



COMPUTACION II

GRUPO 521-7



■ hands-on course: learning by doing!

- explanation of basic concepts, but
- working individually on exercises

- whenever you need help: **ASK!**

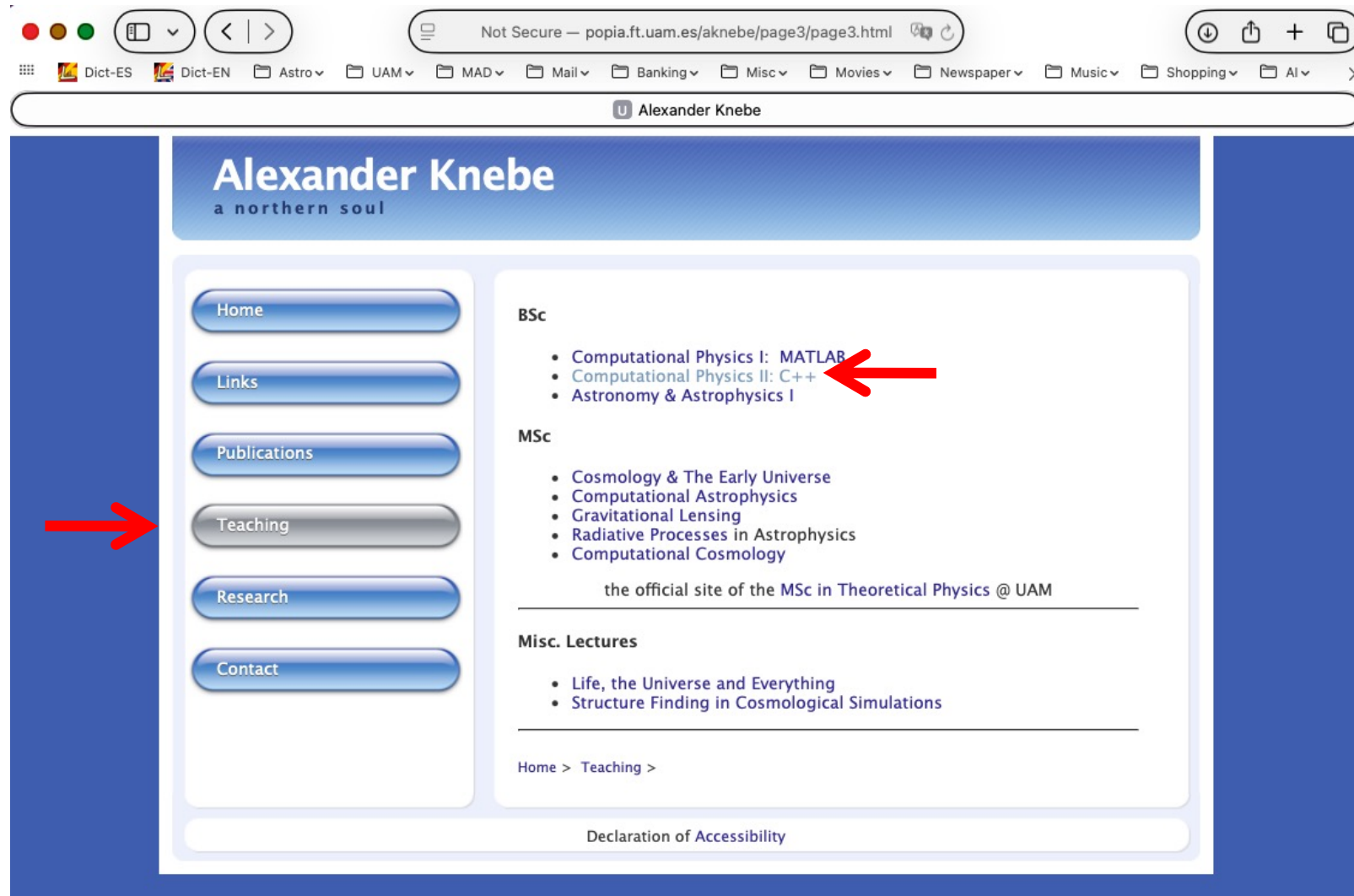
`http://popia.ft.uam.es/aknebe`

PROF. ALEXANDER KNEBE

DEPARTAMENTO DE FISICA TEORICA

GRUPO DE ASTROFISICA

MODULO M – 8, DESPACHO 316



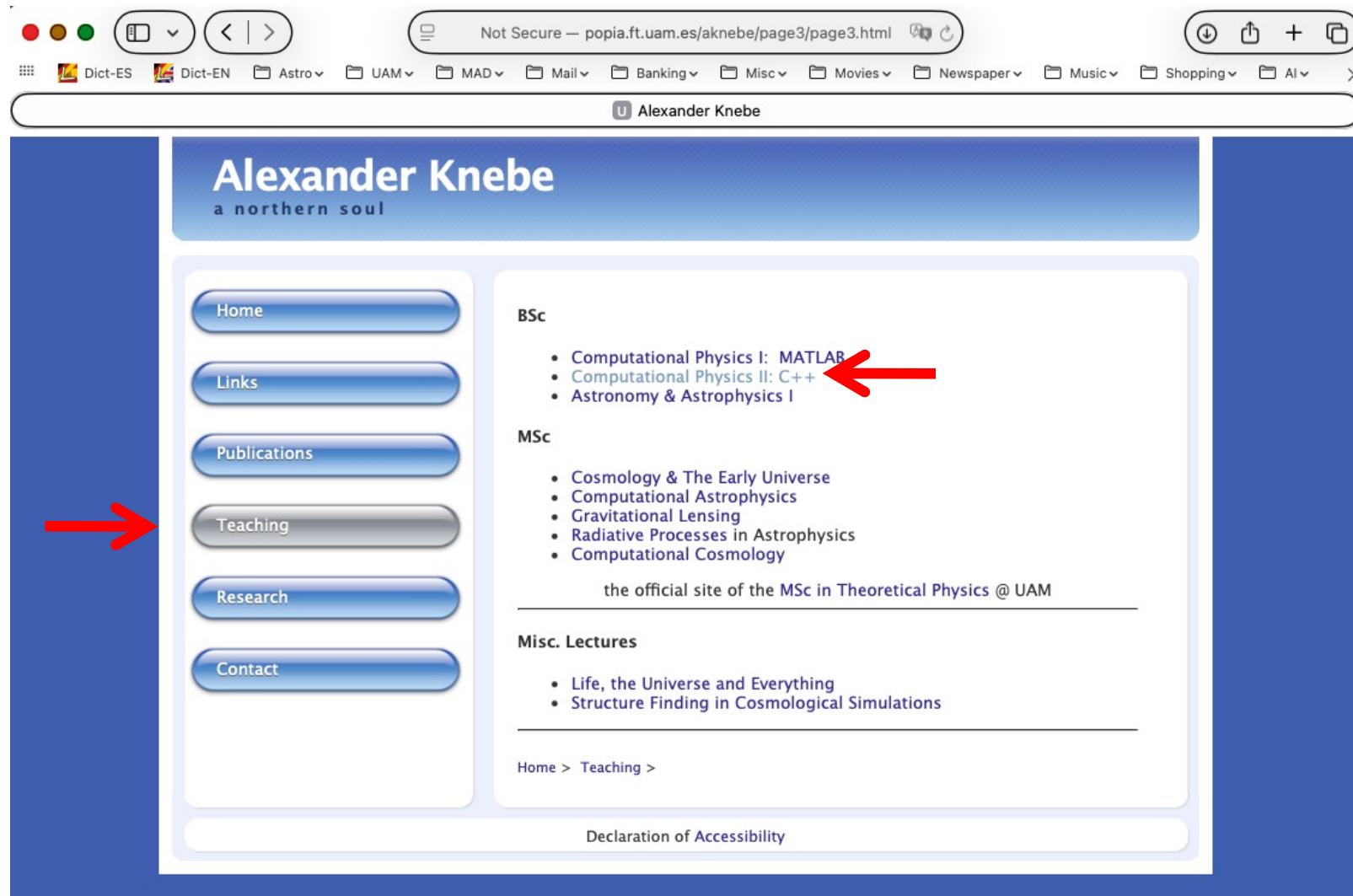
`http://popia.ft.uam.es/aknebe`

PROF. ALEXANDER KNEBE

DEPARTAMENTO DE FISICA TEORICA

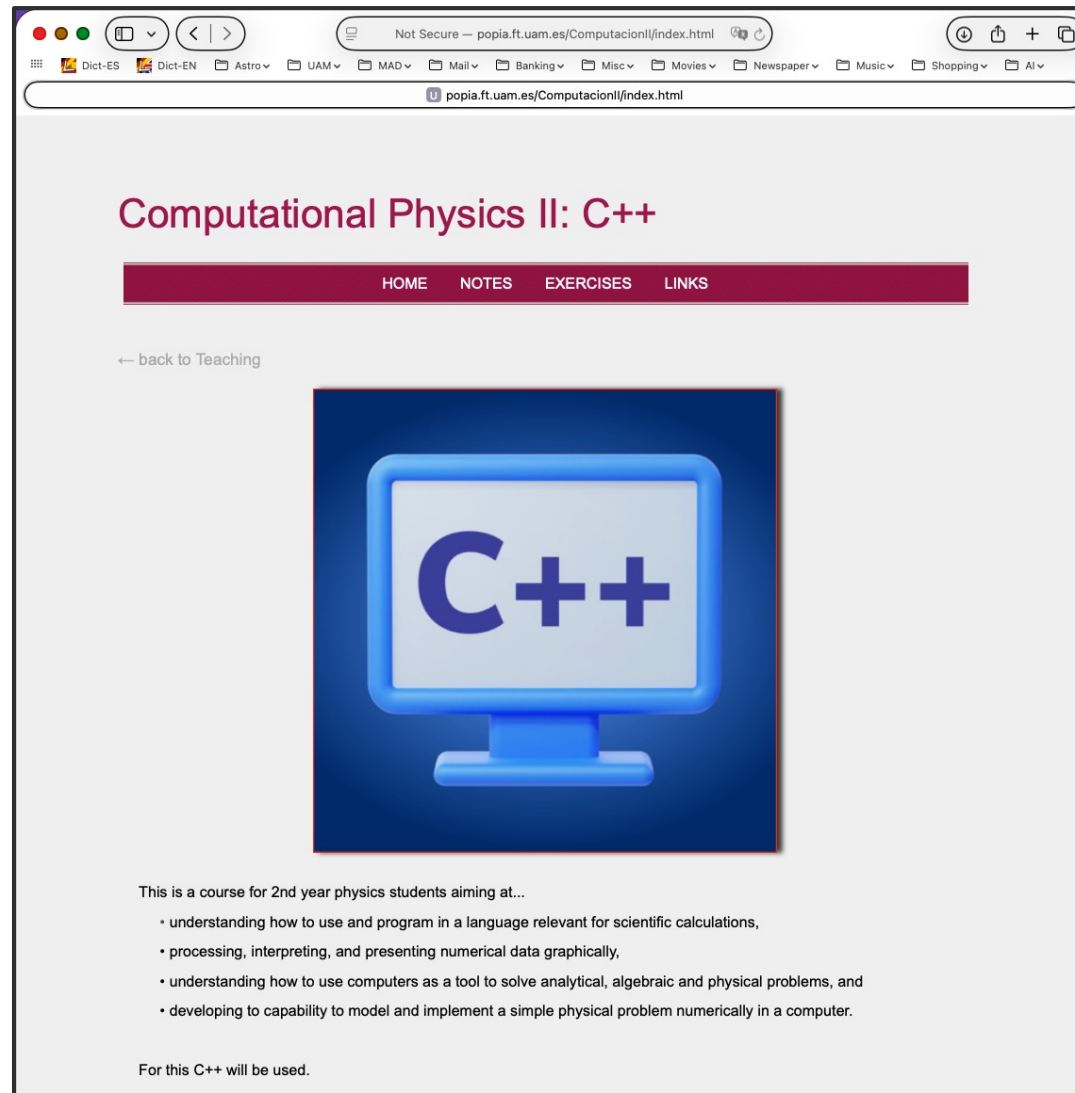
GRUPO DE ASTROFISICA

MODULO M – 8, DESPACHO 316



`http://popia.ft.uam.es/aknebe`

PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316



- Content & Itinerary
- Motivation
- C++
- getting started

- **Content** & Itinerary
- Motivation
- C++
- getting started

■ Objective:

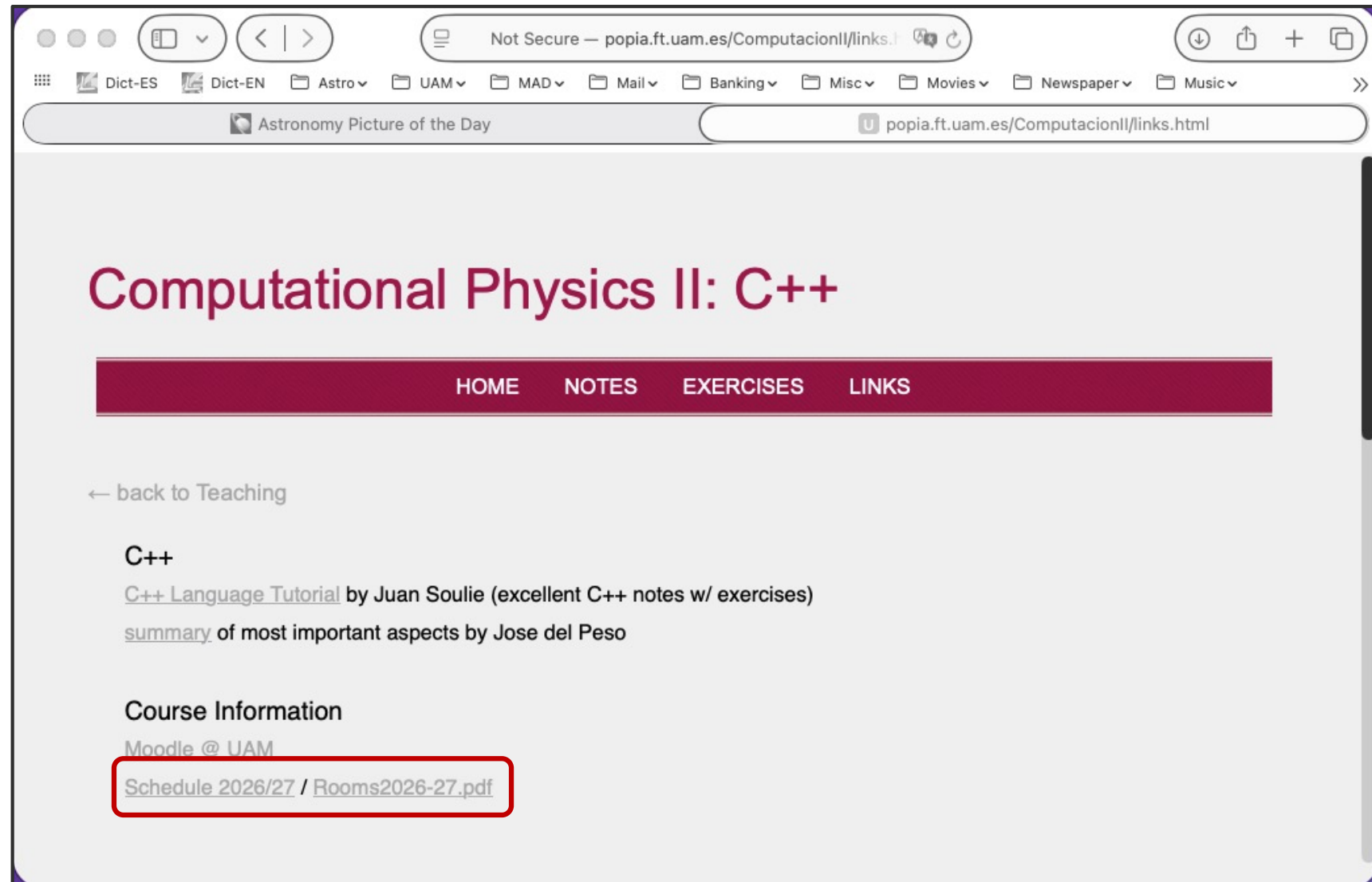
- to understand how to use and program (in a language relevant for scientific calculations)
- to process, interpret, and present numerical data graphically
- to understand how to use computers as a tool to solve physical problems
- to develop to capability to model and implement a simple physical problems
- public presentation of scientific results

■ Objective:

- to understand how to use and program (in a language relevant for scientific calculations)
- to process, interpret, and present numerical data graphically
- **to understand how to use computers as a tool to solve physical problems**
- to develop to capability to model and implement a simple physical problems
- public presentation of scientific results

- Unit 1: Introduction and Basic Concepts of C++
- Unit 2: Root Finding & (Non-)Linear Systems
- Unit 3: Matrix Diagonalization
- Unit 4: Numerical Integration and Derivation
- Unit 5: Differential Equations

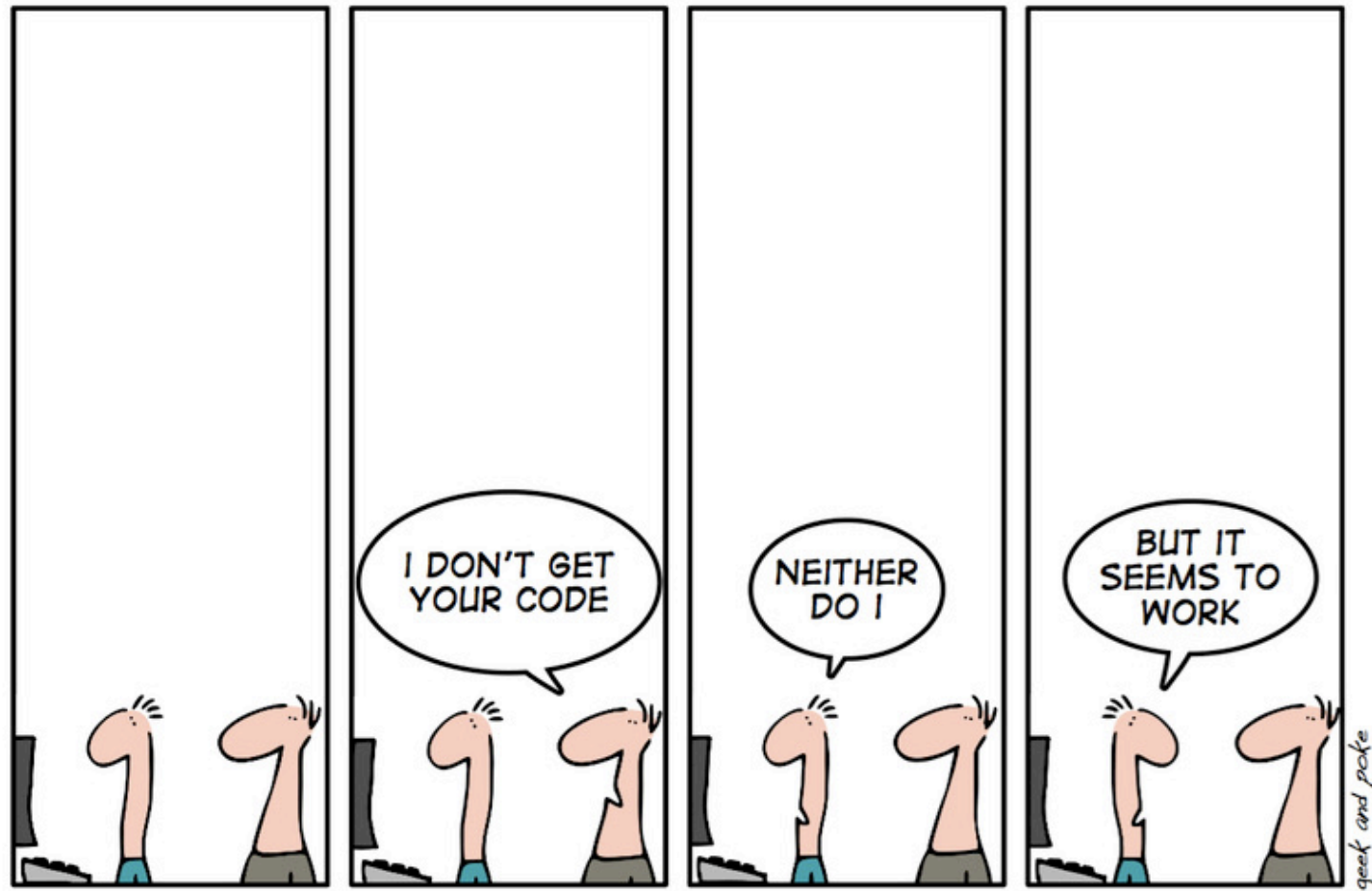
- Content & **Itinerary**
- Motivation
- C++
- getting started



	L 14/9	M 15/9	X 16/9	J 17/9	V 18/9
14:30-15:30	I.1: C++	Pr.1: C++			Pr.2: C++
15:30-16:30					
16:30-17:30					
	L 21/9	M 22/9	X 23/9	J 24/9	V 26/10
14:30-15:30	I.2: C++	Pr.3: C++		Pr.4: C++	
15:30-16:30					
16:30-17:30					
	L 28/9	M 29/9	X 30/9	J 1/10	V 3/10
14:30-15:30	II.1: C++. Ceros de una función	Pr.5: C++		Pr.6: Ceros	
15:30-16:30					
16:30-17:30					
	L 5/10	M 6/10	X 7/10	J 8/10	V 10/10
14:30-15:30	II.2: Sist ec. Lineales	Pr.7: Sist. ec. lineales (Métodos directos)		Pr.8: Sist. ec. lineales (Métodos iterativos)	
15:30-16:30					
16:30-17:30					
	L 26/10	M 27/10	X 28/10	J 29/10	V 31/10
14:30-15:30	II.3: Sist ec. no lineales	Pr.9: Sist ec. lineales tridiagonales		Pr.10: Sist ec. no lineales (Método N.R.)	
15:30-16:30					
16:30-17:30					
	L 2/11	M 3/11	X 4/11	J 5/11	V 7/11
14:30-15:30		III: Diagonalización Matrices. Autovalores		Pr.11: Autovalores. (Método Jacobi)	
15:30-16:30					
16:30-17:30					
	L 9/11	M 10/11	X 11/11	J 12/11	V 14/11
14:30-15:30		IV.1: Integración numérica		Pr.12: Integral	Pr.13: Integral
15:30-16:30					
16:30-17:30					
	L 24/11	L 25/11	L 26/11	L 27/11	L 27/11
					Examen por la mañana
	L 30/11	M 1/12	X 2/12	J 3/12	V 5/12
14:30-15:30	IV.2: Derivadas numéricas	Pr.14: Derivación numérica		V.1: E.D.O. (Condiciones iniciales)	
15:30-16:30					
16:30-17:30					
	L 7/12	M 8/12	X 9/12	J 10/12	V 12/12
14:30-15:30			Pr.15: E.D.O. orden superior	Pr.16: E.D.O. orden superior	
15:30-16:30					
16:30-17:30					
	L 14/12	M 15/12	X 17/12	J 17/12	V 19/12
14:30-15:30	V.2: E.D.O. (Condiciones contorno)	Pr.17: Método disparo		Pr.18: Diferencias finitas	
15:30-16:30					
16:30-17:30					

+ final exam on 18/01/2027

- Content & Itinerary
- **Motivation**
- C++
- getting started



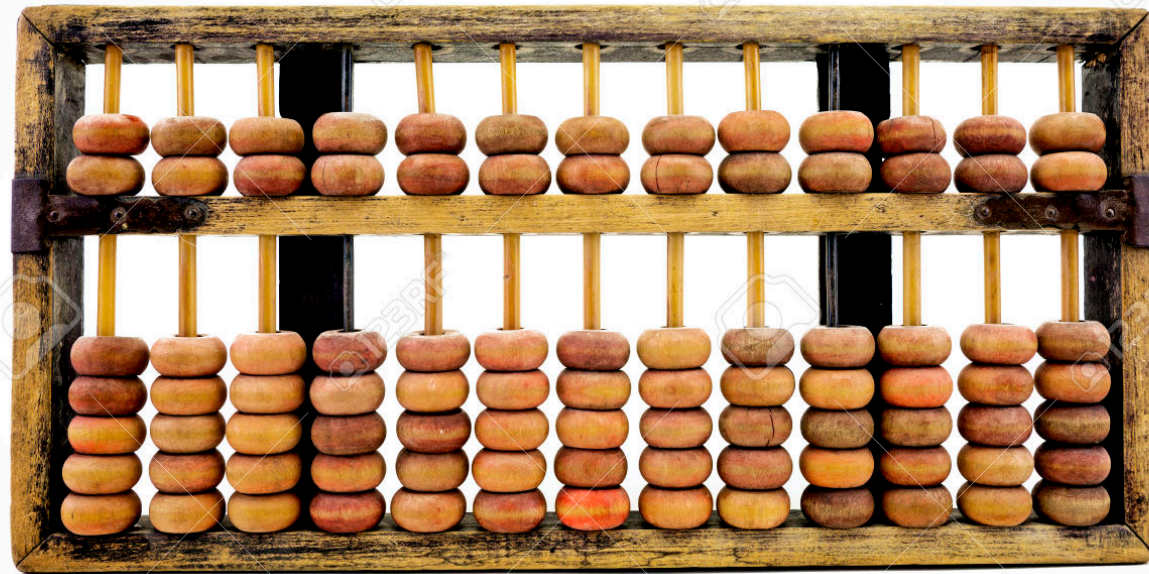
THE ART OF PROGRAMING

how and when did it all begin?



Eurasia

- suitable for any base system
- developed all across the Eurasian continent thousands of years BC

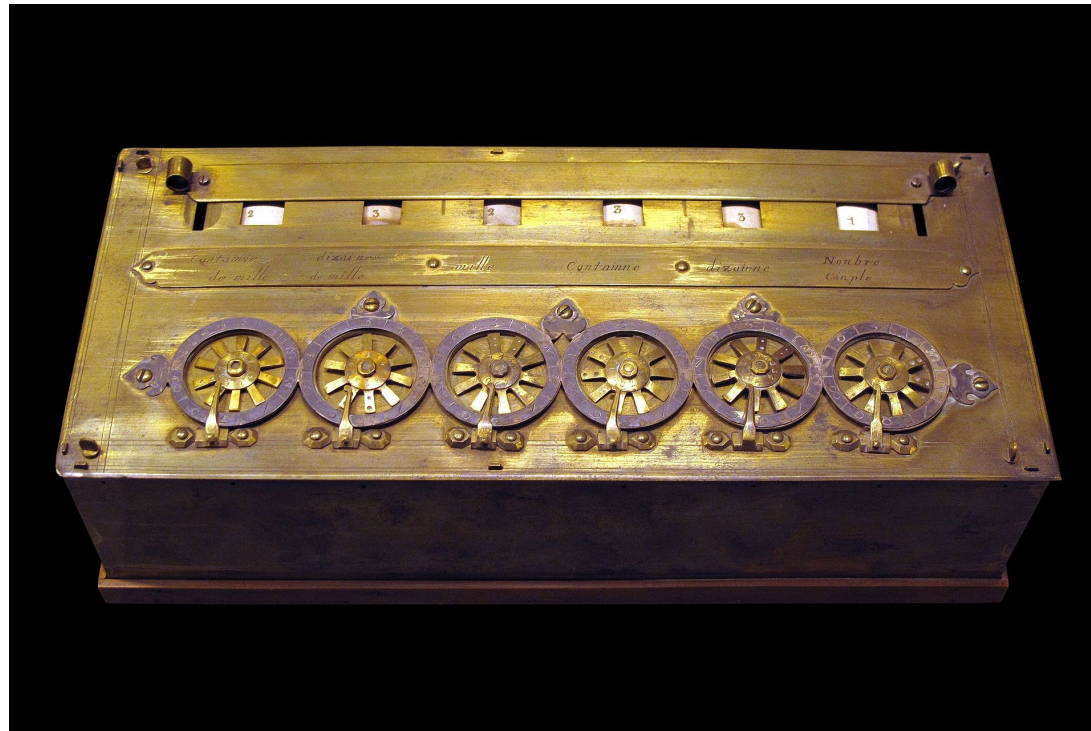


Abacus



Blaise Pascal

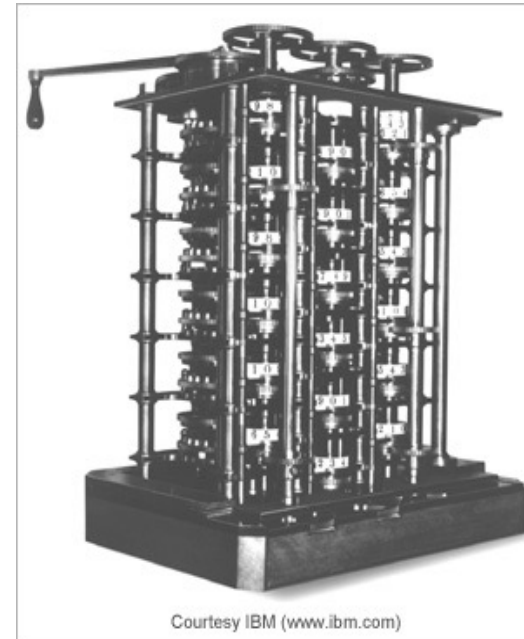
- mechanical calculator
- addition/subtraction
- multiplication/division by successive addition/subtraction
- 5 digit accuracy





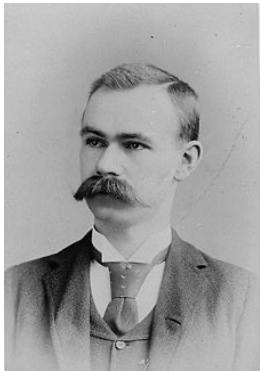
Charles Babbage

- 4000 parts
- 3 tons
- 3m x 2 m
- 31 digits accuracy
- steam driven
- “difference engine”:
 - designed to tabulate polynomials
 - calculates 2nd order differences



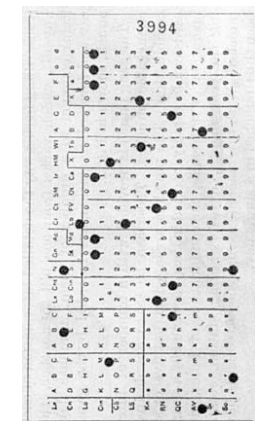
Courtesy IBM (www.ibm.com)

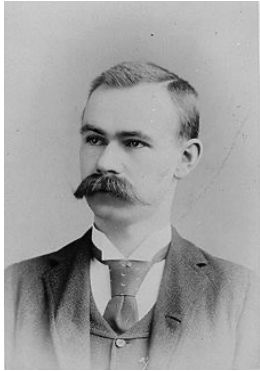
Part of one of Babbage's Difference Engines



Herman Hollerith

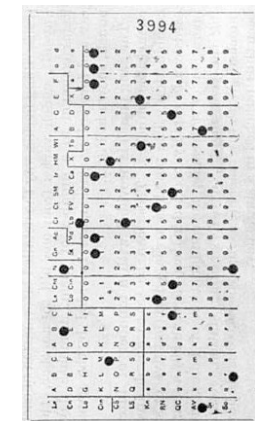
- developed a machine to read punch cards
- machines was used for 1890 census in the USA
- census took only one year to evaluate (prev. 8 years!)
- company name: Computing Tabulating Recording Corporation

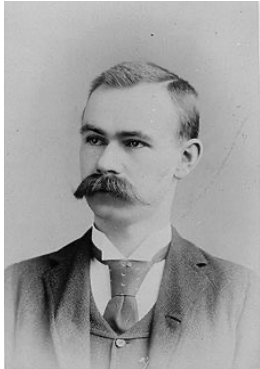




Herman Hollerith

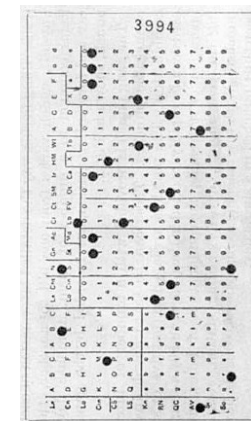
- developed a machine to read punch cards
- machines was used for 1890 census in the USA
- census took only one year to evaluate (prev. 8 years!)
- company name: Computing Tabulating Recording Corporation
- his company was renamed in 1924 to ...





Herman Hollerith

- developed a machine to read punch cards
- machines was used for 1890 census in the USA
- census took only one year to evaluate (prev. 8 years!)
- company name: Computing Tabulating Recording Corporation
- his company was renamed in 1924 to **International Business Machines (IBM)**





Alan Turing

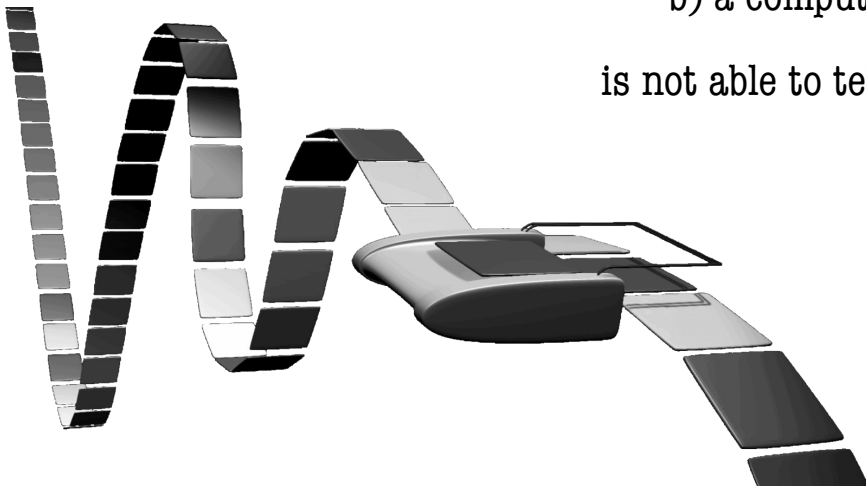
- Turing Machine:

- theoretical concept only:
 - reading instructions from printed symbols on a tape
 - result printed on back of tape
- foundation for theories about computing

- Turing Test:

- if a human interacting with
 - a) another human
 - b) a computer

is not able to tell the difference, the computer is said to “think”





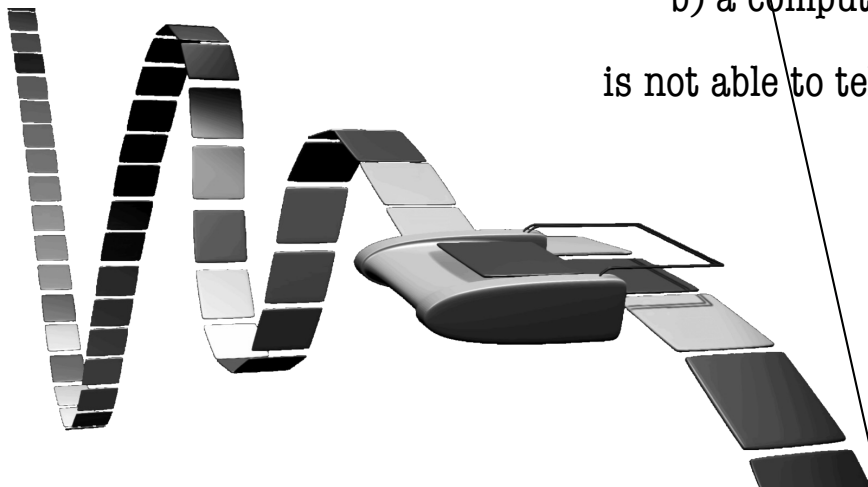
Alan Turing

- Turing Machine:

- theoretical concept
 - reading instructions
 - result printed
- foundation for the

- Turing Test:

- if a human interacting with
 - a) another human
 - b) a computeris not able to tell



Select all images with commercial lorries

Username

Password

Login with

☐

Remember me

Register

← Back

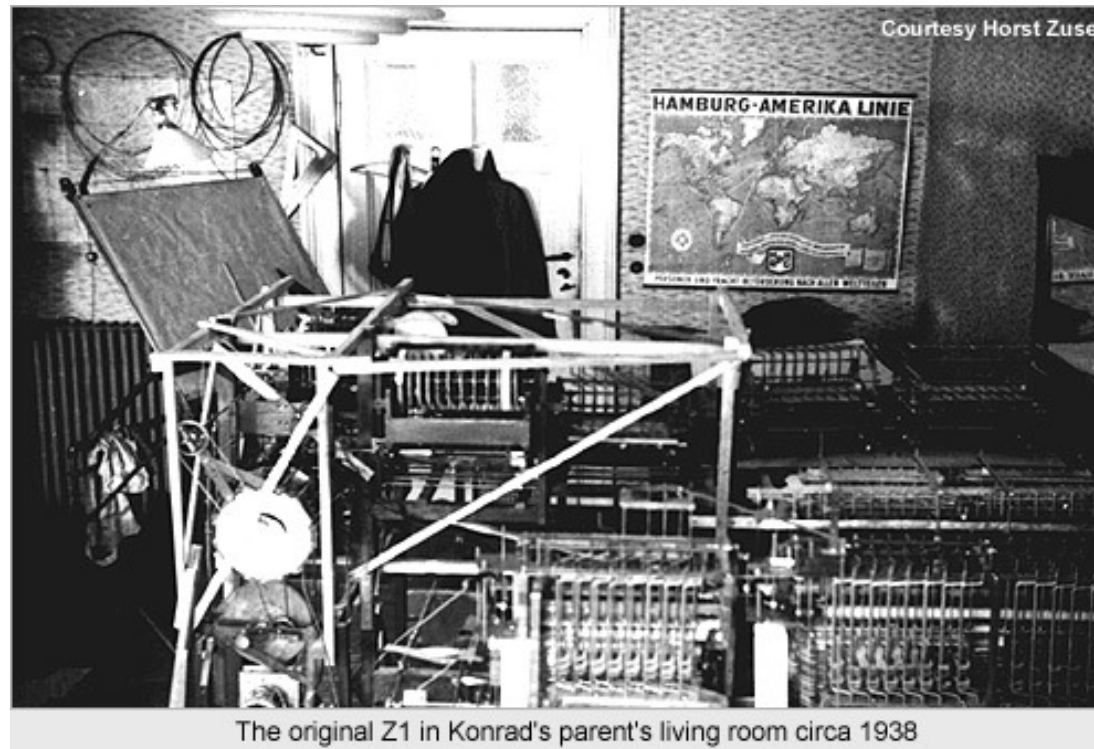
Report a Problem

Verify



Konrad Zuse

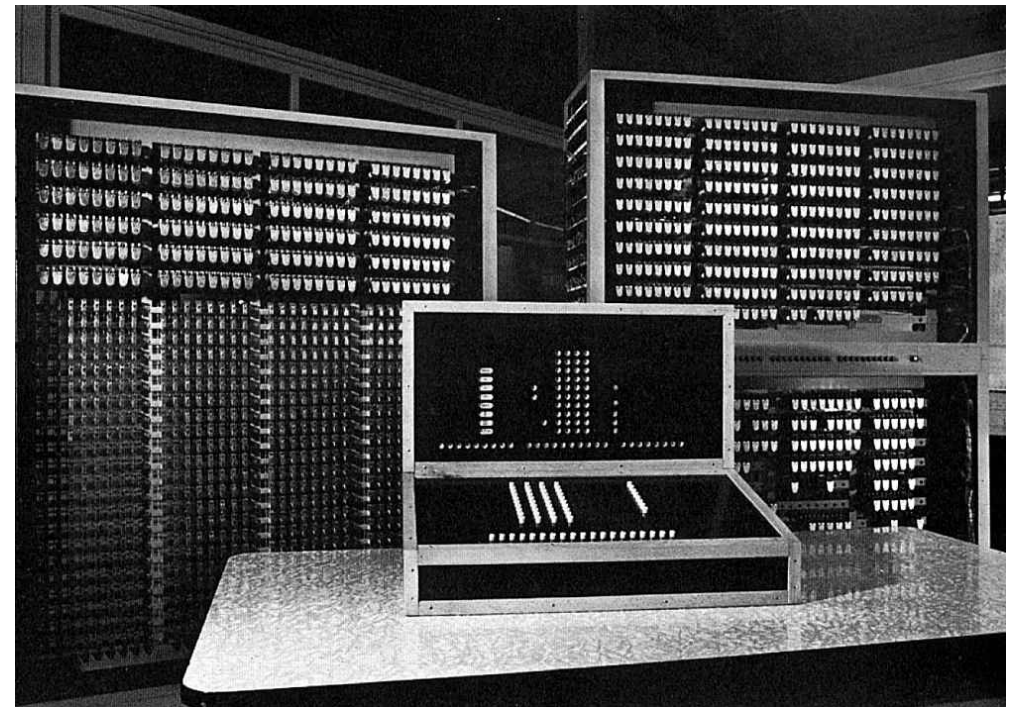
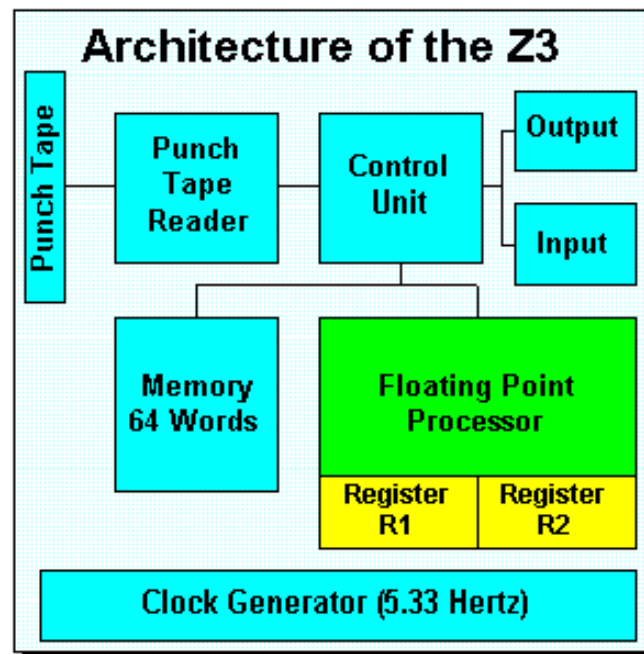
- Z1:
 - 30000 parts
 - calculations were performed in binary (while input and displayed in decimal...)
 - floating point operations
 - freely programmable via punch cards





Konrad Zuse

- Z3:
 - 5.3 Hz, 22bit, 176 bytes memory, 2600 relays
 - speed: 0.8sec/+ and 3sec/* $\Rightarrow$ 0.3 flop/s
 - floating point operations
 - freely programmable via punch cards





...and what about today's computers?

- Fugaku “Mount Fuji” (Japan), #1 in 11/2020:
 - 7,630,848 cores, 5000TB RAM
 - speed: 440×10^{15} flop/s (= 440 PetaFlops)
 - freely programmable (not via punch cards...)
 - operation system: Linux (Red Hat)



RIKEN Center for Computational Science in Kobe

- MareNostrum (Spain), #42 in 11/2020:
 - 153.216 cores, 41TB RAM
 - speed: 7 Pflop/s
 - operating system: Linux (Suse)



Torre Girona Chapel, Barcelona

- Artificial Intelligence



- Artificial Intelligence – *Computacion II exercises*

The screenshot shows the Claude AI web interface. The browser address bar displays `claude.ai/chat/662805b5-0b9f-47a7-8c69-193b80...`. The sidebar on the left contains the following elements:

- Claude** header with a chat icon and a code icon.
- + New** button.
- Customize** button.
- Projects** section with a **Pin projects to keep them here** link.
- Pinned** section with a **Computacion II** project selected.
- Ask DFT-UAM** button.
- Chats and tasks** section with links to **Customize Claude for you** and **Get a morning brief**, each with a **Try** button.
- Design** button.
- Bottom status bar showing **aknebe · DFT-UA** and icons for download, search, and a document icon.

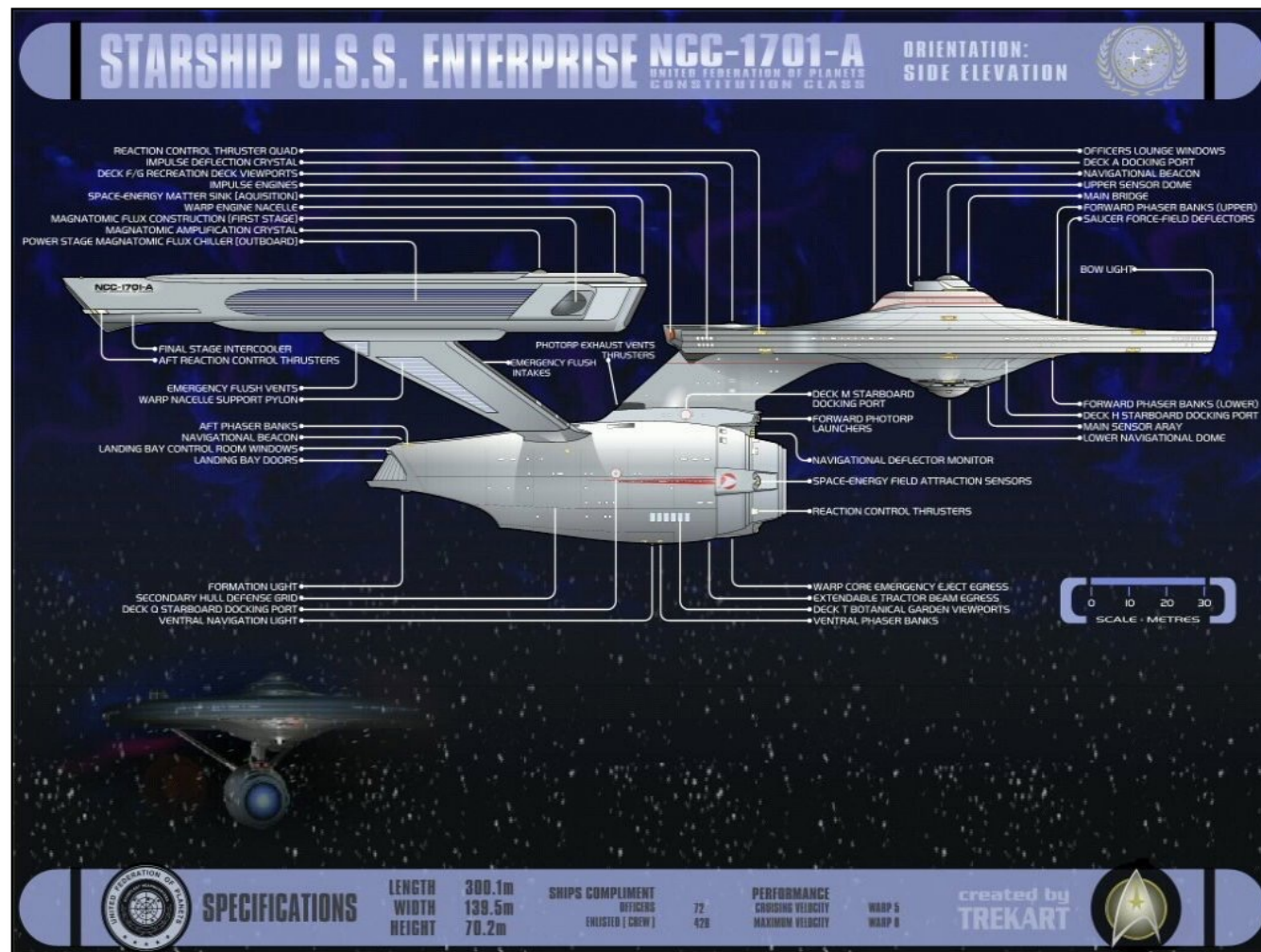
The main content area is titled **Computacion II** and features a table of exercises:

1 — Very Basics	1–9	Output, variables, int/float/double/char, casting, overflow
2 — Operators & Syntax	10–30	Conditionals, logical operators, for/while loops, strings, arrays, vectors, structs
3 — Functions & Pointers	31–40	Functions, nested calls, pointers (variable/vector/struct), matrices
4 — File Reading & Writing	41–45	ofstream, ifstream, counting, copying, round-trip data

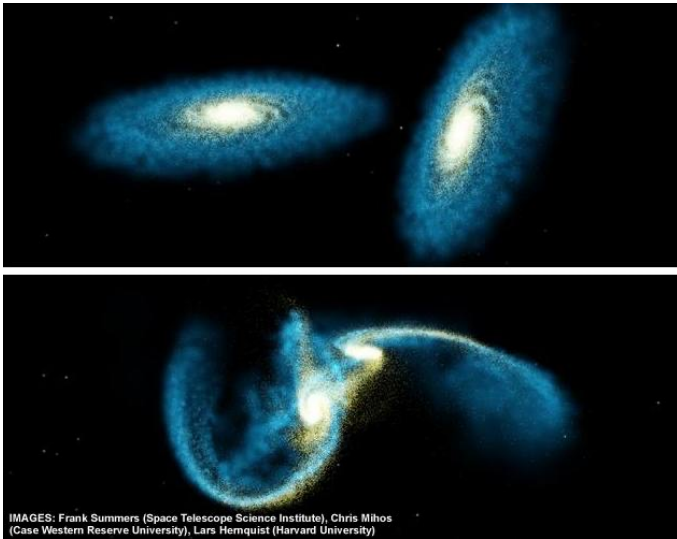
Below the table, there is a **Download** button next to a document icon labeled **C++ beginner exercises** (Document · PDF). A large downward arrow is positioned below the download button.

At the bottom of the main content area, there is a text input field labeled **Write a message...** with a **+** icon on the left. On the right side of the input field, it says **Sonnet 4.6 High** with a microphone icon and a volume icon.

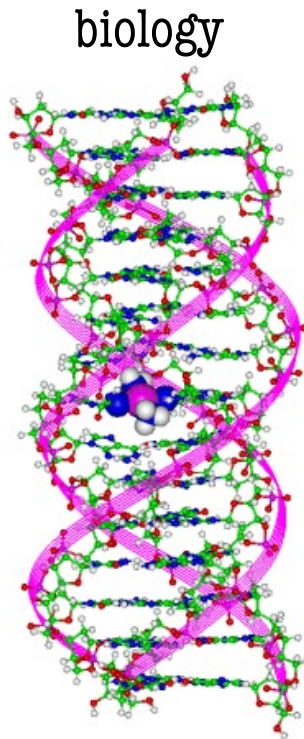
A footer note at the very bottom states: **Claude is AI and can make mistakes. Please double-check responses.**



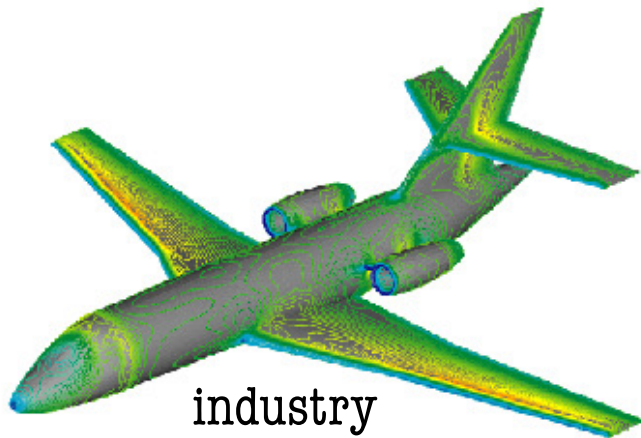
...but what about the science?



astrophysics

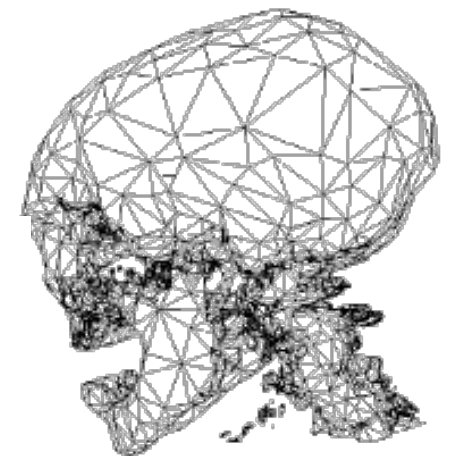


stock market

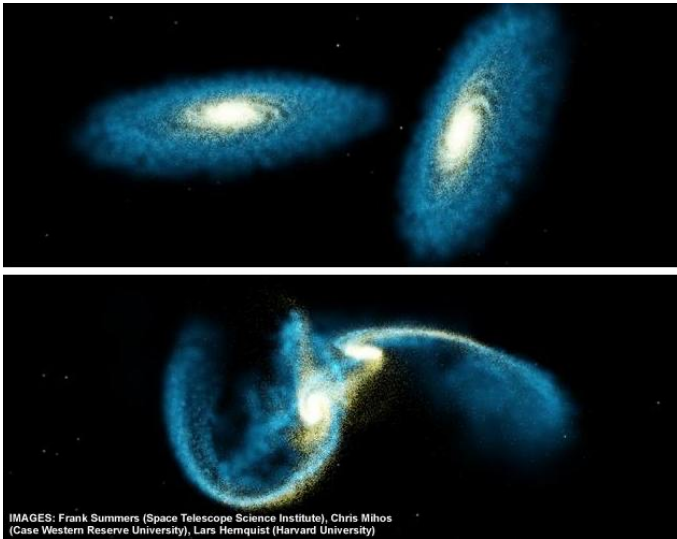


industry

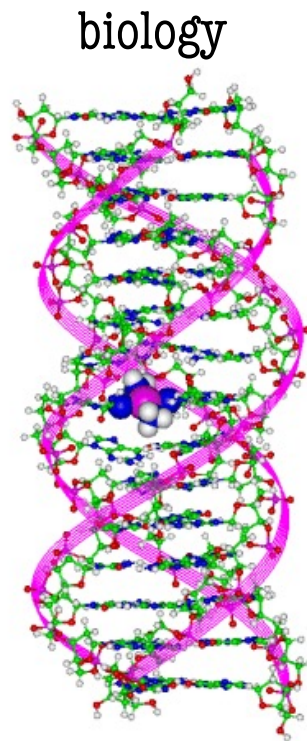
entertainment



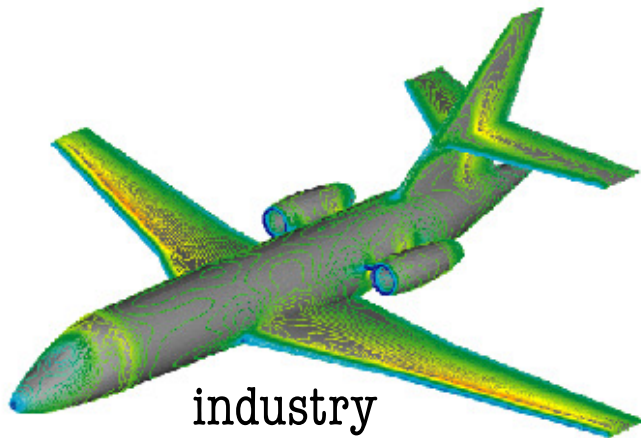
medicine



astrophysics

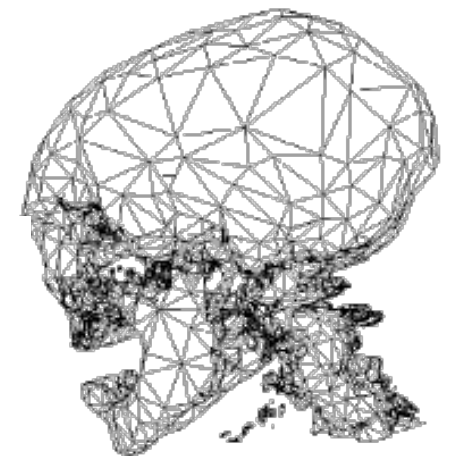


stock market



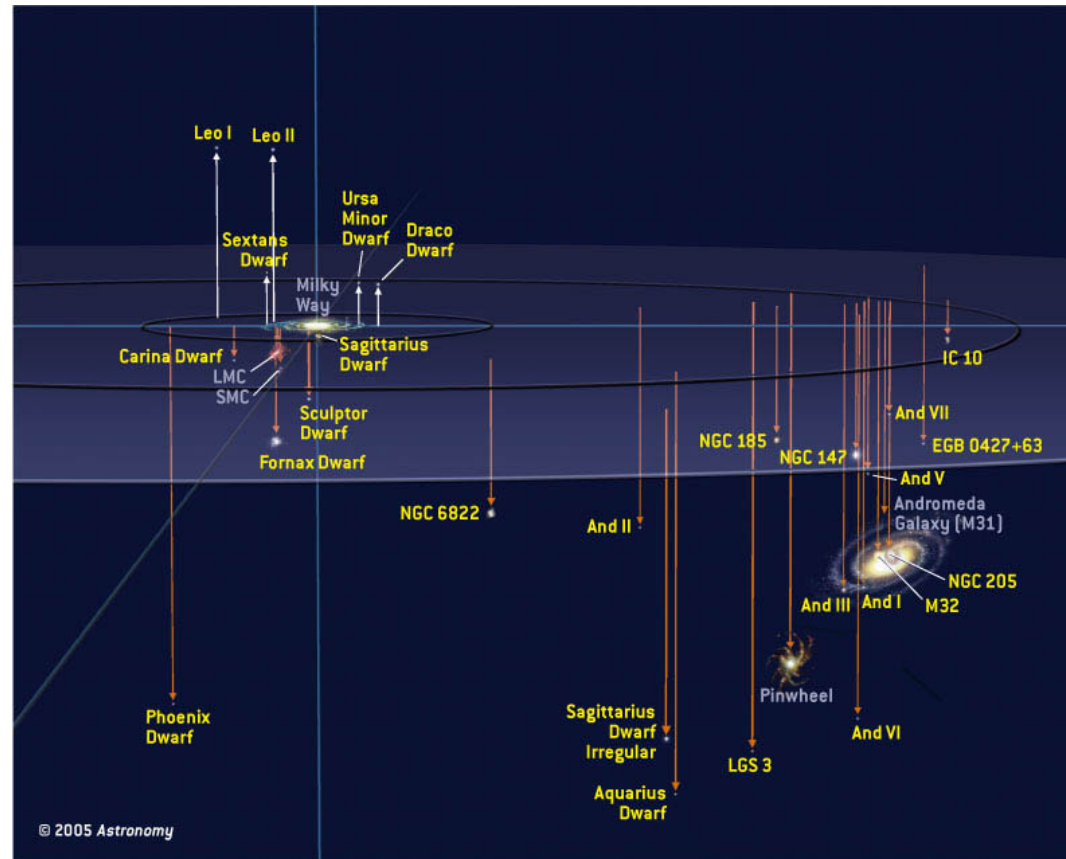
industry

entertainment

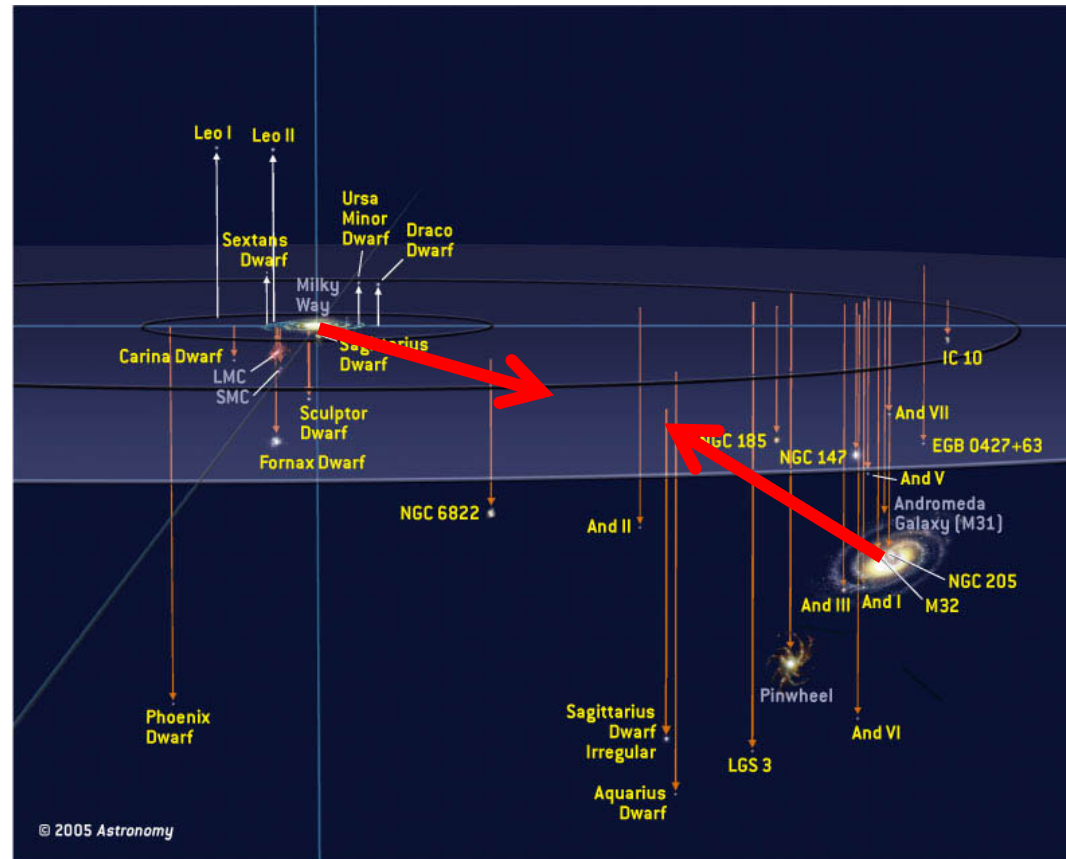


medicine

The collision of our Milky Way with the Andromeda Galaxy!



The collision of our Milky Way with the Andromeda Galaxy!



The collision of our Milky Way with the Andromeda Galaxy!



(courtesy Arman Khalatyan)

...and what about “Computacion II”?

- Content & Itinerary
- Motivation
- **C++**
- getting started

1. Why C++ Matters

- High performance with high-level abstractions
- Used where speed, control, and reliability matter
- Supports multiple programming styles: procedural, object-oriented, generic, and functional
- Huge ecosystem, long-lived codebases, and a mature developer community

2. Why C Wasn't Enough

- C offered speed and close-to-the-metal control
- But large systems needed stronger mechanisms for abstraction and organization
- C++ added classes, constructors/destructors, and stronger type-oriented design
- The result: abstraction without giving up C-like performance

3. Where C++ Is Used

- Operating systems and system software
- Game engines and real-time graphics
- Browsers, databases, and high-performance servers
- Robotics, embedded systems, scientific computing, and finance
- Applications where milliseconds, memory, or hardware access can matter

4. Key Strengths

- Performance: compiled native code and fine-grained control
- Abstraction: classes, templates, and modern libraries
- Portability: runs across major platforms and hardware
- Scalability: suitable for small components and very large systems
- Compatibility: strong relationship with existing C and legacy code

5. Challenges & Criticism

- Large language with significant historical complexity
- Compile times and build systems can become challenging
- Many language features and coding styles coexist

C++ vs. MATLAB

C++

- Compiled executable
- High performance and low-level control
- Widely used in systems, games, HPC

MATLAB

- Interpreter environment
- Matrix-oriented language
- Focus on numerical computing and plotting

C++ vs. MATLAB

choose C++

- Real-time applications
- Maximum execution speed
- Large software projects

choose MATLAB

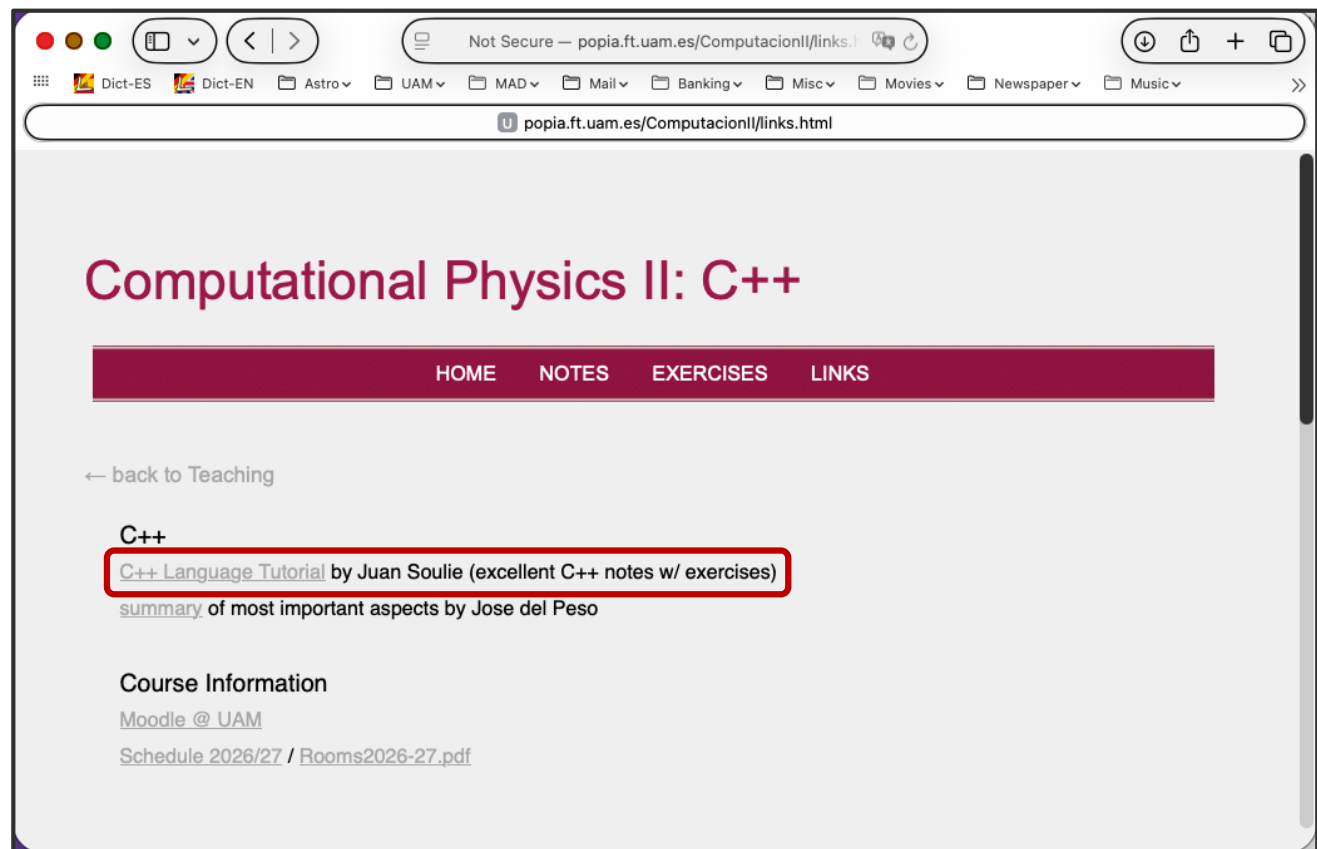
- Scientific computing
- Signal/image processing
- Rapid prototyping and visualization

- Content & Itinerary
- Motivation
- C++
- **getting started**



- learn your way around Linux

- learn your way around Linux
- take the “C++ language tutorial”



- learn your way around Linux
- take the “C++ language tutorial”...
- ...copy-and-paste the examples and run them

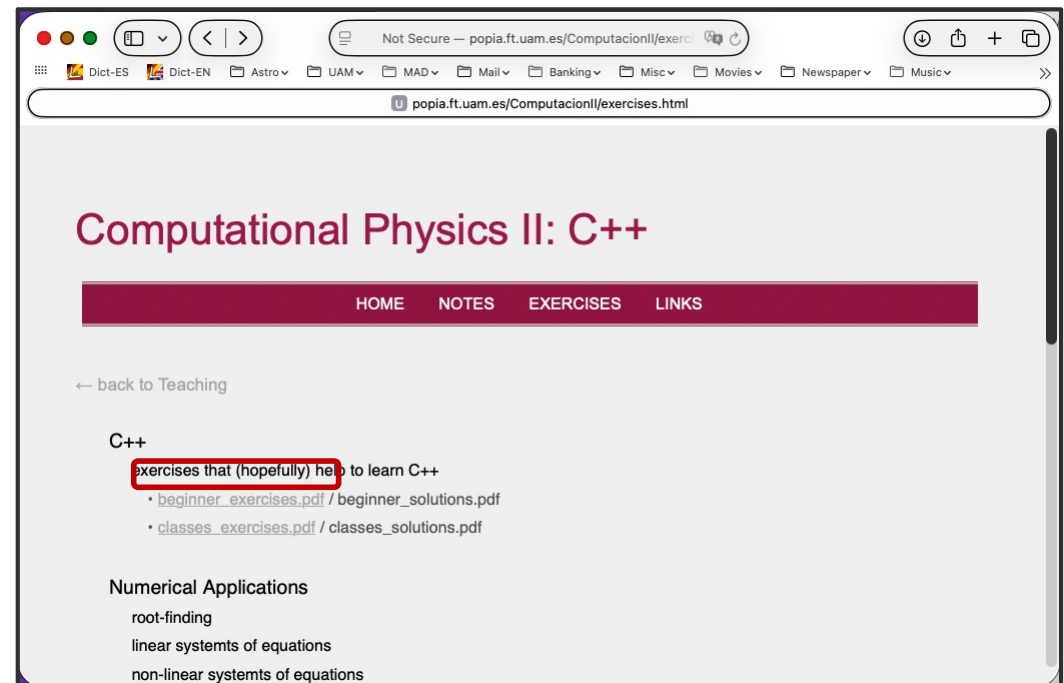
```
/* my second program in C++
   with more comments */

#include <iostream>
using namespace std;

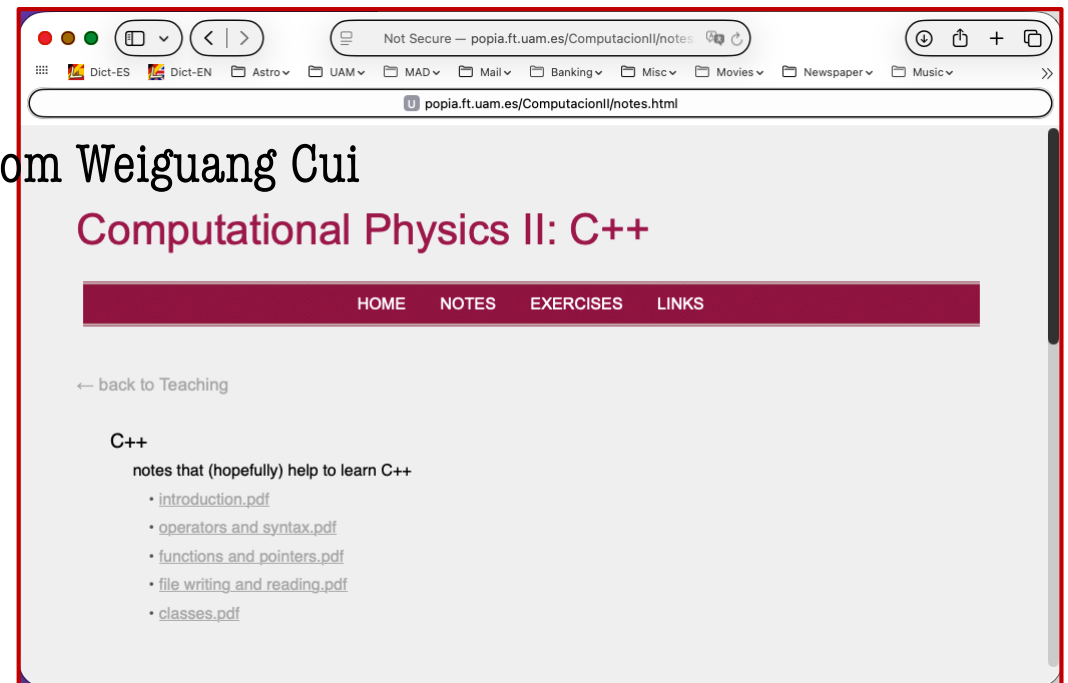
int main ()
{
    cout << "Hello World! ";    // prints Hello
    World!
    cout << "I'm a C++ program"; // prints I'm a
    C++ program
    return 0;
}
```

```
Hello World! I'm a C++ program
```

- learn your way around Linux
- take the “C++ language tutorial”...
- ...copy-and-paste the examples and run them
- do the exercises proposed by AK



- learn your way around Linux
- take the “C++ language tutorial”...
- ...copy-and-paste the examples and run them
- do the exercises proposed by AK
- check explanations and exercises from Weiguang Cui



- learn your way around Linux
- take the “C++ language tutorial”...
- ...copy-and-paste the examples and run them
- do the exercises proposed by AK
- check explanations and exercises from Weiguang Cui
- decide on some plotting program (MATLAB, gnuplot, ...)

■ Unit 1: Introduction and Basic Concepts of C++

no submission and marking of solutions

■ Unit 2: Root Finding & (Non-)Linear Systems

■ Unit 3: Matrix Diagonalization

■ Unit 4: Numerical Integration and Derivation

■ Unit 5: Differential Equations

■ Unit 1: Introduction and Basic Concepts of C++

■ Unit 2: Root Finding & (Non-)Linear Systems

■ Unit 3: Matrix Diagonalization

■ Unit 4: Numerical Integration and Derivation

■ Unit 5: Differential Equations

• *same exercises for all groups*

• *submission of solutions*

■ Unit 1: Introduction and Basic Concepts of C++

■ Unit 2: Root Finding & (Non-)Linear Systems

■ Unit 3: Matrix Diagonalization

■ Unit 4: Numerical Integration and Derivation

■ Unit 5: Differential Equations

• *same exercises for all groups*

• *submission of solutions*

do not use AI, for your own sake!
seek help and ask your teacher instead!