

PRO1: Aprender a Programar en el context de la IA generativa

...

PRO1. Curs 2026-2027

AI coding assistants are here

- AI coding assistants are already very powerful—and getting better quickly (Claude Code, Codex, Kiro, Cursor, Antigravity, Copilot, etc.).
- They are becoming ubiquitous and integrated into most IDEs. You will certainly use them in your professional career.
- Today, they can solve every PRO1 exercise from the Jutge using the problem statement alone.
- This is very tempting. Also **potentially disastrous for learning**.
- But used appropriately, they can become a **valuable learning tool**.

AI in Coding: 10 Years of History

- **2016-2021: AI-Assisted Coding**

- Early models were primarily used for code completion, recommendation, and bug detection.
- Focused on improving developer productivity within existing workflows.

- **2022-2023: Conversational Coding Assistants**

- LLMs enabled natural-language interaction with code.
- Developers could generate, explain, debug, refactor, and document code through chat.

AI in Coding: 10 Years of History

- **2024-2025: Repository-Aware Development**

- AI expands from individual files to understanding entire codebases, documentation, and project structure.
- Enables project-level assistance: navigation, multi-file changes, and large-scale refactoring.

- **2025-2026: Agentic Software Engineering**

- AI evolves from assisting developers to planning and executing multi-step development tasks.
- Agents can edit code, run tests, fix issues, and create pull requests, shifting developers toward architecture, requirements, and validation.

AI in Coding: A Question

**Why should I learn basic programming if
AI can already program for me?**

AI in Coding: Answers

- Studying Computer Science is becoming more important, not less, in the age of AI.
 - AI will dramatically increase the amount of software being created.
 - Generating code becomes cheaper; understanding, validating and maintaining it becomes more valuable.
 - Verification, security, testing and software quality become central engineering skills.
 - To supervise AI in software generation, **you need to understand software.**

AI in Coding: Answers (2)

- Programming remains a core component. AI provides tools which may change how we produce software, but not the fundamental intellectual foundations of CS
- What does not change?
 - Computational thinking
 - Algorithms and data structures
 - Abstraction and decomposition
 - Complexity analysis
 - Debugging and testing
 - Software design
 - Verification and security

AI in Coding: A Practical Question

Okay, so I am a PRO1 student, what should I do about AI coding assistants?

AI in Coding: A Practical Question

Okay, so I am a PRO1 student, what should I do about AI coding assistants?

In short: the goal is **learning**, so you should avoid using AI to replace your cognitive effort, but you can use it to support it.

To AVOID

- Do not ask AI to solve PRO1 problems for you.

This is **why**:

- Delegating the solution means delegating the learning.
- Reading a generated solution can create an **illusion of understanding**: code often looks obvious once you see it.
- Learning programming requires struggling with a problem, making mistakes, debugging them, and iterating.
- AI may produce solutions that use concepts or C++ features outside the scope of PRO1.
- **In the exam, you will need to solve problems independently.**

To AVOID (2)

- **"I can learn everything with AI, so I don't need to attend classes"**
 - AI assistants can be useful learning tools, but they are not a substitute for the structured learning process of the course and the teachers' experience.
 - Theory and lab sessions develop different skills: conceptual understanding, problem solving, programming practice, debugging, and discussion.
 - Ask your teachers questions. Explaining what you do not understand is itself part of learning.
 - Attend classes, be patient and take your time. There are no magic shortcuts to learning programming.

To AVOID (3)

- **Avoid AI-based code completion while learning to program.**
 - Syntax highlighting, indentation and basic editor support are fine.
 - But aggressive AI autocomplete can complete expressions, loops, conditions or entire functions before you have formulated them yourself.
 - In PRO1, writing the code is part of the cognitive exercise.
 - You should be able to formulate and write programs independently i.e., the same conditions you will encounter in the exam.

But AI assistants CAN be useful... as long as you remain the problem solver

- Test question. *Who is doing the thinking?*
 - If AI is doing the work that you are supposed to learn, don't use it.
 - If you have done the thinking and AI helps you understand, test, critique or improve your work, it can be useful.

AI Assistants CAN be Useful: Examples

- **Quick reference and examples**
 - "Remind me of the syntax for passing parameters to a function in C++."
- **Hints, not solutions**
 - "Here is my code. Don't solve the exercise. Give me a hint about what might be wrong."
- **Generate counterexamples / tests**
 - "Can you find an input for which my program fails?"

AI Assistants CAN be Useful: Examples

- **Critique completed code**
 - "My solution passes the Judge. Critique its clarity, structure and efficiency according to the PRO1 correction guide."
- **Debugging** (unlikely in PRO1)
 - "Explain this error message without fixing the program for me."
- Learning more about a topic **once you have completed your personal study** and solved all the problems from the Judge

Warning: Do not ask for help too quickly

- Struggling with a problem is part of the learning process.
- If your solution does not pass the Judge, give yourself time to find the problem.
- If you are still stuck the next day and cannot ask a teacher, use AI for hints, not solutions:
 - Ask it to critique your code, ask for hints about what might be wrong, ask for test cases that prove your code incorrect.

Final thoughts

- **Your learning is ultimately your responsibility.**
- Ask yourself what you want to get from PRO1.
- If your goal is to build solid programming foundations to prepare you for more complex programming courses:
 - **Don't delegate to AI what you are here to learn.**
- Use AI assistants to **support your learning, not replace it.**

Extra Slides

Avoid	Good use
“Solve this problem”	“Give me a hint”
“Write a fix for this error”	“Explain this compiler error”
“Fix my code”	“Find an input where my code fails”
“Give me a better solution”	“Critique my solution”
“Optimize my solution”	“Help me analyze its time complexity”
“Write the algorithm”	“Ask me questions that help me find the algorithm”

Extra Slides: example prompt

- If you are stuck with a problem and you want to engage in a **conversation** with an AI assistant to help your progress:
 - use a prompt that describes the right context for the assistant to work as a **tutor**, not as a problem solver
 - provide the full problem statement, your C++ program (which does not fully work) and eventually the PRO1 coding rubric
 - We are providing a general prompt for you to use (check the PRO1 website)

Extra Slides: example prompt

Estic estudiant primer curs d'Enginyeria Informàtica i necessito ajuda per resoldre un problema de programació.

Actua com un professor de l'assignatura de Programació 1 i ajuda'm a trobar i entendre els errors del meu programa, però **sense resoldre el problema per mi ni proporcionar-me una solució completa**.

Vull que actuis com un tutor. Analitza l'enunciat i el programa que he escrit i guia'm perquè sigui jo qui arribi a una solució correcta.

Segueix aquestes pautes:

- Dona'm **una pista cada vegada**, començant per les menys explícites.
- Sempre que sigui possible, fes-me preguntes que m'ajudin a descobrir l'error per mi mateix.
- Pots proporcionar exemples d'entrada que facin fallar el meu programa i demanar-me que raoni què hauria de passar.
- Pots indicar-me quin tipus d'error hi pot haver (condició incorrecta, cas límit no considerat, bucle mal plantejat, etc.).
- Pots assenyalar aspectes de l'enunciat que potser no estic tenint en compte.
- Pots ajudar-me a seguir l'execució del meu programa pas a pas.
- **No escriguis el programa correcte ni substitueixis el meu codi per una versió corregida.**
- Evita donar-me directament el fragment de codi que corregeix l'error. Ajuda'm a descobrir com corregir-lo.
- Després de cada pista, espera que jo intenti modificar o explicar el programa abans de donar-me una pista més explícita.

T'adjunto en PDF l'enunciat del problema, inclosos els exemples d'entrada i de sortida.

Aquest és el programa en C++ que he escrit i que no funciona correctament:

```
<programa> [codi en C++] </programa>
```

Comença analitzant el meu programa i dona'm **només una primera pista** que m'ajudi a avançar.

Extra: Can the AI Assistant Replace Lab Classes?

Lab sessions offer much more than “practice-at-home-with-AI”:

- Teachers observe the programming process, not only the final solution
- Identify misconceptions students don't know they have, not just answer the questions they ask.
- Guide learning according to the course objectives
- Promote peer learning through discussion, code comparison, and collaborative problem solving.
- Verify genuine understanding by asking students to explain, justify, and modify their own code.
- Structured environment for deliberate 3-hour practice, with dedicated time, instructor support, and regular engagement.