- Functions and parameter passing
- Recursive functions
- Classes (introduction)

Course Content (not ordered)

Content by topics

Basic code programming concepts

- Algorithms and PC architectures
- Programming phases and languages
- Introduction to (imperative) C++

Variables and predefined types

- Representation, variables and constants
- Data type (integers, booleans, characters)

Control structure

- Basic instructions
- Expressions and assignments
- Structured instructions (sequences, conditions, cycles)

Input/Output

- I/O standard
- I/O on files (argc & argv)

Advanced data types and memory management

- Array
- Ordered array
- Array multidimensional (matrices)
- Strings and texts
- Pointers and references
- Structure (struct)
- Dynamic memory allocation
- Dynamic allocation of array and struct
- Fundamental data structure (lists, stacks, and queues)
- Binary Trees and Graphs

Program organization and structure

- Structure and organize a program on more files
- Functions and parameter passing
- Recursive functions
- Classes (introduction)

Advanced topics: program testing and debugging, generative programming*(depending on the time)*

Teaching method



Prerequisites

- This is an introductory class on programming
- The objective is to provide all the knowledge to model, solve and develop software program to simple problems. Hence, there are no specific pre-requisites to address the course.

Teaching method

- Lectures for the theoretical part: introduction of notions, principles, tools, and examples of problems and programs
- Laboratory exercises mainly for practicing with C++ programming (learning by doing)
- Tutoring Labs: tutoring sessions will be conducted in the laboratory to support the students while learning (tutoring by other students, and tutoring by teachers)

Material

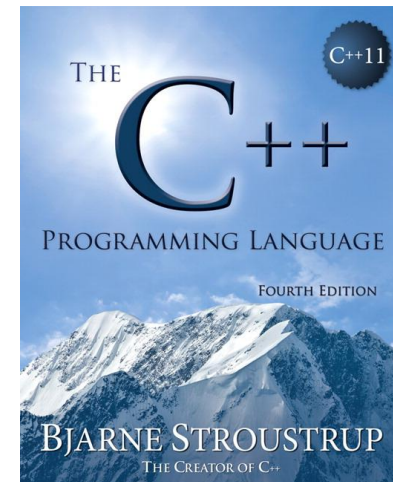
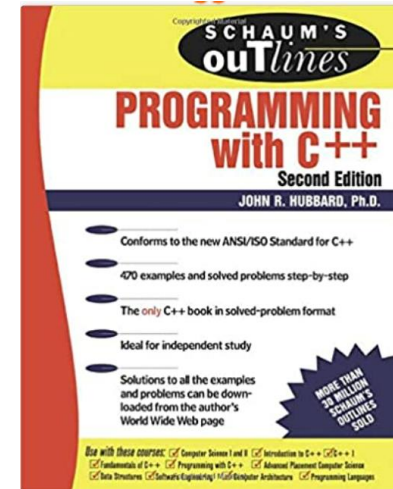
- Slides and additional documents will be available on the course web site
- Examples of code programs, the ones used in both lectures and laboratories, will be available on the course web site as well as additional examples will be also made available
- Textbooks

(Alternatives) Reference for programming with C++

- John R. Hubbard. "Schaum's Outline Of Programming With C++ (2ed)
McGraw Hill, ISBN: 9780071353465
<https://www.mhprofessional.com/schaum-s-outline-of-programming-with-c-9780071353465-usa>
- Bjarne Stroustrup "Programming: Principle and Practice Using C++",
Pearson - Addison-Wesley, ISBN: 9780321992789 (2nd ed.)
<https://www.pearson.com/uk/educators/higher-education-educators/program/Stroustrup-Programming-Principles-and-Practice-Using-C-2nd-Edition/PGM1087672.html?tab=overview>

(Alternatives) Reference for the C++ language (optional):

- Bjarne Stroustrup "The C++ Programming Language"
Pearson – Addison-Wesley, ISBN: 9780321563842 (4 a ed.)
<https://www.pearson.com/uk/educators/higher-education-educators/program/Stroustrup-C-Programming-Language-The-4th-Edition/PGM948798.html>
- Herbert Schildt "C++: The Complete Reference"
McGraw Hill, ISBN: 9780071502399 (4 a ed.)
<https://www.mhprofessional.com/9780072226805-usa-c-the-complete-reference-4th-edition-group>



Assessment methods and criteria

Exam sessions:

- January 2026,
- February 2026,
- June 2026,
- July 2026,
- September 2026

Time: 2h each session

Each session is composed of two parts

- Questions (closed and open) to assess the acquired theoretical competence
- Exercises to assess the acquired practical competence in terms of C++ code programming

The exam will be done in laboratory by using the laboratory PC and tools (no other material is admitted)

Exam organization and grade

- Exam
 - The grade of the exam is expressed in /30 and it represents the weighted average of the scores of the two parts (40% theory and 60% laboratory)
 - Both the two parts are considered passed when a score of at least 18 is achieved.
- A simulation of the exam session will be organized (towards the of the course).
 - Goal:
 1. understand how the exam is organized,
 2. self-evaluation
 - Not valid as exam and not evaluated
 - To be scheduled (depending on the course): end of November 2025 / Beginning of December 2025

Suggestions (1)

- To attend the lectures and the laboratories
- Please notice that each argument discussed during the lectures and the laboratories could be part of the exam
- Ask if you did not understand something is a good practice, for you and for the other course attendees
- Try to study day-by-day and after each lecture
- Try to execute/re-execute each exercise and examples that have been seen during the lectures
- Try to execute always the proposed exercises and ... implement, implement, implement
- Experience plays a fundamental roles in programming
- Understanding the messages from the compiler and the debugger is crucial for a programmer, the experience helps in understanding them
- ...

Suggestions (2)



- ...
- When programming:
 - Before starting to write the code, **analyze the problem and organize (at least) a mental schema of the solution** to implement
 - Write **simple, clear and readable code** as much as possible
 - Code indentation is strongly recommended!!!
 - Leave the adequate space among code instructions
 - Use meaningful name and terms, by following the conventions (upper case, lower case, etc.)
 - ...
 - Adopt **“coding standards”**:
 - Adopting “coding standard” lead to develop code with a quite clear and “uniform” format, this is important to decrease the code complexity and to increase the code readability, promoting code reusability, maintainability of the code, and identification of errors.
 - Be carefully
 - Deal with programming with humility ... try to avoid: “I’m already able of programming!”
 - If needed, be ready to change your way of working/programming: “I have always done in this way!”

Interactions with teachers

- Ask questions before/during/after the lectures, laboratories, and breaks
- Discuss doubts and solutions with teachers and colleagues (outside the lecture/laboratory time!)
- Ask for 1-to-1 meetings only for clarifications and (personal) problems
- Ask questions via email: try to limit yourself to relevant questions, otherwise we will be not able to answer at all!
 - E.g.,
 - Meeting for clarifications
 - Alerts about mistakes and problems with the material
 - Alerts about problems such as time overlaps
 - Alerts about specific and personal problems of the student (e.g., working student, etc.)
- Tutoring will be activated *soon*
- Please, for any contact with us it is mandatory to use the email address from UniTN
name.surname@studenti.unitn.it

A note on plagiarism

Total or partial plagiarism is forbidden!

- Exam tests (written part)
 - Exercises
 - Programs
 - ...
-
- “I took this text from a colleague of mine” No! ... no reuse of material produced by other colleagues is allowed
-
- In case of plagiarism, you will fail the class.

Copyright and Credits



- The material (slide, text, figures, etc.) is intended solely for students at the University of Trento registered to the relevant course for the Academic Year 2024-2025.
- The material (slide, text, figures, etc.) used in the course is partially inspired to the one used in the past years by Marco Roveri, Giuseppe Riccardi, and Roberto Sebastiani.
- The copyright of the material is held by the authors. Copying, editing, translation, storage, processing or forwarding of content in databases or other electronic media and systems without written consent of the copyright holders is forbidden. The selling of (parts) of this material is forbidden. Presentation of the material to students not involved in the course is forbidden. The unauthorized reproduction or distribution of individual content or the entire material is not permitted and is punishable by law.