Programming II

Pedro Ribeiro

DCC/FCUP

2024/2025

General Information

Main Resources

-  -  **Website:**
  `http://www.dcc.fc.up.pt/~pribeiro/aulas/pii2425/`
  (all material for this course, including lectures and practical classes)

-  - **Discord:** Slack: `https://discord.gg/D7r4y8Tgsa`
  (used for communcation in the course)

-  - **Mooshak:**
  `http://mooshak.dcc.fc.up.pt/~pii/`
  (code submission and automatic evaluation)

Teaching Team

- **Theoretical Classes:** *(2h per week, auditorium)*
  - Pedro Ribeiro

- **Practical Classes:** *(2h per week, computer lab)*
  - Pedro Ribeiro (1 class)
  - Alberto Barbosa (2 classes)

Note that classes are concentrated on only **6 weeks**! *(3 ECTS)*

Teaching Team - Pedro Ribeiro

Name: Pedro Ribeiro
Office: FC6 1.47 (DCC/FCUP)
Website: https://www.dcc.fc.up.pt/~pribeiro/

Main Research Interests

- Complex Networks, Network Science, Graph Mining, Data Science
- Algorithms and Data Structures, Complexity
- Parallel and Distributed Computing
- Computer Science Education and Programming Contests

Competitive Programming - Pedro Ribeiro

I'm involved in many **algorithmic programming contests**:
(organization, problem creation, mentoring and training, ...)

- Pre-Universitary Students (Basic and Secondary Education)
  - National and Internacional Informatics **Olympiads** (e.g: ONI, IOI)
  - **Bebras** - Computational Thinking International Challenge
- University Students
  - National and Internacional **ICPC** contests (e.g: MIUP, SWERC, EUC)

Eligibility for exams ("Frequência")

To be eligible for the exam you need to:
attend to at least 50% of the practical classes

- 6 practical classes predicted, so you need to attend at least 3

There is **no minimum grade** on any evaluation component

Calculation formula of final grade

The final grade is obtained taking into account the following components:

- **A** (10%): grading of the exercises submitted to the automatic assessment system during the lessons
  (set of exercises each class, +/- 2 weeks to submit them)
- **T** (40%): practical test at the end of the 6 weeks of classes
  (after 6 weeks of classes, 2h, individual, computer lab)
- **E**: final written exam
  (2h30m, you will have access to model exam)

The final grade (**F**) is determined by $\mathbf{F = A \times 0.1 + T \times 0.4 + E \times 0.5}$

Improvements:

- One chance to improve the practical test before the exam period
- "Recurso" will only give the option to improve exam component

Practical Component

-  You will be using **C** programming language
- You will be coding in a computer lab, in **Linux**, with usual editors and GCC (the C compiler)
- There will **no internet**, but will have access to language **documentation** and **class material**
- You will be given specific **goals for the test** (including number of problems and valuation)

Predicted dates for practical tests:

- **Practical Test:** right after easter vacations (end of April?)
- **Improvement Practical Test:** before exams (end of May?)

Goals of the course

Objectives

The aim is to **complement** previous knowledge of programming in an **interpreted** language (**Python**) with specific training in programming in a **compiled**, **low-level** language (**C**).

Emphasis will be placed on practical **problem-solving** with a focus on notions of **algorithms**, **modularity**, **structured programming** and **code quality**.

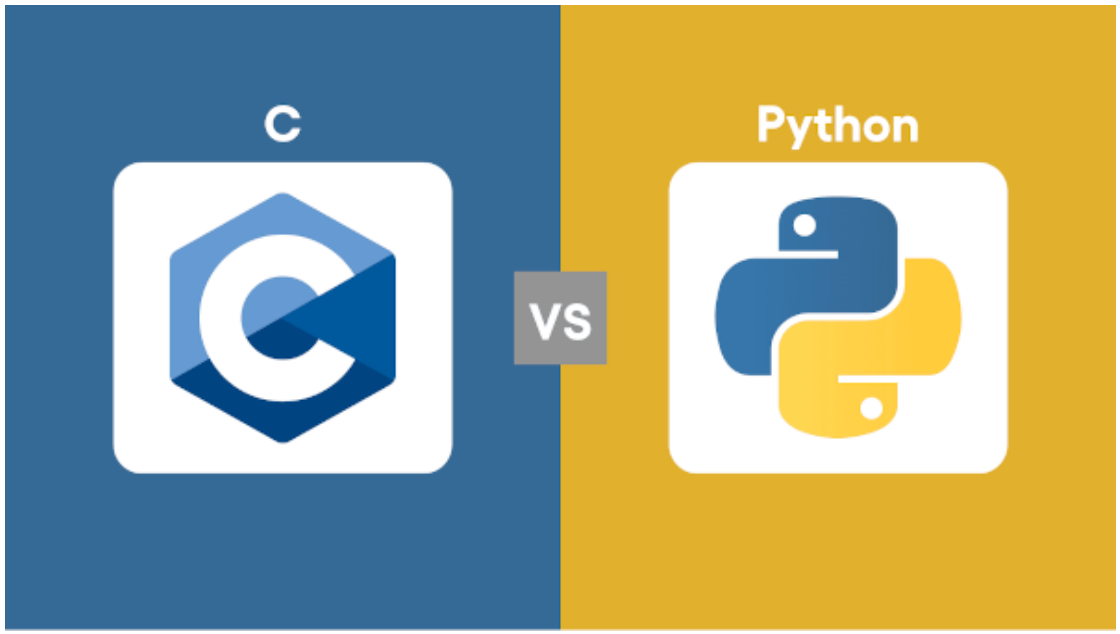
Learning outcomes and competences

At the end of this course, students should be able to:

- use the **syntax** and **semantics** of the fundamental components of the C language
- **write**, **test** and **execute** programmes to solve simple problems from an informal specification
- implement some **elementary algorithms** in C
- know the concept of pointer and use it to process **indexed variables** and **character strings**
- know and use functions from the standard C **language libraries**
- have basic skills in writing **structured**, **correct** and **efficient** code

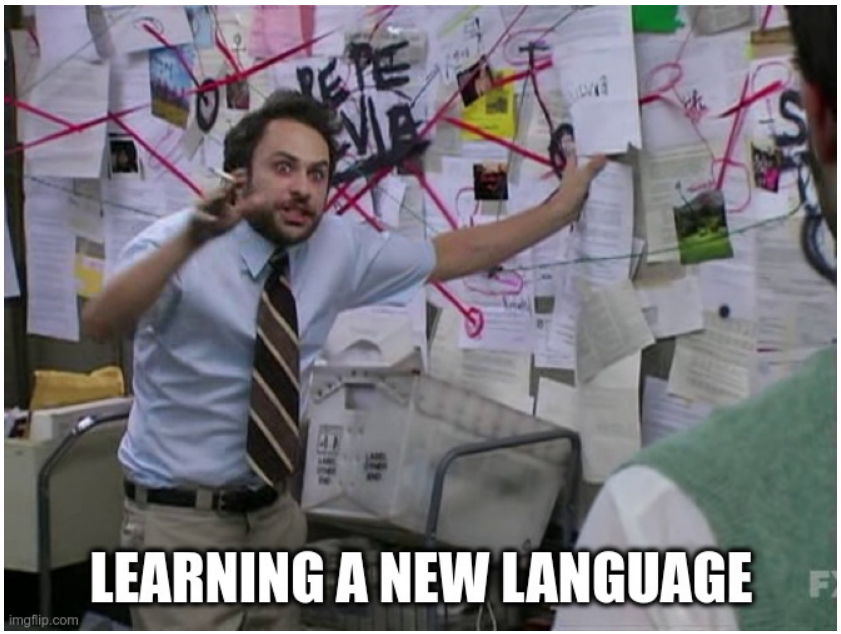
Syllabus

- Introduction to the **C language**. Characteristics of the language: advantages, disadvantages and care when using it.
- C **language fundamentals**. Syntactic structure of programs. Directives, declarations, expressions. Compilation and execution.



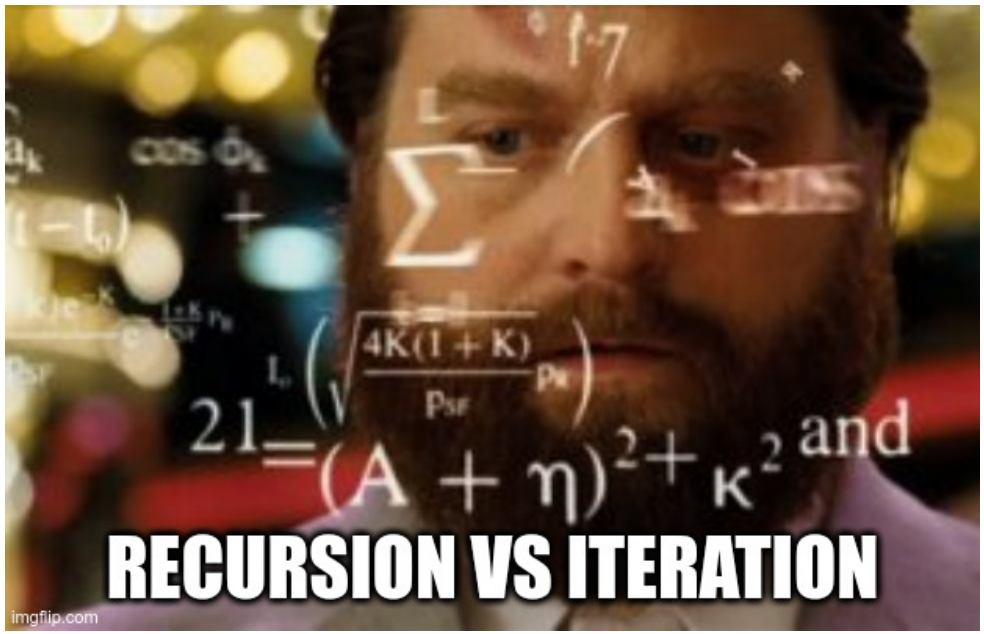
Syllabus

- Basic **types** (integers, floating point, characters). **Flow** control. **Cycles**. **Function** definition. Formatted **input** and **output**.
- Notions of **pointers**. Indexed variables. Character strings. Structured types.



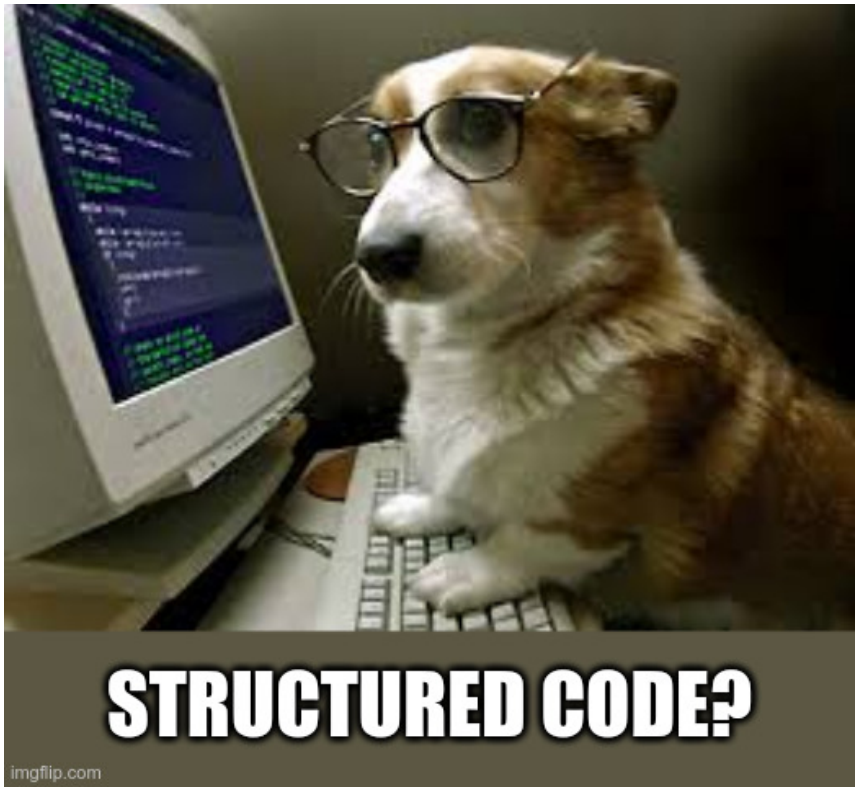
Syllabus

- Elementary **algorithms** (counting, searching, sorting, numerical algorithms).
- **Recursion**. Solving simple problems using iterative and recursive algorithms.



Syllabus

- **Incremental development.** Error detection and correction. Notions of efficiency and correctness.



How will the classes be?

- **Theoretical**: face-to-face classes in auditorium with:
  slides + livecoding + board + visualizations + ...
- **Practical**: class guide (html) + implementing (in C)
  + submitting in Mooshak

Extra material published on website (e.g. tutorials, documentation, ...)

It's important to work outside of classes!

The lessons are "only" the **contact hours**. It's important to be present, but it's not enough just to remember Programming II during the lessons...

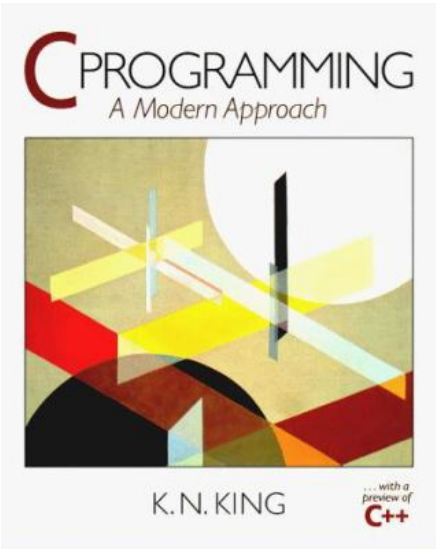
Bibliography

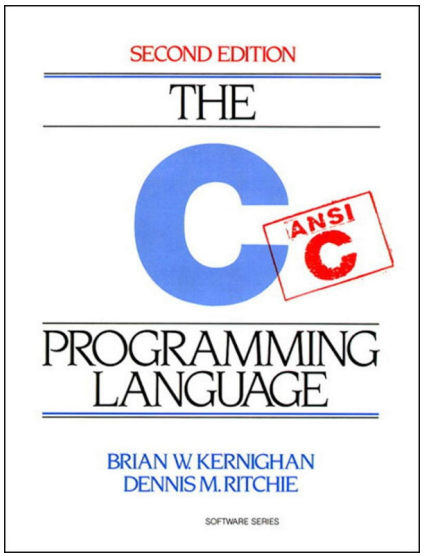
Main Book:

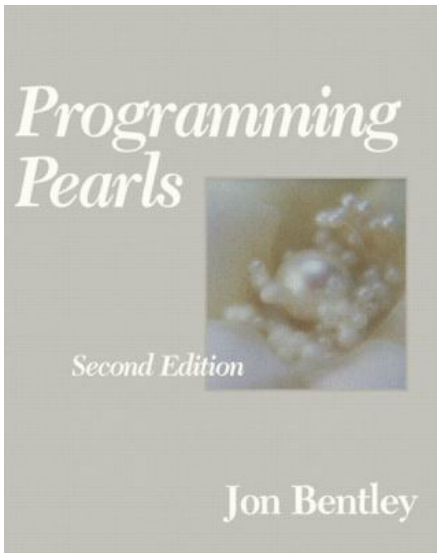
- **C Programming: A Modern Approach** (K. N. King)

Other recommended books:

- **The C Programming Language** (B. W. Kernighan, D, M. Ritchie)
- **Programming Pearls** (Jon Bentley)







Super Mario Effect

The Super Mario Effect: tricking your brain into learning more

(Mark Rober — TEDxPenn)

`https://youtu.be/9vJRopau0g0`

The Super Mario Effect

Focusing on the Princess and not the pits, to stick with a task and learn more