

# Software Engineering

# Practical organization

# Lectures

- Regular room: (Sarfatti 25, piano terra) **Aula 5**
- Regular time:
  - Wednesdays 10:15 – 12:45
  - Fridays 8:30 – 10:00
- Plus:
  - Wed Sep 11 13:45 – 16:00 (Sraffa 13, piano 3) **Aula N31**
  - Fri Sep 13 13:45 – 16:00 (Sarfatti 25, piano terra) **Aula 5**
  - Fri Sep 20 13:45 – 16:00 (Sraffa 13, piano terra) **Aula N05**

# Contact

- Lecturer:      Laurent Poirrier      `laurent.poirrier@unibocconi.it`
  - drop-in office hours:    **TBD**
  - any other time:            **email me**
  
- TA:                Caicai Chen                `caicai.chen@unibocconi.it`
  - office hours (remote):    **book appointment**

# Material

- No reference book
- Slides will be available
- Some additional resource links will be shared

# Evaluation

- 20% assignments (individual)
- 20%+ project (individual or groups of 2)
- written exam
  - open-book, no devices
  - multiple-choice + open-ended questions

# Course overview

- Part 1: How computers works

- Boolean logic, integers
- Instructions
- Memory

- Part 2: Software development

- Compiling (clang, make, ...)
- Architectures, portability (ABIs, ...)
- Code management (git)

- Part 3: Correctness

- Specifications
- Documentation, testing
- Static & dynamic analysis, debugging

- Part 4: Performance

- CPU pipelines, caches
- Data structures
- Parallel computation



# Project

# Choice of project topic

- Submit your own topic
- Subject to my approval
- There will be a deadline for topic submission (but changes are possible)
- I will make suggestions

# Example topics

- add **features** to an open source project (ideally useful to you, look at e.g. F-Droid apps)
- improve **performance** of an open source project
  - aim for low-hanging fruit
  - performance is not just speed: memory, network data, power
- **find bugs** in an open source project
  - aim for low-hanging fruit
- **fix bugs** in an open source project
  - look at bugzilla, github/gitlab issues
- develop **your own** project (ideally useful to you)

# Project organization

- Individual or groups of two
- I will help you in class and after class

# Project grading

- Overall weight 20% of final grade at least
- More than 20% for outstanding projects
- Write a 1-page report

# Evaluation criteria

- Correctness
- Technical difficulty
- Originality
- Impact and presentation

# Policy for participation in open source projects

- no extra marks for getting “upstreamed”
- you “must” get my approval before contacting project developers  
(email, pull requests, etc.)

# Preparation for tutorials



# Development environment

We will need

- `clang` or `gcc`
- `make`
- `hexdump`
- `objdump`

See also: [List of useful shell commands for this course](#)

# Installing a suitable development environment

- On Windows:

Install [Windows Subsystem for Linux](#) (WSL2, specifically)

- On MacOS:

Install Homebrew: <https://brew.sh/>

- On GNU/Linux, FreeBSD, OpenBSD:

Any distro should work.

