

# Unit 0: Overview

## Programming 2

---

Degree in Computer Engineering  
University of Alicante  
2025-2026



**WHO ARE WE?**



**CSE STUDENTS!**



**WHAT ARE OUR  
EXPECTATIONS?**



**CODING GAMES  
AND HACKING!**



**WHAT ARE WE  
ACTUALLY DOING?**



**cout << "HELLO WORLD!";**



- Core subject in the Computer Science Degree
- 6 ECTS credits
- Main (official) policies for the course can be found at:  
<https://cvnet.cpd.ua.es/Guia-Docente/?wlengua=en&wcodasi=34008&scaca=2025-26>
- **Course materials can be found in Moodle via UACloud**
- News and tutoring through UACloud and Moodle
- Follow us on X: @prog2ua

- Coordinator:
  - David Tomás
- Theoretical sessions teachers:
  - Jorge Calvo, Paco Castellanos, Mariano López, David Tomás (ENG, I2ADE) y Antonio Toral (VAL)
- Lab sessions teachers:
  - Gonzalo Alcalá, Pablo Cubillas, Félix Fuentes (ENG), Felicidad García (VAL), Sandra García, Mariano López, Paco Moreno, Juan Antonio Pérez, Roberto Tardío, David Tomás (ENG) y Antonio Toral

- Face to face:
  - In the office or by video conference
  - Office hours available at <http://www.dlsi.ua.es>
  - **Necessary to book an appointment via UACloud**
- Online:
  - Using UACloud (questions are answered as soon as possible)
  - **Do not make enquiries by e-mail**

*"C++ makes it harder to shoot yourself, but when you do it blows your whole leg off"*

Bjarne Stroustrup, C++ creator

- Unit 1. Introduction
- Unit 2. The `string` class
- Unit 3. Files
- Unit 4. Dynamic memory
- Unit 5. Introduction to object oriented programming

# Aims

- To analyse the requirements of a problem
- To design and implement mid-sized programs
- To develop abstraction and generalisation skills
- To properly organise a program using functions and classes
- To identify the most efficient solutions
- To implement software using an adequate and understandable programming style
- To develop criticism in the algorithm verification process
- To use basic programming tools
- To learn basic notions of object oriented programming

# Evaluation (1/3)

- Theory (50% of the final grade):
  - Multiple-choice test in the theory classroom
  - Theoretical concepts and short exercises
  - **Date: May 27, 2026**
- Practical work (50% of the final grade):
  - Three assignments:  $p1$ ,  $p2$ , and  $p3$
  - These account for 30% of the practical grade
  - Practical exam: accounts for 70% of the practical grade
  - **Date: May 27, 2026**
  - **Practical grade =  $0.1 * (p1 + p2 + p3) + 0.7 * exam$**
  - A self-correction tool (*autocorrector*) will be provided before each submission
  - The grade for each assignment will be determined by an automated grader (*corrector*), which will be an expanded version of the *autocorrector*



## Evaluation (2/3)

- To average between theory and practical work, a minimum score of 4 is required in both parts
- The final grade must be equal to or greater than 5 to pass
- Algorithm to calculate the final grade:

```
practicalGrade = 0.1 * (p1 + p2 + p3) + 0,7 * exam;  
  
if (theoryGrade >= 4 && practicalGrade >= 4)  
    finalGrade = 0.5 * theoryGrade + 0.5 * practicalGrade;  
else  
    finalGrade = kFAIL;
```

## Evaluation (3/3)

- In case of failing in June:
  - The practical grade is preserved for July if it is greater than or equal to 4
  - The theory grade is not preserved for July
- July session:
  - Theory and practicals: exam on July 7, 2026
  - Practical work: you must submit all three practical assignments and pass all automated grading tests (*correctores*) to be eligible for the exam. The final grade is calculated based exclusively on the exam score

# Lab sessions guidelines

- Your lab group can be consulted at UACloud
- It is not possible to change your assigned group (if justified reasons such as work or family issues exist, you may request group change at the secretariat of EPS)
- Lab sessions begin: Monday, January 26
- **Assignments are done individually**
- **Attendance is mandatory: maximum 3 unjustified absences**
- It is necessary to strictly follow the instructions in the assessments (especially those regarding the output format of your program or submission instructions)

- The only way to learn to program is by programming
- Assessments are designed so that students learn in a practical way during the semester
- A student who copies (or who gets the assignment done by someone else) rarely learns and passes the course

- Regulation for student evaluation at UA, Article 14.1:  
*“Students must respect the rules about the authenticity of the presented work and its privacy.”*
- Copying an assignment from another student, in whole or in part, violates the Article 14.1 (authenticity)
- Sharing an assignment (so other students could copy it or get inspiration), also violates Article 14.1 (privacy)

- Regulation for students evaluation at UA, Article 14.4:  
*“In any case, fraudulent actions in an evaluation procedure will result in a failing grade, with a numerical score of zero in that test, without prejudice to the disciplinary procedure that could be initiated against the student and, if appropriate, the sanction that could be applied in accordance with the present legislation”*
- Students involved in cheating will get 0 in the corresponding assignment or exam and a report will be sent to EPS for subsequent disciplinary actions

# Using AI in Programming 2

- We know you use **ChatGPT** (or similar AI tools) to help you develop your code
- It is a tool that will be essential in your career as programmers, and it is okay to use it, but...
- **What is allowed with AI in Programming 2?**
  - While programming
    - Obtain explanations about the use of functions
    - Document your code
    - Review code style
  - After finishing your code
    - Create tests
    - Debug/optimize
- **What NOT to do with AI in Programming 2?**
  - Ask it to write your code from scratch by simply providing the instructions...
  - ... because you will not learn...
  - ... **and during the exam, you are not allowed to use AI**

# Temporal planning

Monday	Wednesday	Thursday	Friday	Submission
26/01 U0	28/01 U0	29/01 U0	30/01 U0	-
02/02 U1 (1)	04/02 U1 (1)	05/02 U1 (1)	06/02 U1 (1)	-
09/02 U1 (2)	11/02 U1 (2)	12/02 U1 (2)	13/02 U1 (2)	-
16/02 U1 (3)	18/02 U1 (3)	19/02 U1 (3)	20/02 U1 (3)	<i>p1</i>
23/02 U2	25/02 U2	26/02 U2	27/02 U2	-
02/03 U3 (1)	04/03 U3 (1)	05/03 U3 (1)	06/03 U3 (1)	-
09/03 U3 (2)	11/03 U3 (2)	12/03 U3 (2)	13/03 U3 (2)	-
16/03 U4	18/03 U4	19/03 -	20/03 U4	-
23/03 U5 (1)	25/03 U5 (1)	26/03 U4	27/03 U5 (1)	<i>p2</i>
30/03 Exer.	01/04 Exer.	02/04 -	03/04 -	-
06/04 -	08/04 -	09/04 -	10/04 -	-
13/04 -	15/04 Exer.	16/04 -	17/04 -	-
20/04 U5 (2)	22/04 U5 (2)	23/04 U5 (1)	24/04 U5 (2)	-
27/04 U5 (3)	29/04 U5 (3)	30/04 U5 (2)	01/05 -	-
04/05 U5 (4)	06/05 U5 (4)	07/05 U5 (3)	08/05 U5 (3)	-
11/05 U5 (5)	13/05 U5 (5)	14/05 U5 (4)	15/05 U5 (4)	-
18/05 Exer.	20/05 Exer.	21/05 U5 (5)	22/05 U5 (5)	<i>p3</i>



## In order to pass Programming 2...

- You must practice a lot
- You need to do the theoretical exercises and the assignments
- You cannot pass by starting to study a week before the exam
- Raise your doubts in both theoretical and lab practice sessions
- If you get lost, do in-person or online tutoring