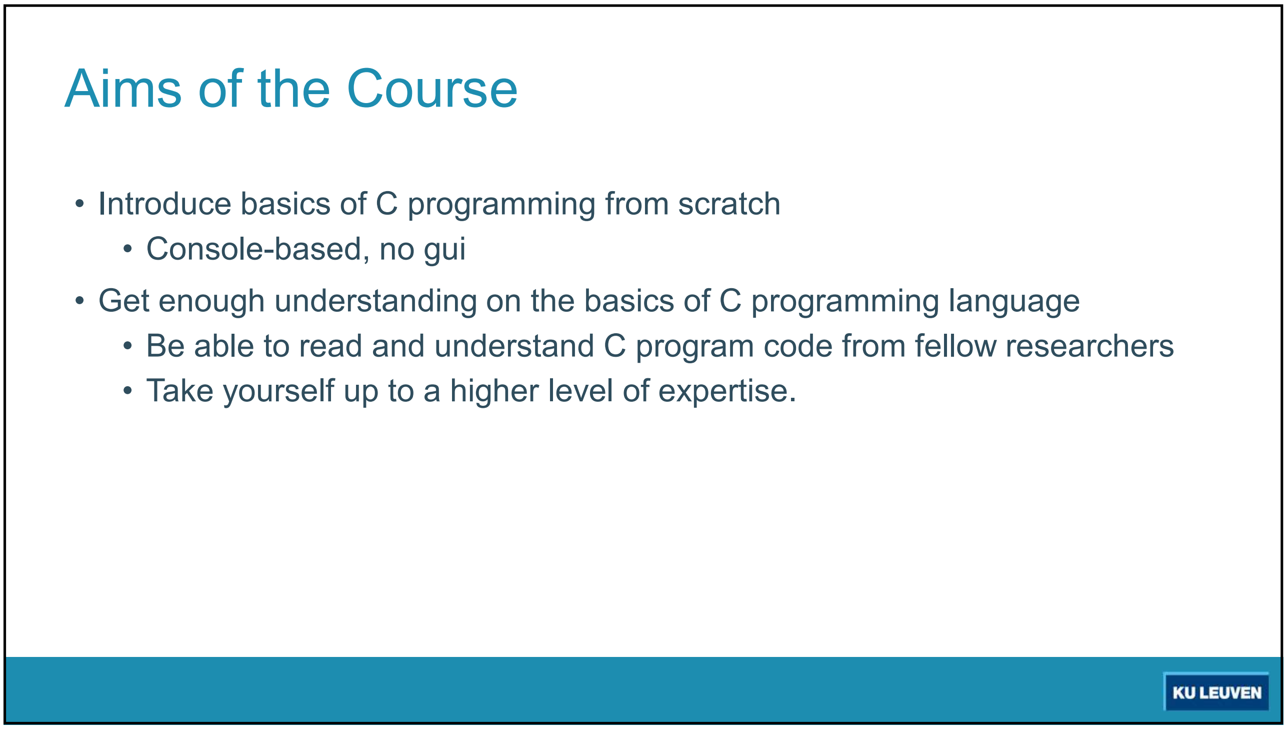
C: an introduction

Topics covered

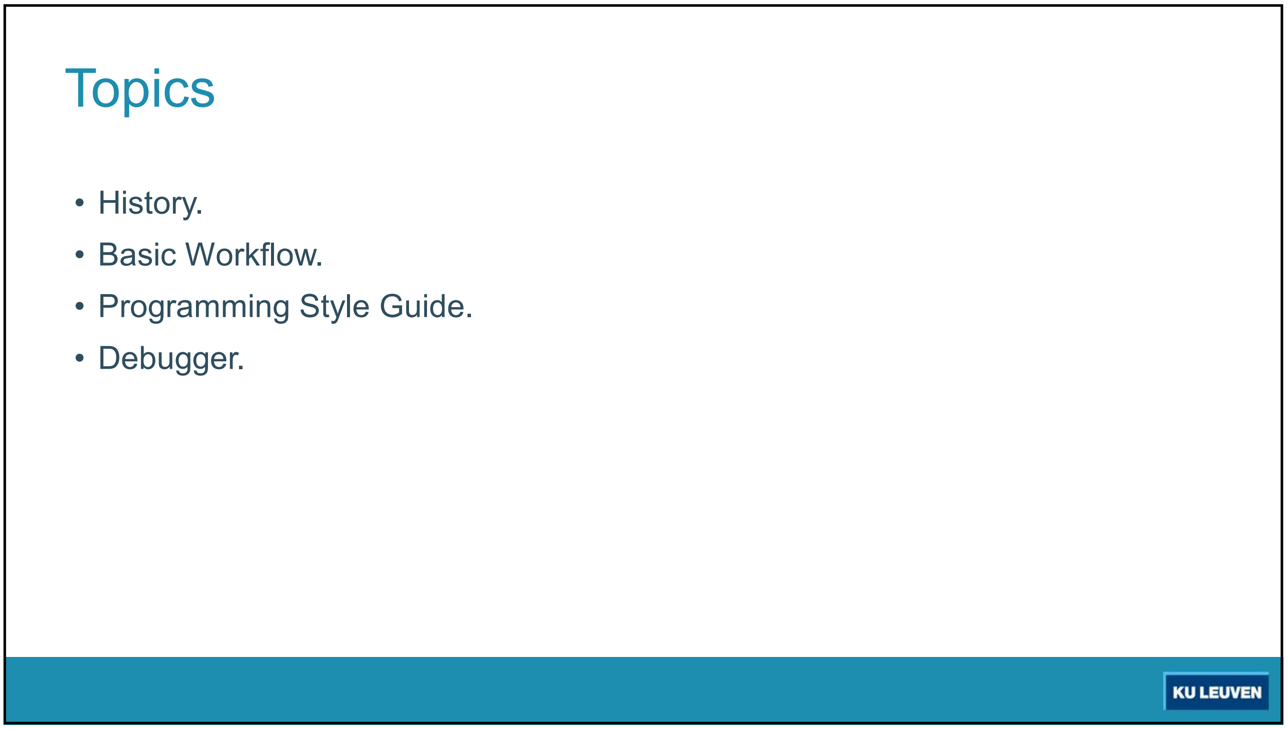
Course documents

- https://franklbvp.github.io/c_intro/



Aims of the Course

- Introduce basics of C programming from scratch
  - Console-based, no gui
- Get enough understanding on the basics of C programming language
  - Be able to read and understand C program code from fellow researchers
  - Take yourself up to a higher level of expertise.



Topics

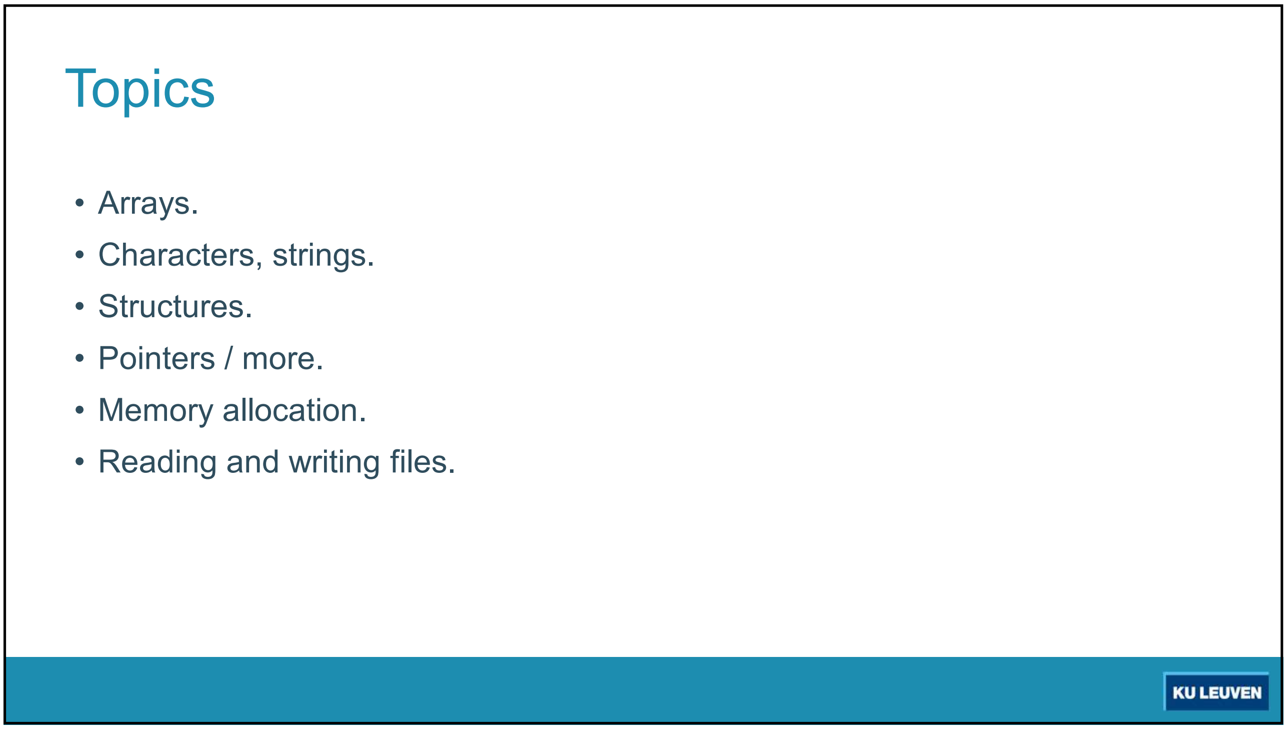
- History.
- Basic Workflow.
- Programming Style Guide.
- Debugger.

Topics

- Operators, operands and their precedence.
- Different number types in C (Integers and Floating Point).
- Conversions and casts.
- Expressions.
- Programming statements:
  - Choice: `if, switch`
  - Loops: `while, do-while, for`

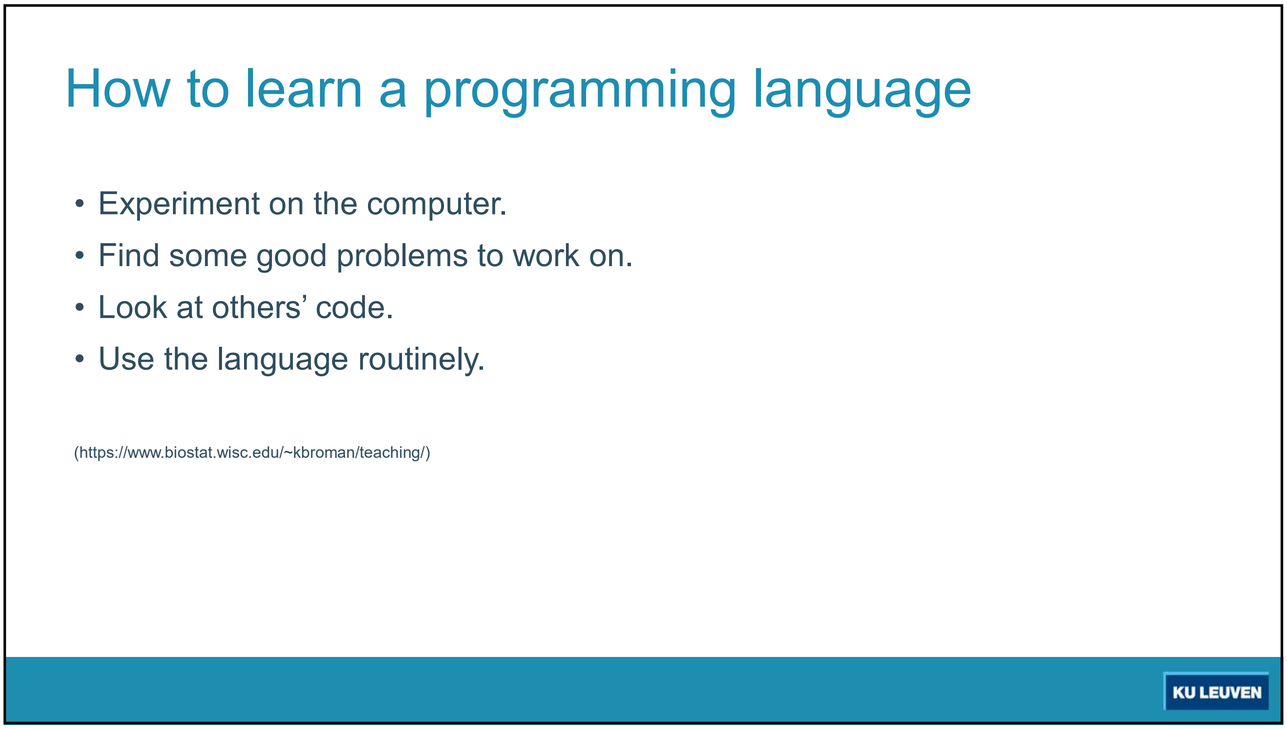
Topics

- Functions.
- Scope.
- Preprocessor.
- Pointers.



Topics

- Arrays.
- Characters, strings.
- Structures.
- Pointers / more.
- Memory allocation.
- Reading and writing files.



How to learn a programming language

- Experiment on the computer.
- Find some good problems to work on.
- Look at others’ code.
- Use the language routinely.

(https://www.biostat.wisc.edu/~kbroman/teaching/)

Programming process

1. Specify the task
2. Discover an algorithm for its solution
3. Code the algorithm (in C)
4. Test the code

(Kelley & Pohl: C by dissection)

Thank you for the information

- Slides based on
  B. Kernighan & D. Ritchie (1988)
  The C Programming Language (ANSI C)
  Prentice Hall Software Series

- https://www.gribblelab.org/CBootCamp/index.html
- https://newton.ex.ac.uk/teaching/resources/jmr/
- https://www.cse.msu.edu/~cse251/
- https://www.csee.umbc.edu/~tinoosh/cmpe311/
- https://www.cs.yale.edu/homes/aspnes/classes/223/notes.html
- https://github.com/angrave/SystemProgramming/wiki
- https://www.zentut.com/c-tutorial/introduction-to-c-language/
- https://overiq.com/c-programming/101/intro-to-c-programming/
- https://github.com/Keith-S-Thompson/how-to-c-response/blob/master/README.md