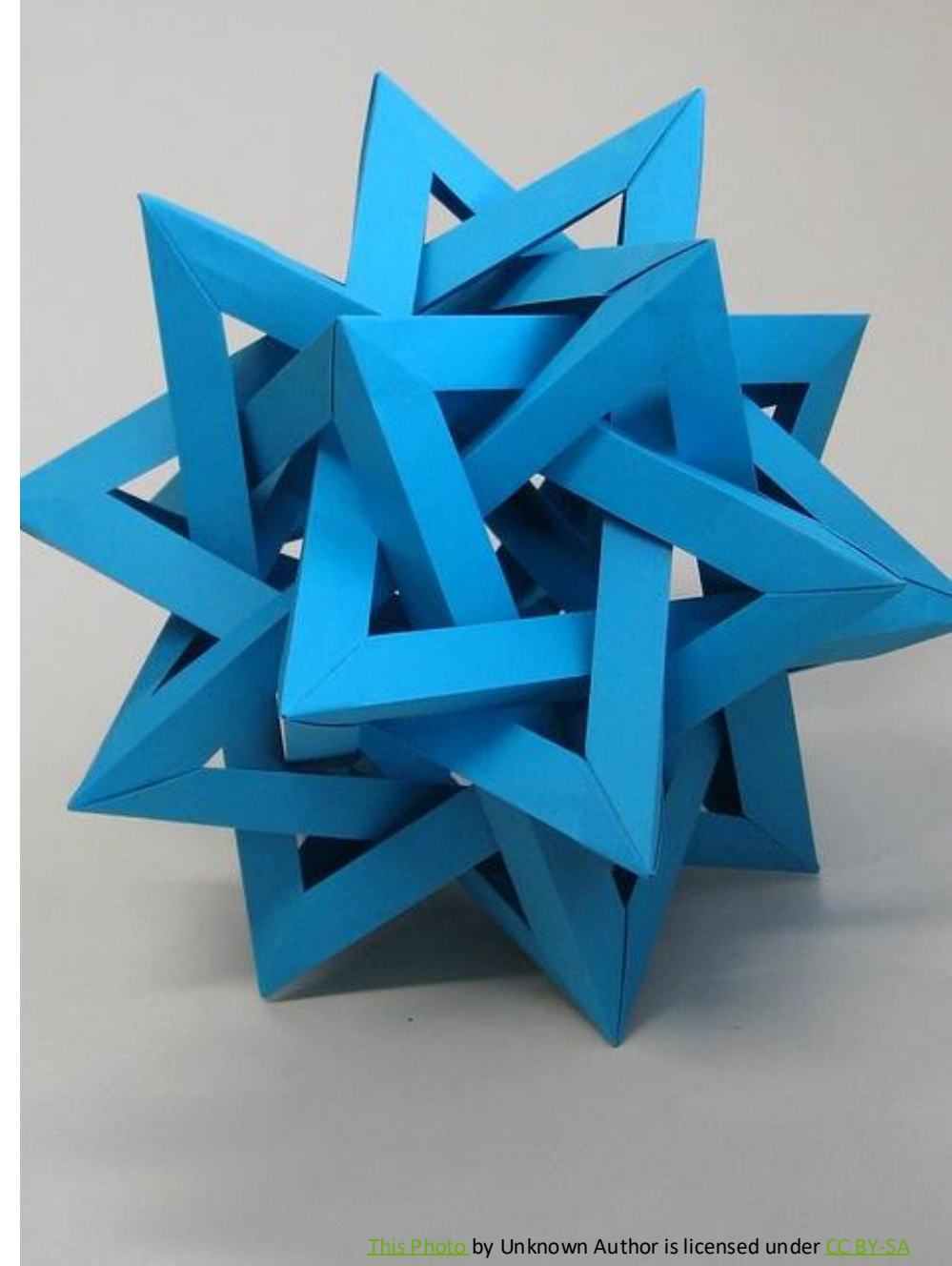**Politecnico di Torino**

Dipartimento
di Automatica e Informatica

1859

Computer Sciences

INTRODUCTION TO THE COURSE

This Photo by Unknown Author is licensed under CC BY-SA

Summary

- Who are we?

- Why are we here?

- How to get out of here?

- … and some practicalities

Who are we?

Computer Sciences (07JCJ*)

- ## Course 2 (FAV-OZJ)

Daniele Jahier Pagliari

Dept. of Automatic Control and Computer Science
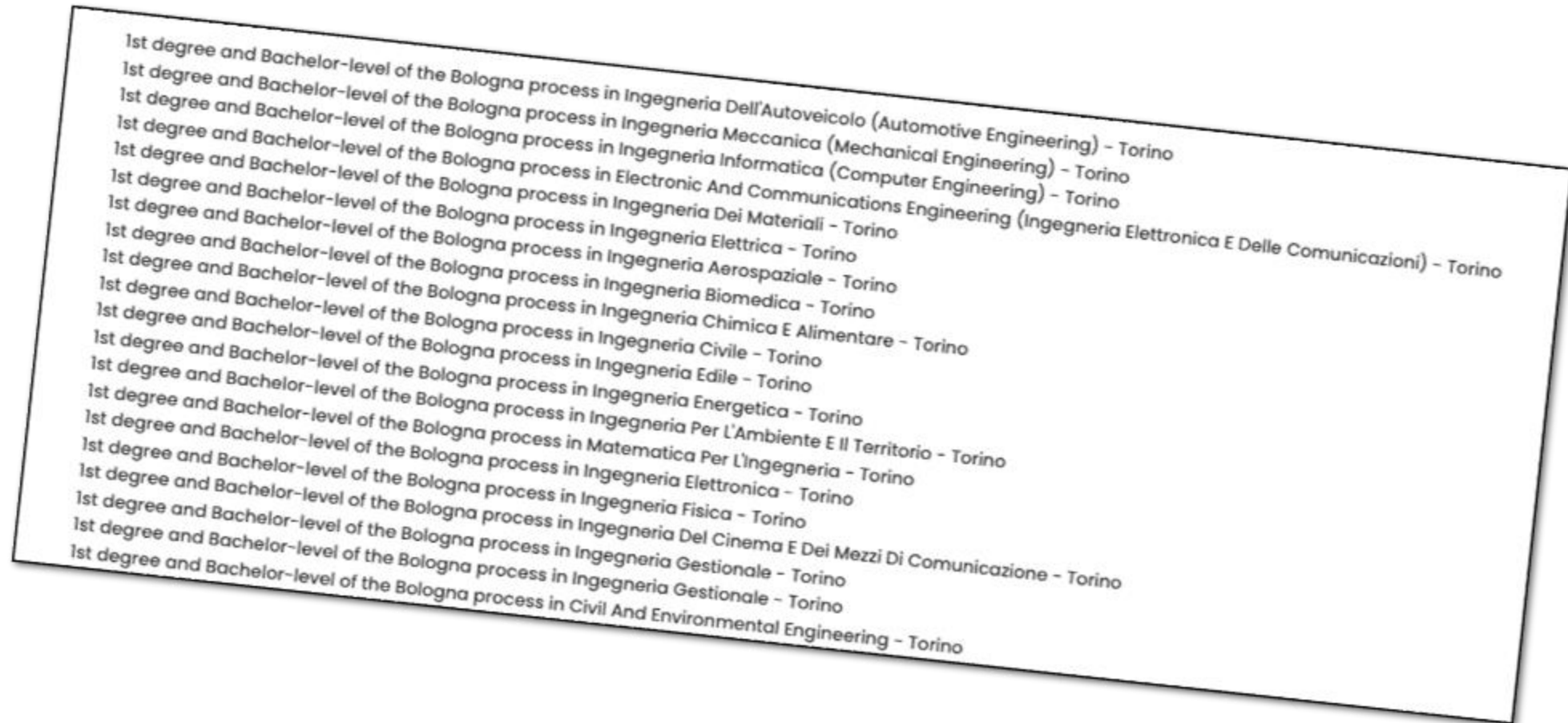
E -mail: **daniele.jahier @ polito.it**

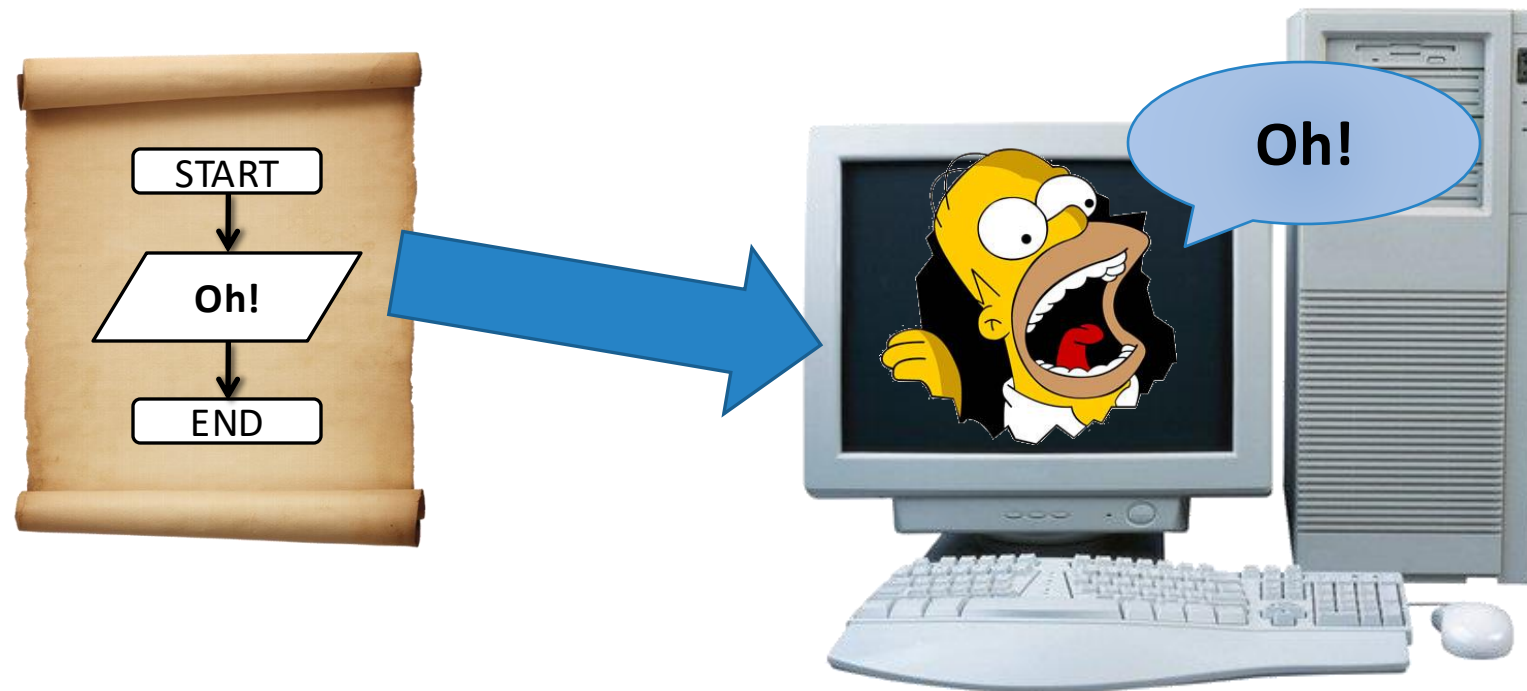**TA: Mohamed Amine Hamdi**

E -mail: **mohamed.hamdi@polito.it**

Who are you?

1st degree and Bachelor-level of the Bologna process in Ingegneria Dell'Autoveicolo (Automotive Engineering) – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Meccanica (Mechanical Engineering) – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Informatica (Computer Engineering) – Torino
1st degree and Bachelor-level of the Bologna process in Electronic And Communications Engineering (Ingegneria Elettronica E Delle Comunicazioni) – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Dei Materiali – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Elettrica – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Aerospaziale – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Biomedica – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Chimica E Alimentare – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Civile – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Edile – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Energetica – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Per L'Ambiente E Il Territorio – Torino
1st degree and Bachelor-level of the Bologna process in Matematica Per L'Ingegneria – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Elettronica – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Fisica – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Del Cinema E Dei Mezzi Di Comunicazione – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Gestionale – Torino
1st degree and Bachelor-level of the Bologna process in Ingegneria Gestionale – Torino
1st degree and Bachelor-level of the Bologna process in Civil And Environmental Engineering – Torino

650+ Students &
2 Instructors

COMPUTER SCIENCES

# Why are we here?

What do we learn in this course?

- From the specification of a problem, to the realization of a solution to that problem, in the form of a computer program

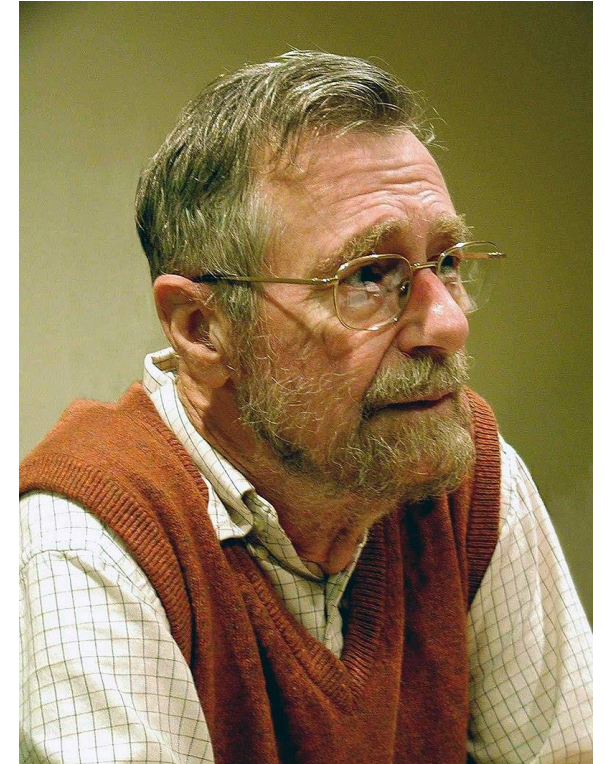## The first ENGINEERING DESIGN course at Politecnico

Welcome to Politecnico

- This is your very first Engineering class

- Engineering is
  - Design
  - Solve problems
  - Finding solutions
  - Meet the specifications
  - Comply with constraints
  - Use the available tools

- Computer Engineering =
  - Any possible type of problems
  - The computer is the **tool**

The greatest difficulty is acquiring **the right mindset** for the engineering approach
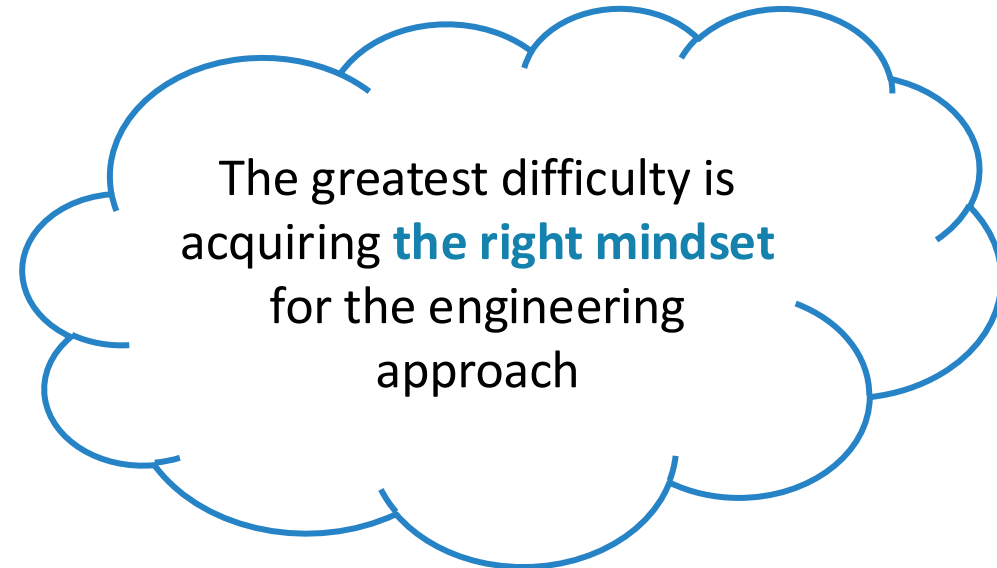
"COMPUTER SCIENCE IS NO MORE ABOUT COMPUTERS THAN ASTRONOMY IS ABOUT TELESCOPES…"

- Attributed to Edsger Dijkstra

…and what do we learn to do?
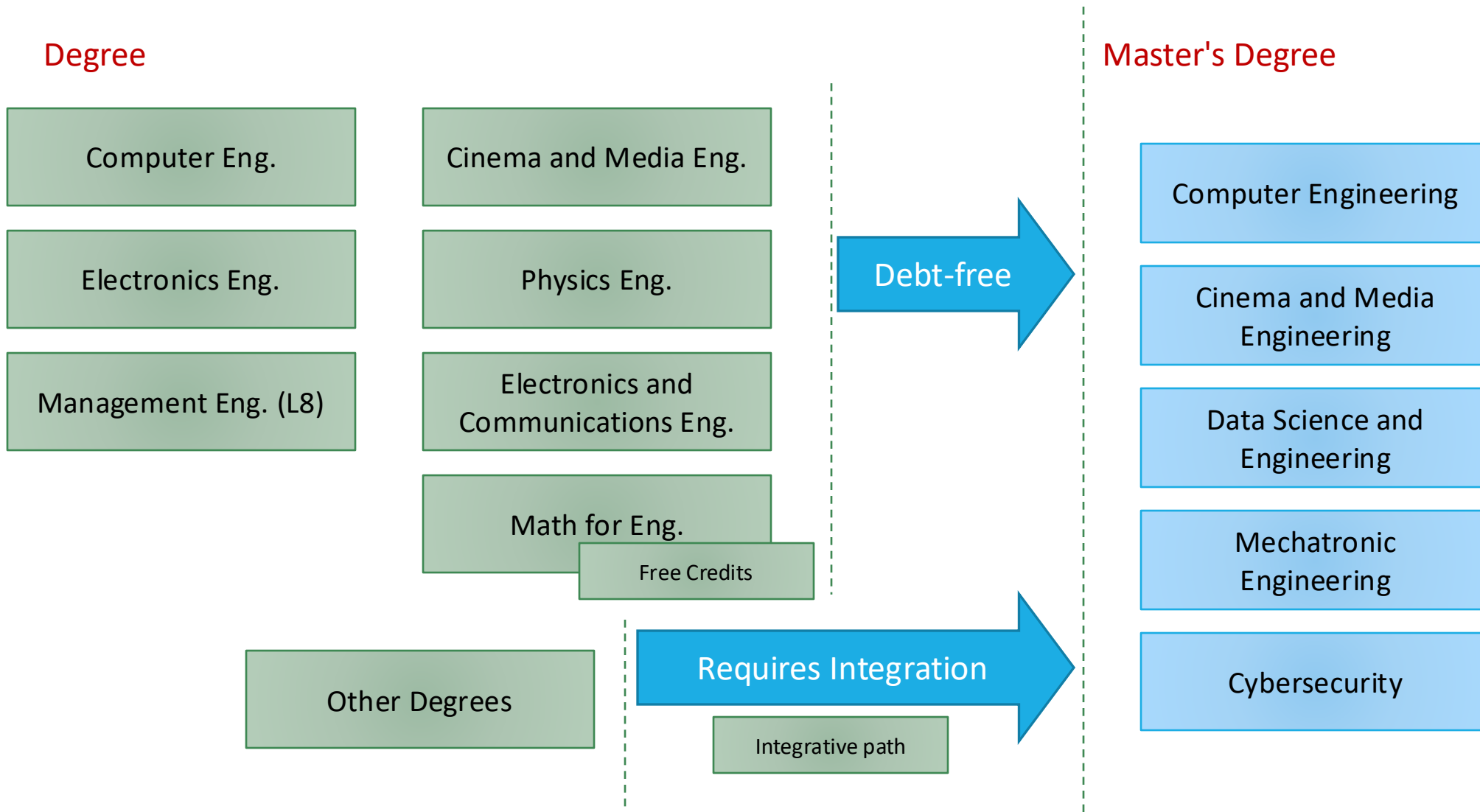
- **Example:** What are the most common first names in this classroom?

Analyses
- Understanding the problem
- Defining constraints and specifications

Design
- Choosing the strategy
- Formulation of an algorithm

Implementation
- Coding the solution
- Testing and debugging

Execution
- Application on real data

After « Computer Science »

	1st year		2nd year		3rd year	
Computer Eng.	Programming Techniques	Algorithms and data structures	Computer Architectures	Operating Systems	Computer networks	Free Credits
		Databases		Object Oriented Programming	Automatic controls	Data Analysis and Visualization
						Embedded Systems and IoT
Cinema Eng.		Algorithms and Object Oriented Programming		Databases	Computer networks	Introduction electronics embedded systems
				Computer Graphics		Introduction to Web Applications
Management Eng.		Databases		Object Oriented Programming	Internet Application Design	
			(L8+L9)	Programming Techniques	Telematic networks and internet	(L8 only)
Electronics Eng.	Algorithms and Programming					
ECE	Communication networks			Algorithms and Programming		

After « After « Computer Science » »

Degree

- Computer Eng.
- Electronics Eng.
- Management Eng. (L8)
- Cinema and Media Eng.
- Physics Eng.
- Electronics and Communications Eng.
- Math for Eng.
- Free Credits
- Other Degrees
- Integrative path

Debt-free

Requires Integration

Master's Degree

- Computer Engineering
- Cinema and Media Engineering
- Data Science and Engineering
- Mechatronic Engineering
- Cybersecurity

Course Program

Computer Science

- **Problem Posing and Solving (PPS) methodologies**
  - Analysis techniques (flow-chart, pseudo-code)
  - Implementation through computer programs

- **Data structures and information representation**
  - Numbers, Strings, Vectors, Sequences, Lists, Sets, Dictionaries, …

- **The Python programming language**

Computer Science

- Syllabus
  - Theory (9h total)
  - Problem Posing and Solving (12h total)
  - The Python programming language (41h total)

- Course organization
  - Lectures (4.5 hours/day)
  - Labs (1.5 hrs/w)

Book
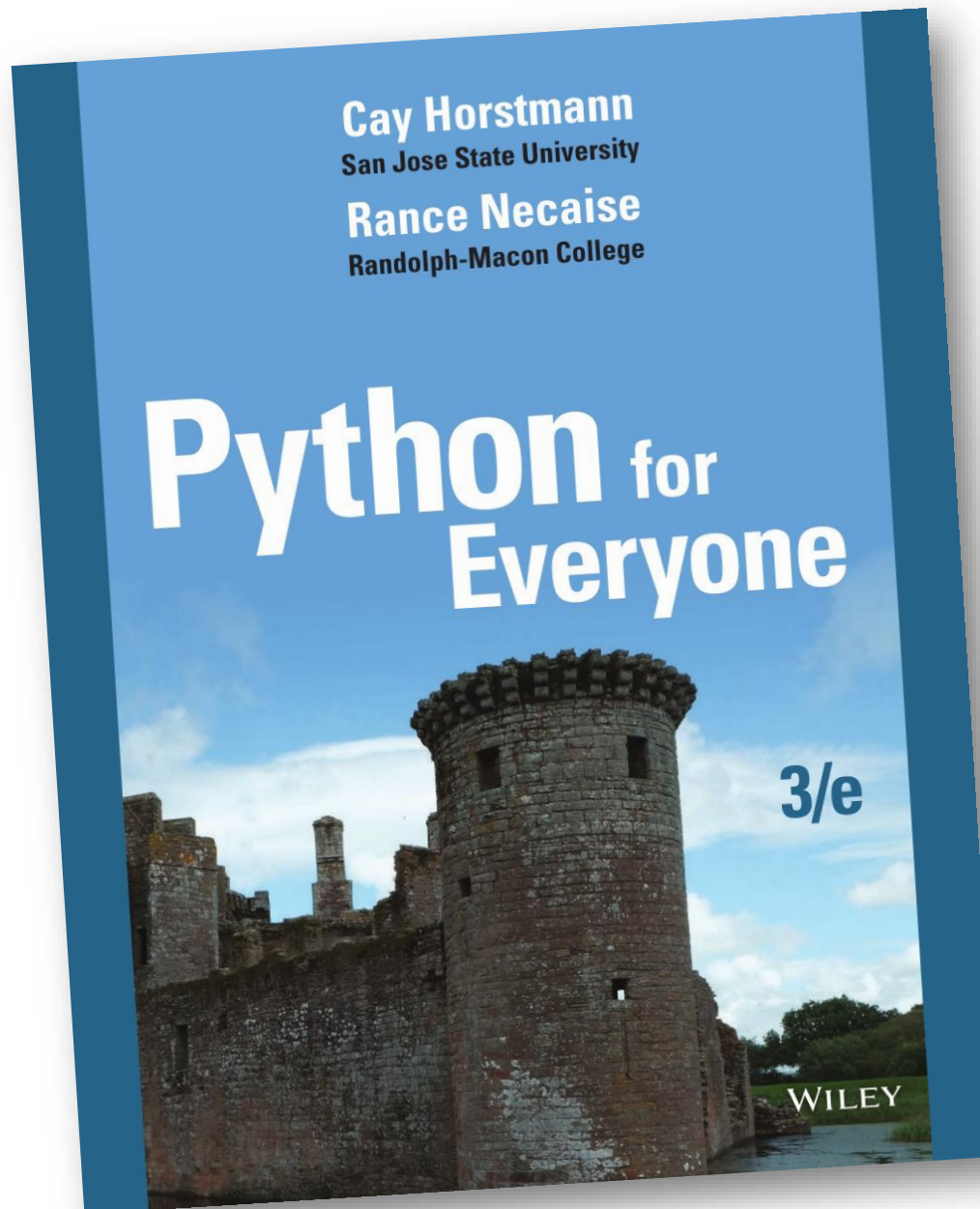
- Python For Everyone
- 3rd Edition
- Cay S. Horstmann , Rance D. Necaise
- Wiley
- ISBN: 978-1-119-49853-7 December 2018
- https://www.wiley.com/en-it/Python+For+Everyone,+3rd+Edition-p-9781119498537

Book

How to actually learn any new programming concept

Essential

Changing Stuff and Seeing What Happens

O RLY?

@ThePracticalDev
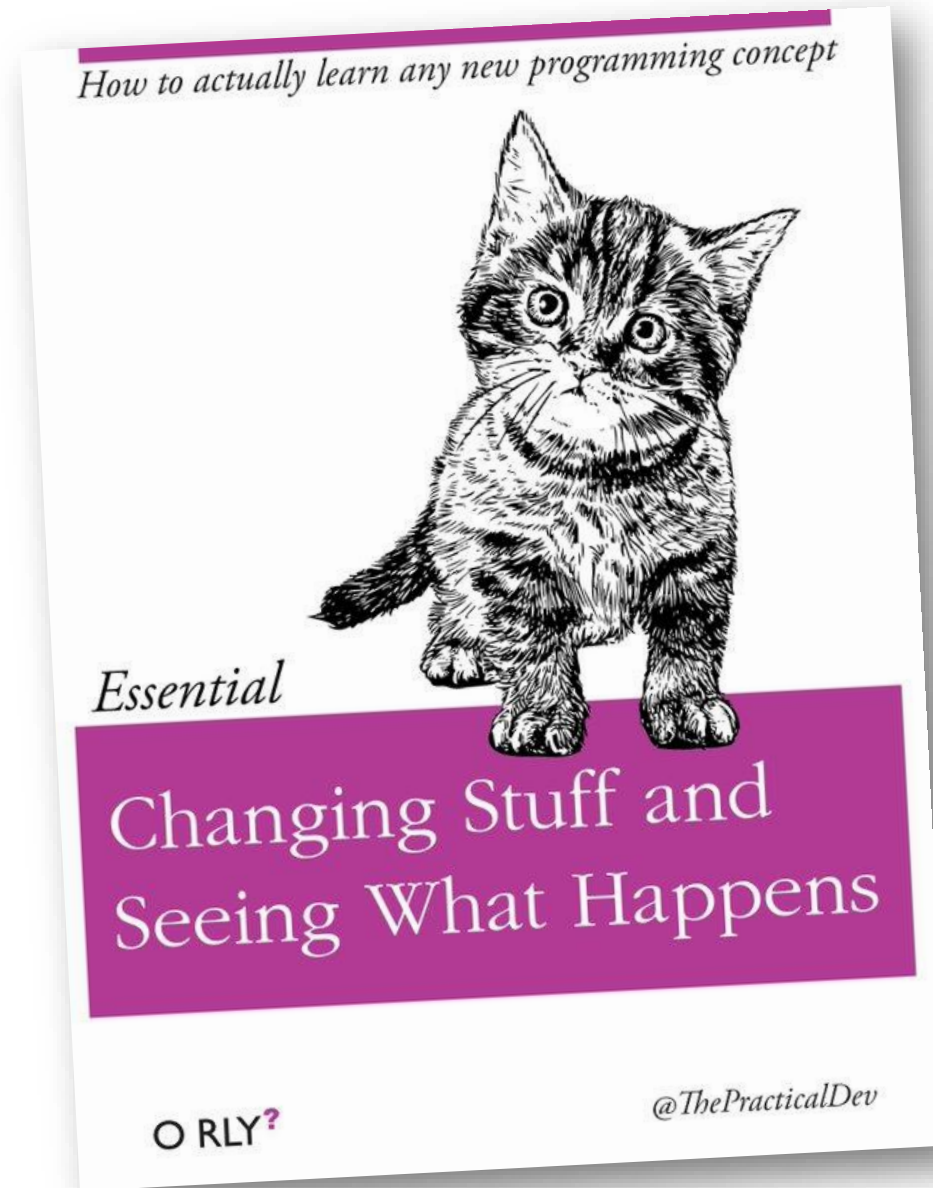
Material

- **Website** of the course at *Portale della Didattica*

http://didattica.polito.it

- o Slides used in class
- o Additional material
- o Labs
- o Exercises with and without solutions
- o Practical information

Do We Need All of This?

The Elephants in the Room

The Elephants in the Room

- ChatGPT, Co-pilot and friends can solve most of the code we'll write in this course with close to 100% correctness

- Just give them the problem description in free-text form and wait few seconds...

- **Why bother learning to program, then?**

Programming in the LLM Era

- **The Pragmatic Reason:**
  - You won't have access to those tools during the exam!

- **The Real Reasons:**
  - It is one thing to <u>write</u> code, another to <u>understand</u> what a program does.

  - What if AI gives you buggy code?
    - On more complex problems, it happens

  - Computer Science is much more than "coding".
    - Design an architecture for the solution
    - Choose algorithms, data structures, etc.

ChatGPT LeetCode Performance

Totally Wrong: 14.00% · Time Limit Exceeded: 13.00% · Failed Test Case: 45.00% · Correct: 28.00%

[source] https://dkb.blog/p/chatgpt-fails-the-coding-interview
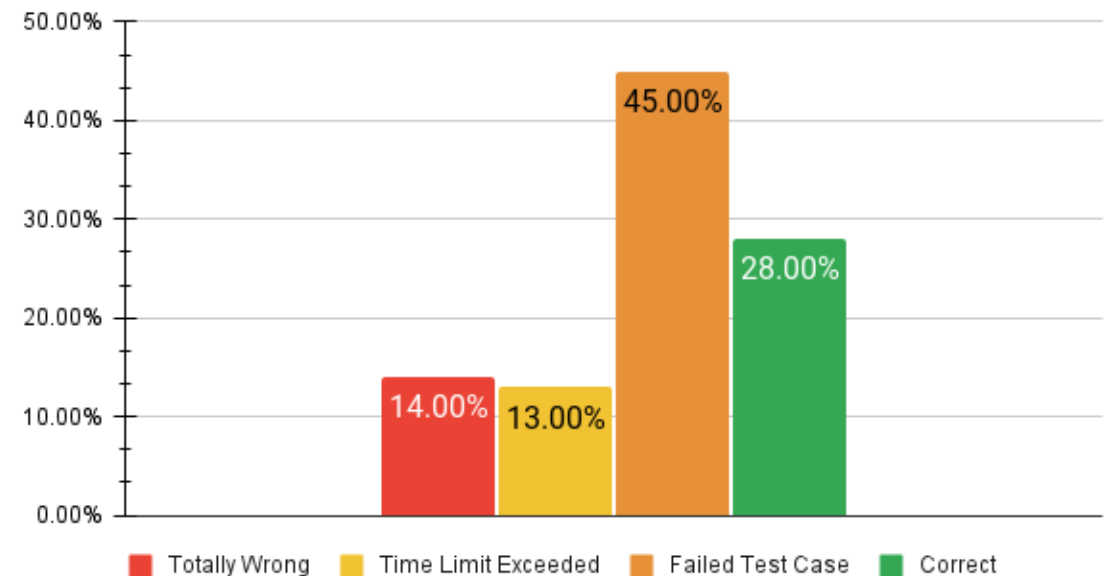
Programming in the LLM Era

- **The Real Reasons (cont'd):**
  - We have **calculators**, but we still need to understand how multiplications work.
  - We have "**fly-by-wire**" but we still want pilots to be able to land a plane in case of problem, etc.

  - <u>If ChatGPT can do your job… you'll be unemployed!!</u>

- AI-based coding tools are <u>**awesome**</u>, and you are fully **encouraged** to use them to prepare for the exam:
  - Verify your solution to exercises, ask them to explain why it doesn't work, etc.
  - Ask them to generate new exercises…
  - In your future work, using these tools will surely boost your productivity.

Programming in the LLM Era

- But please practice **<u>without</u>** these tools!

  - Try to really understand the logic of what you are doing
    - Remember: you're learning a **mindset**

  - Do the debugging effort yourself

  - Use your biological brain!!!

Practical information

	Monday	Tuesday	Wednesday	Thursday	Friday
08:30-10:00					
10:00-11:30					
11:30-13:00	LAB * (LAIB1)				Class (Classroom 1)
13:00-14:30	LAB** (LAIB1/4)				
2.30pm-4pm					
16:00-17:30		Class (Classroom 1)			
17:30-19:00					

Hours Weekly

(*Team 1)   (**Team 2 in LAIB1, Team 3 in LAIB4)

Labs

- The most important part of the course, where you learn how to solve problems and write programs

- The course is divided into 3 teams
  - At «LAIB» laboratories

- Labs start on : **30/09/2024**

- Text published on the course website

Division into teams

- Alphabetical Order based on your **Surname**

- Team 1 (11:30 am, LAIB1): from **FEC..** to **ING…**

- Team 2 (13:00 am, LAIB1): from **IOA…** to **LOD…**

- Team 3 (13:00 am, LAIB4): from **LOG…** to **OZG…**

Exam
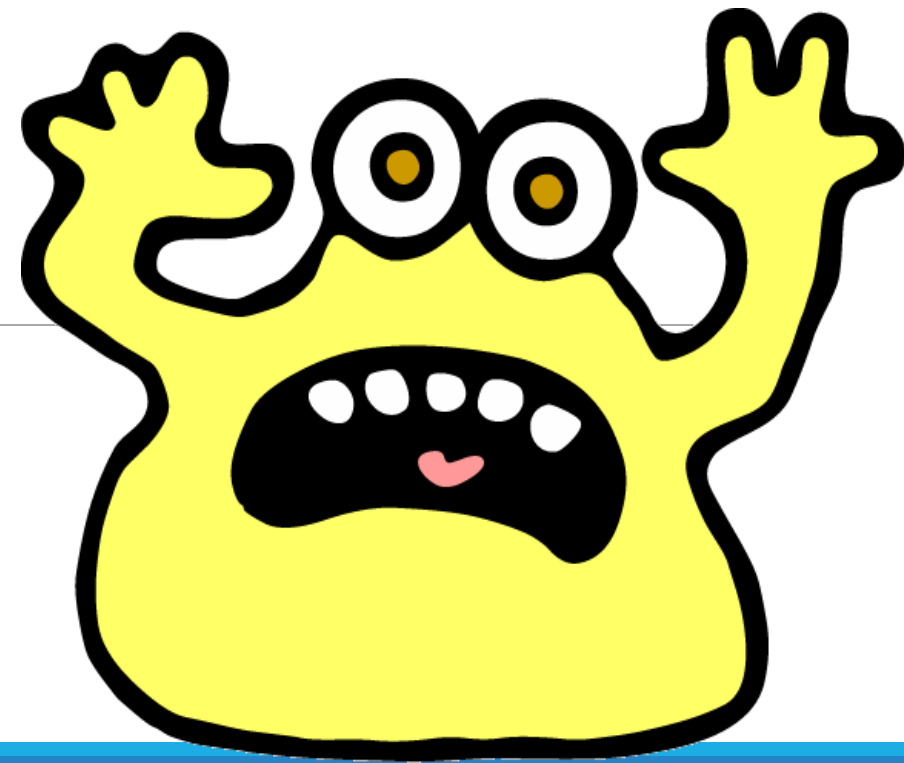
Exam contents

- Short questions about the theory part of the course [6]
  o Threshold!

- Programming [26]
  o Students work on IDE to develop their program
  o The program **<u>must be executable</u>** , maybe yielding incorrect results
  o Code will be reviewed manually only if working

Exam – New for 2024/2025

- **Experiment: mid-term test**

  o Around the end of November, during one of the labs:

  a) **2 theory questions (exam-like) + 1 Python exercise**

  - Will give you a score from **0 to 6 points** (2x3) which **can replace** the theory part of the final exam.

  - <u>At the exam:</u> you'll have to decide if you want to keep the score or re-do the theory part. If you keep it, it will be added to the programming part [26pts]

  - **Advantage**: more time for programming!

  b) **Extra exercise – or other evaluated task – to grant 0 to 3 "bonus" points**

  - We're evaluating two possibilities: 1) one extra exercise; 2) bonus for lab participation.

  Valid for the 4 exam sessions of 2024/2025

- Organization aspects (modality etc) will be clarified later...
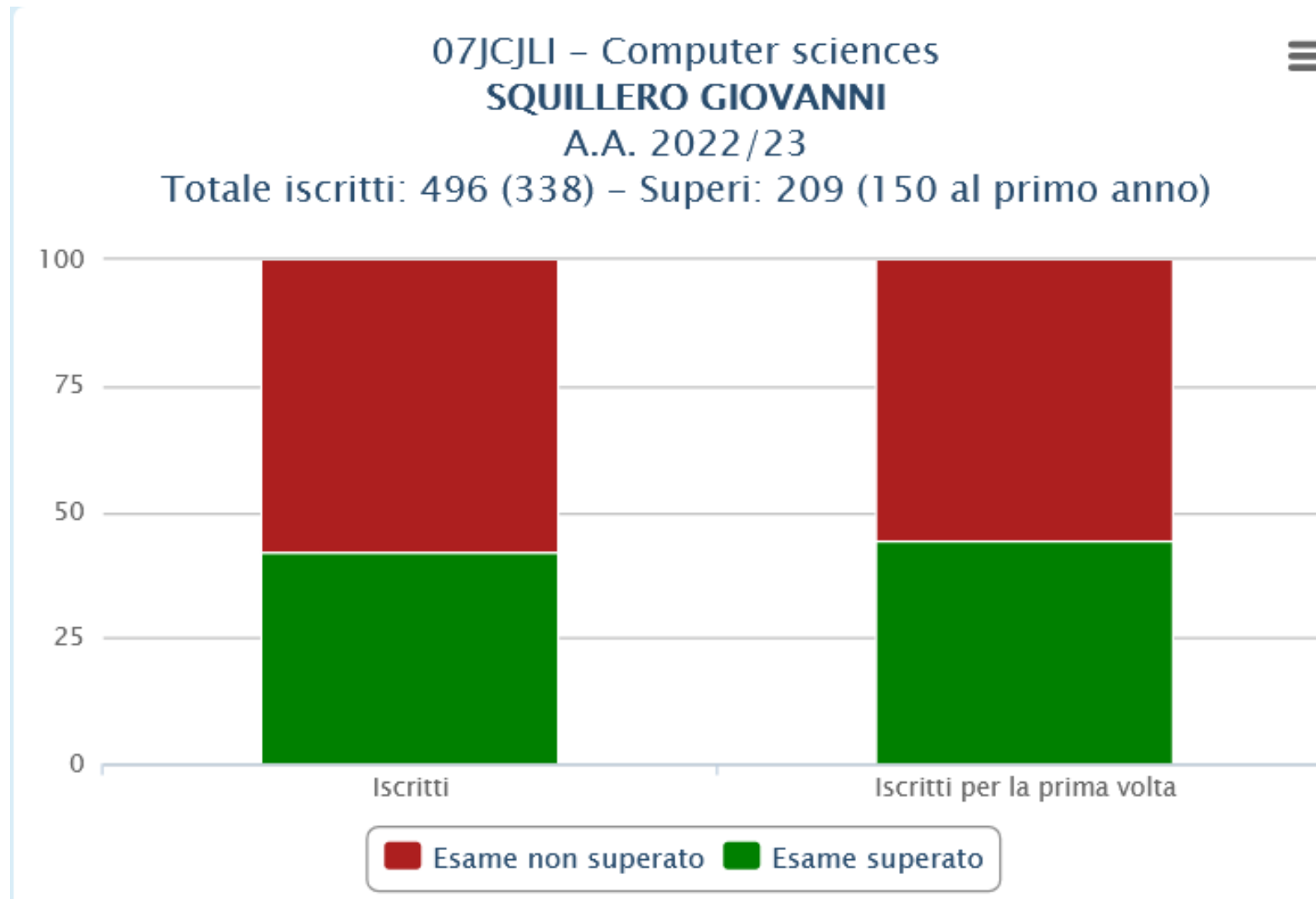
Use the English language

- **Answer the question**

- **Words make a difference**
  - "Even numbers **are** powers of two"
    "Even numbers **may be** powers of two"
    are two different sentences (the first is false, the second is true)

- **Answers need to be intelligible**

What does it take to pass the exam [well]?

- **Logical-rational analysis and synthesis skills**
  - Understand your resolution processes and be able to formalize them

- **Do all the exercises suggested**
  - Really
  - Even the ones [that seem] easy
  - Alone
  - On a computer
  - Verify them with different data
  - Try to put your programs "under stress" → Try the baddest possible cases!

- **Inventing new problems, or variations of proposed ones**
  - And then solve them

Past Experience

Last Year Statistics (Course 1)

07JCJLI – Computer sciences
SQUILLERO GIOVANNI
A.A. 2022/23
Totale iscritti: 496 (338) – Superi: 209 (150 al primo anno)

Iscritti   Iscritti per la prima volta

Esame non superato   Esame superato

Criticalities

Lack of appropriate stimuli (probably true)

Lack of creativity (true)

Lack of practice (true)

Lack of background knowledge (false)

Programming is learned by programming!

Thanks

- Part of these slides are [edited versions] of those originally made by **Prof Giovanni Squillero** (Teacher of Course 1)

License

- These slides are distributed under the Creative Commons license "Attribution - Noncommercial - ShareAlike 2.5 (CC BY-NC-SA 2.5)"

- You are free:
  o to reproduce, distribute, communicate to the public, exhibit in public, represent, perform and recite this work
  o to modify this work

- Under the following conditions:
  o **Attribution** — You must attribute the work to the original authors, and you must do so in a way that does not suggest that they endorse you or your use of the work.
  o **Non-Commercial** — You may not use this work for commercial purposes.
  o **Share Alike** — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or equivalent license as this one.

- http://creativecommons.org/licenses/by-nc-sa/2.5/it/