

Analyse et Verification de Logiciel

Intro

Guille Polito – CRIStAL – CNRS – RMoD - Inria Lille
guillermo.polito@inria.fr
@guillep

The Course

- 6 semaines
 - 3 semaines - Testing - G. Polito
 - 3 semaines - Debugging - S. Costiou

<https://github.com/avl-univ-lille/testing>

The Course

- 6 semaines
 - **3 semaines - Testing - G. Polito**
 - 3 semaines - Debugging - S. Costiou

<https://github.com/avl-univ-lille/testing>

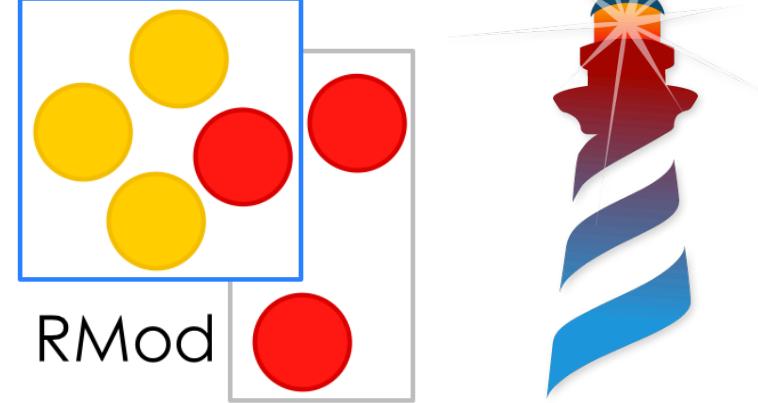
Inria RMod Team

Software Engineering and Language Implementation

- <https://rmod.inria.fr>

The screenshot shows a web browser window with the title bar "RMOD". The address bar contains the URL "https://rmod.gitlabpages.inria.fr/website//index.html#rmod". The page itself is titled "RMOD TEAM" and features a navigation menu with links to "RMoD", "Team Members", "Research", "Publications", "Software", "News", "Jobs", and "Contacts". On the left side, there is a logo consisting of four yellow circles in a square frame, with the word "RMod" written below it. To the right of the logo, the word "RMoD" is written in large, bold letters. Below this, a section titled "Vision" contains the text: "The goal of RMoD is to support ever-running systems. This objective is tackled from complementary perspectives: reengineering of large systems, tools for every programming and infrastructure for dynamic reflective programming languages." At the bottom of the page, there is a bullet point: "• Reverse Engineering & Software Evolution. In this objective we develop".

What do I work on?



- Compilers, interpreters, garbage collectors
- Automated testing
- Programming language design



We look for smart students

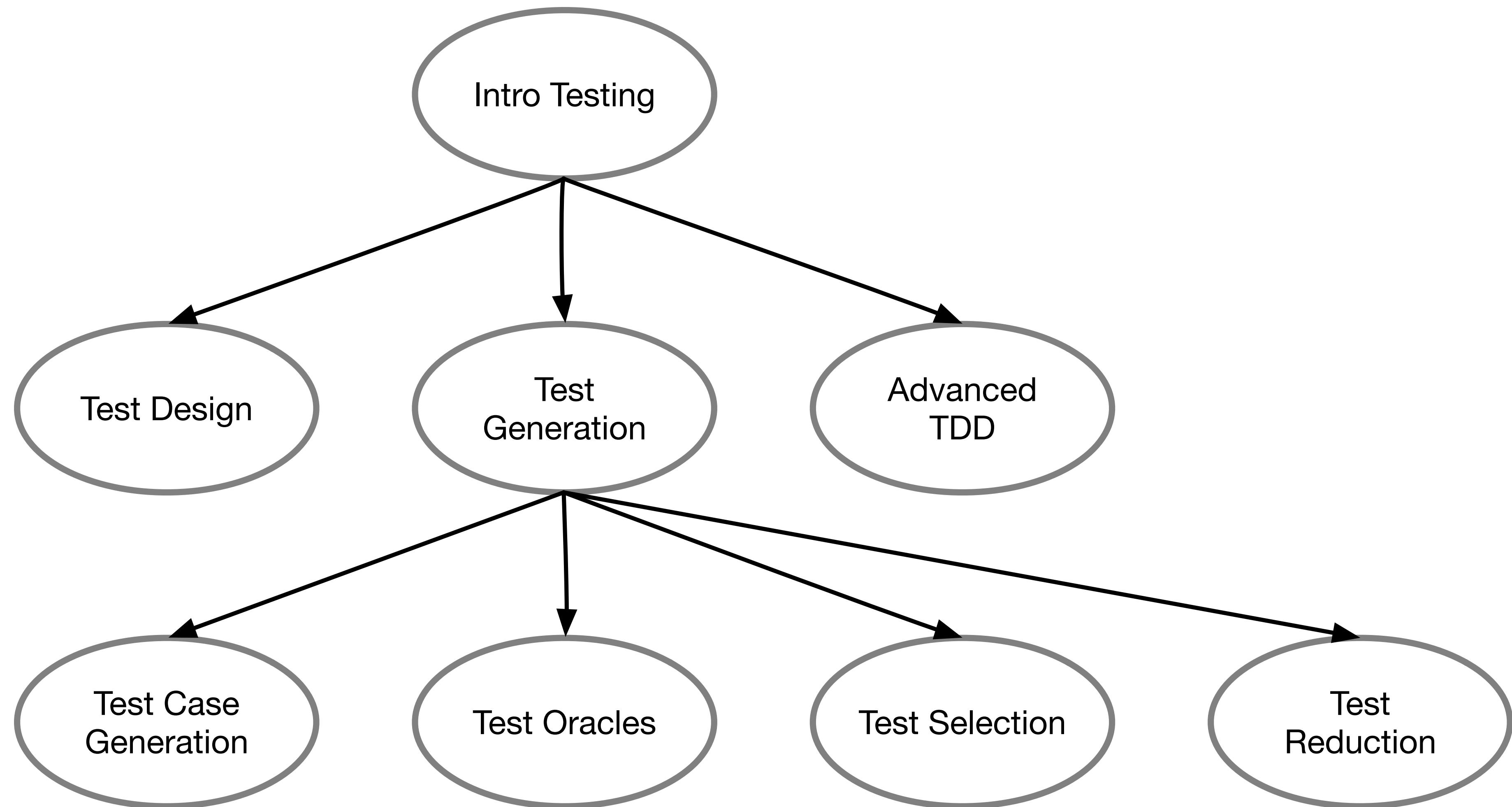
- If you are interested in research:
 - We have internships
 - We have (young) engineering positions
 - We have PhD positions
- Talk to me!

guillermo.polito@inria.fr
- Topics:
 - Code analysis
 - Compilers, interpreters, Virtual Machines
 - Garbage collectors
 - AI implementation/optimisation
 - Benchmarking, profiling
 - Security
 - Testing
 - Programming tools

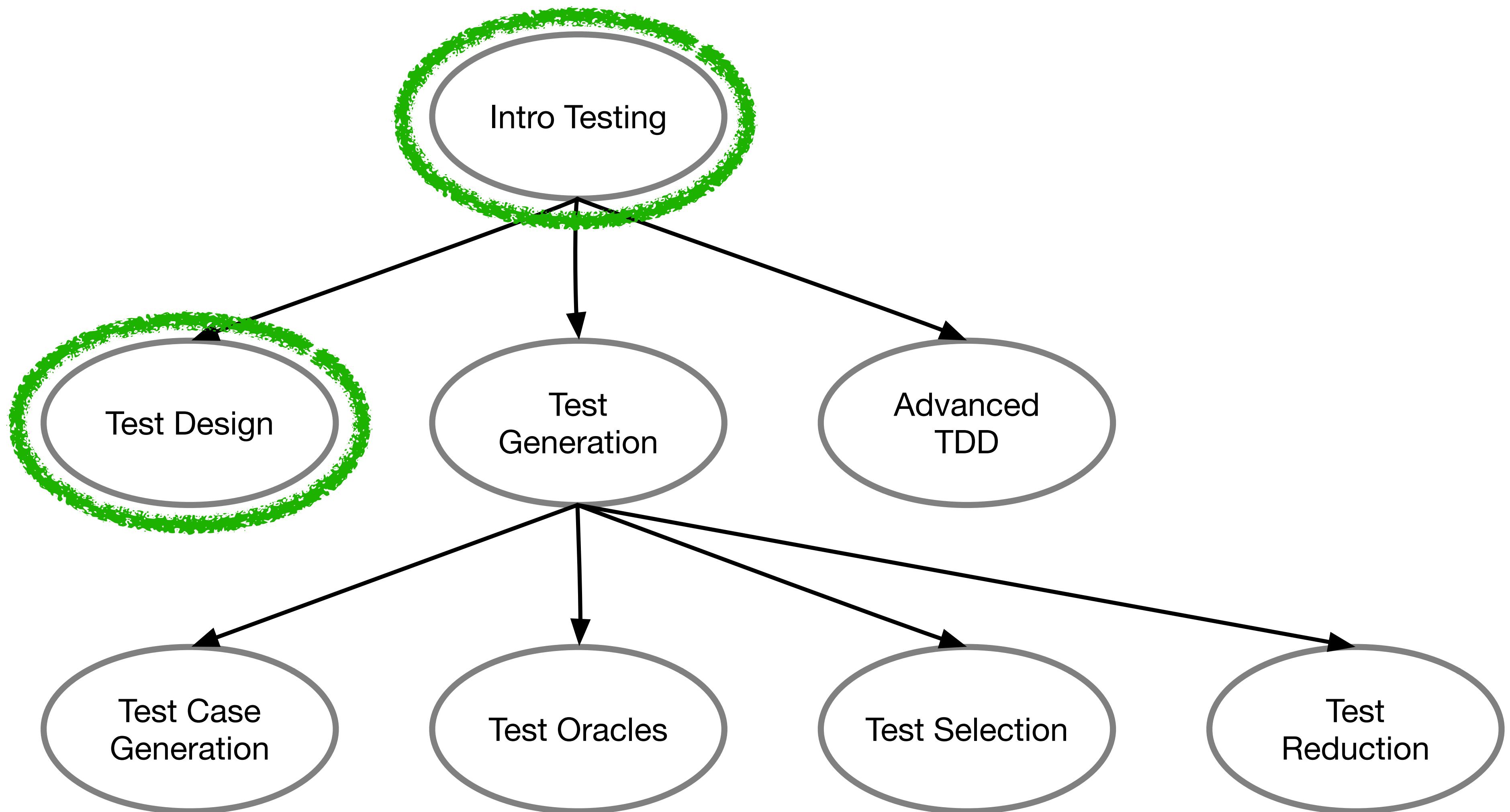
The Course

- 6 semaines
 - **3 semaines - Testing - G. Polito**
 - 3 semaines - Debugging - S. Costiou

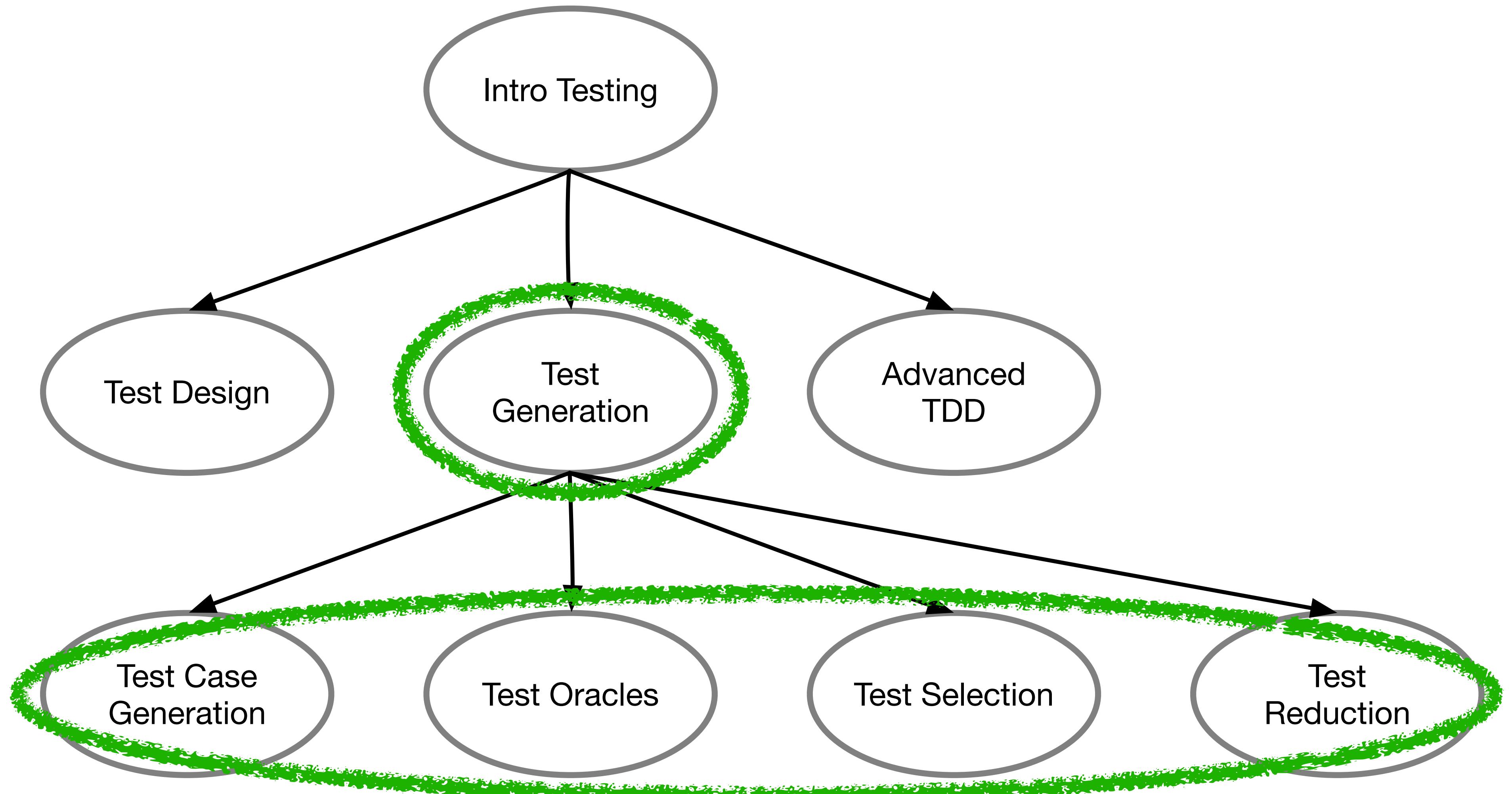
1st Part Overview



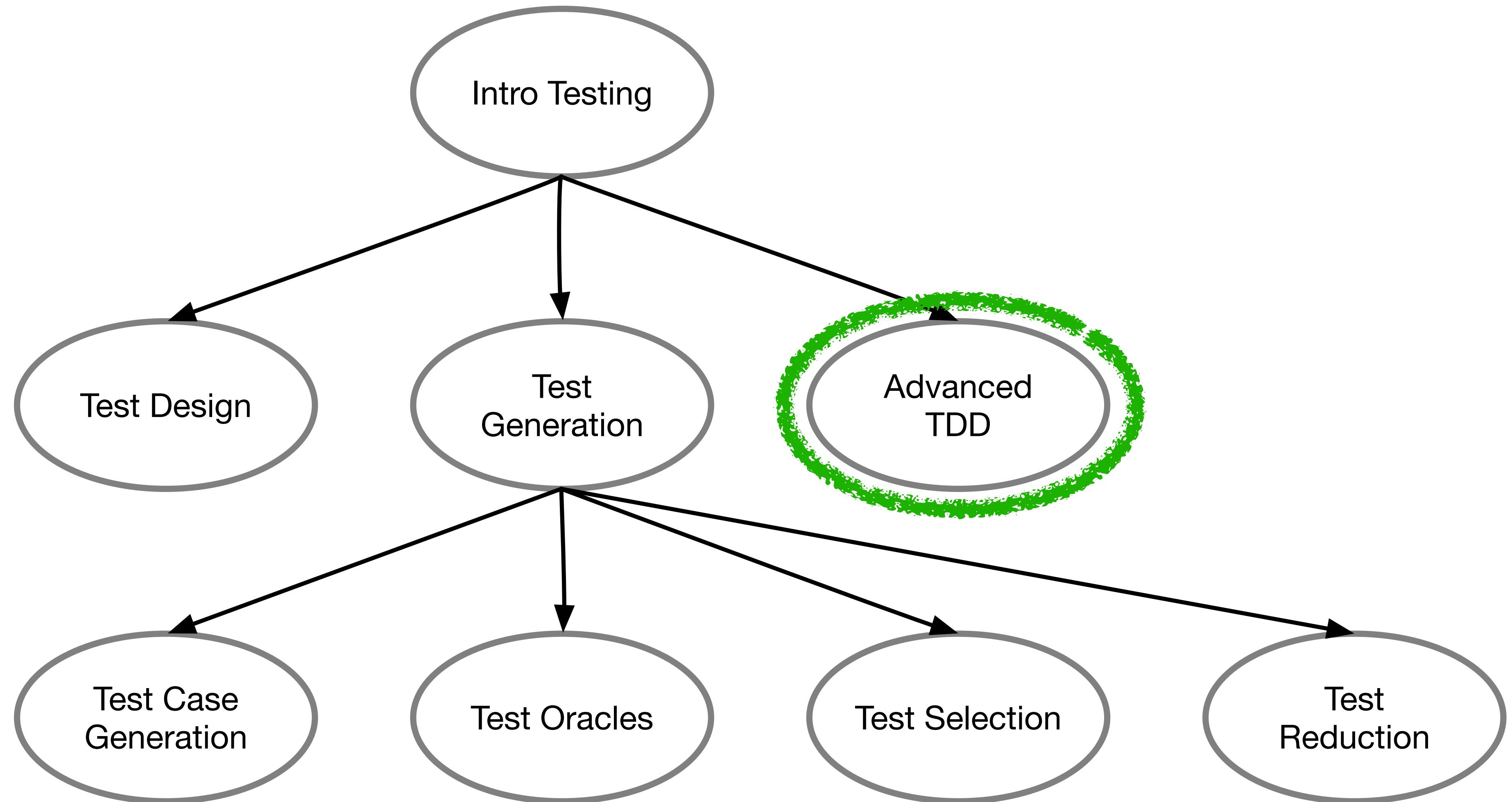
Today, 1st Week: Intro + Design



2nd Week: Test Generation



3rd Week: Advanced TDD



How will we work?

- Each day: theory + practice
- Practice: 2 ***team projects***
 - 1 programming project
 - 1 reading + presentation project

Grading

- 3 grades
 - **Presentation**
 - **Programming project**
 - **Short exam**, the 3rd week. Topic: *test design*

Reading + Presentation Project

- Choose one research paper
- Read it, understand it...
- Make a 15' presentation in the **third week**
- *The strict minimum (for a 10): just explain the paper*
- For **extra grading**, go further: examples, implementations, advances...

Programming Project

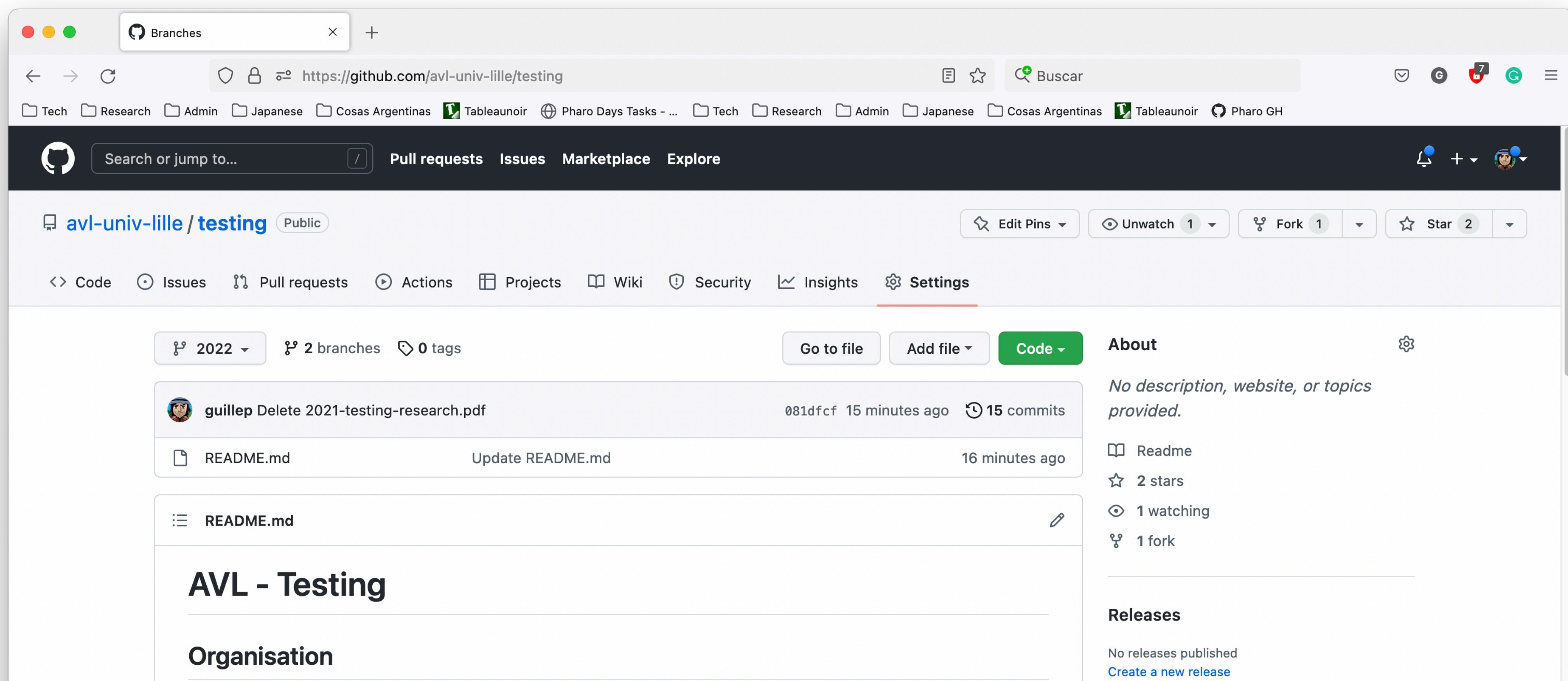
- Choose a topic (e.g., SQLite database, vue.js GUI)
- Program a small application on that topic (e.g., a grocery list)
- **Test it!** Thoroughly, consciously => show you *know* what you are doing
- + Write a report
- *The strict minimum (for a 10): good manual coverage**
- For **extra grading**, go further: implement at least one advanced technique!

Exam

- The 3rd week
- Topic: *test design*
- Two scenarios. Make testing plans.
 - What would you test?
 - What would you not test?
 - What advanced techniques would you use?
 - ...

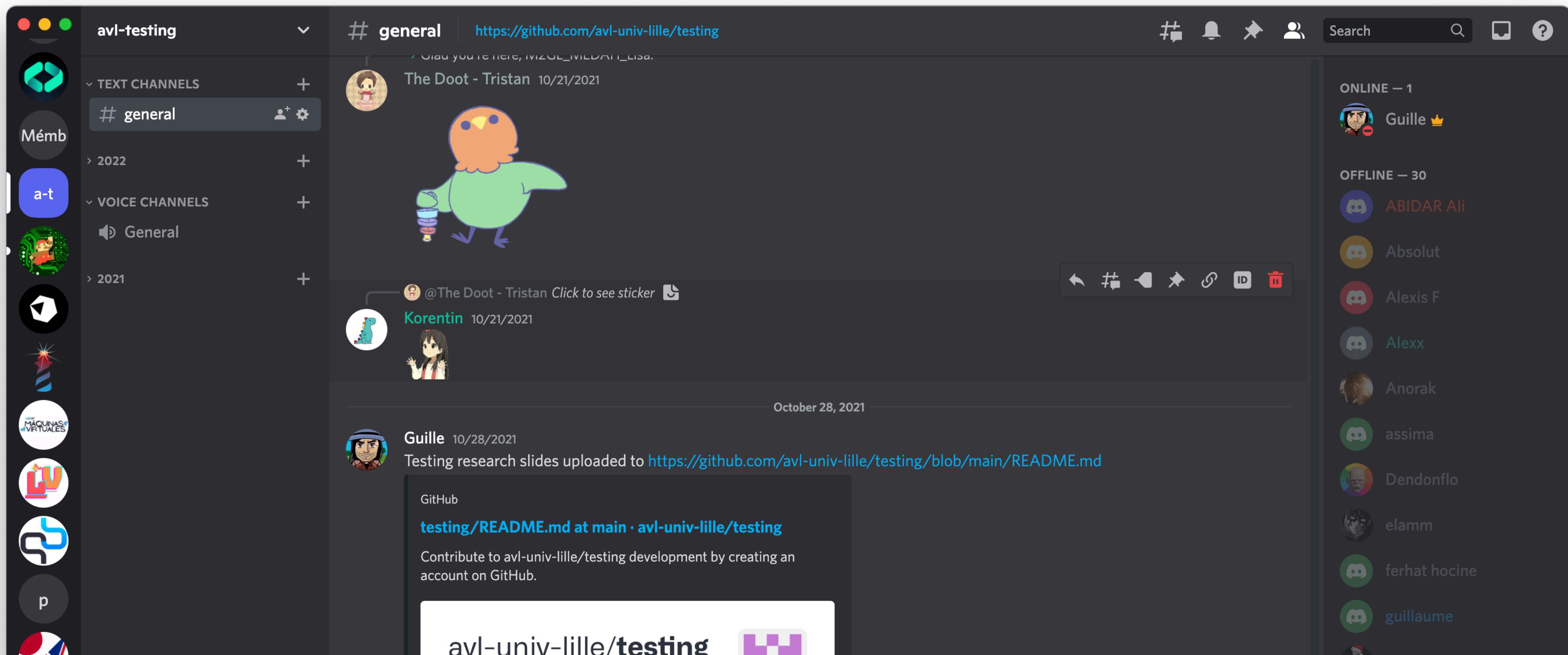
Communication - GitHub

- <https://github.com/avl-univ-lille/testing>



Communication - Discord

- <https://discord.gg/h2mH5Daq> (It's also on GitHub!)



Take away

- 3 grades: project, presentation, exam
 - Presentation + Exam the third week
 - The bare minimum gets a 10
- Presentation + project are in teams
 - Make a team NOW!