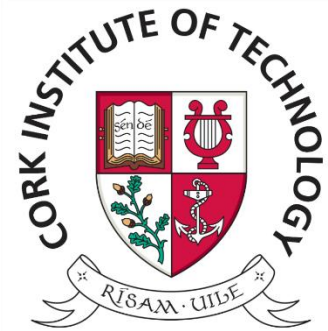
MTU
Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

Programming Language Design

Module Presentation

Francisco Ortin



Department of
Computer Science

Zoom

- For **online lectures**, please follow the next instructions
- If you are **not** enrolled in the module
  1. Join with URL https://cit.instructure.com/enroll/7JNHEM
  2. When asked, download and install Zoom
  3. Test your speaker and microphone
  4. Join the meeting
- If you are enrolled in the module
  1. Log in Canvas https://cit.instructure.com
  2. Click on the module *SOFT9022_Xlist Programming Language Design*
  3. In the Tools unit, click on *Zoom meeting for lectures*
  4. When asked, download and install Zoom
  5. Test your speaker and microphone
  6. Join the meeting

Lecturer

- Francisco Ortin
  - Professor at the University of Oviedo
  - Adjunct Lecturer at MTU / CIT
- If you want to contact me, please use the **Inbox** menu in **Canvas**
  - Please, do **not** send me an email directly to my MTU / CIT email account

Questions / Comments

- Upon lectures, if you have any questions / comments, you either
  - Write them in the **chat**
  - Or use the **microphone**

- Please, **disable your mics while not speaking** to avoid background noise

Module Delivery

- Each **week** there will be
  - An online **lecture** (Wednesdays at 6 pm)
  - An autonomous **lab** (next 7 days, one-week time)
- Use the **discussion forums** in Canvas for any questions about lectures and labs (*Tools* section, *Module discussion forum*)
  - Please, create one topic for each discussion
- **Office hours**
  - Please, arrange an appointment by sending me a message through **Canvas Inbox**
  - We will use Zoom meetings for office hours (available in Canvas, in the *Tools* section)

Learning Outcomes

- Design the **main elements** of a programming language, assigning responsibilities to the different elements of its **architecture**
- Select regular expressions and **context-free grammars** to implement **lexers** and **parsers**
- **Design** Abstract Syntax Trees (AST) with design patterns, considering all the **quality principles** of software engineering
- Evaluate the **foundations** of **language semantics** and **type systems**
- **Generate** and **optimize code** from type-annotated abstract syntax trees

Course Contents

1. Basics of programming language design
2. Lexical analysis
3. Syntactic analysis
4. Semantic analysis
5. Intermediate programming languages and representations
6. Code generation

Teaching Method

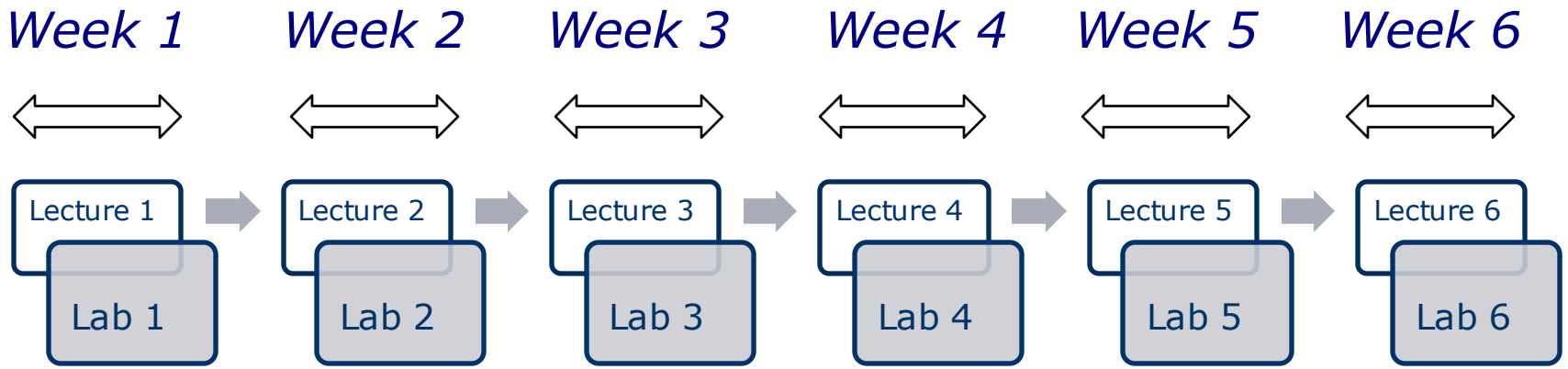
- **Practical** nature of the course (project-based)
- Online **lectures** aimed at solving typical scenarios in language design and compiler construction
- The module is **project-based**
  - Each **lab** is a step forward in achieving the main **course goal**: design and implementation of a programming language
  - Labs are **incremental**: the previous one must be finished to implement the next one
  - The compiler must be developed **individually** (plagiarism will be checked)

Assessment

- **Two exams** (assignments)
  - 1st midterm lab exam (project-based)
    - Lexical and syntax analysis plus AST design
  - 2nd lab exam (project-based)
    - The rest of the language (semantic analysis, runtime environment and code generation)

Midterm Exam

- **Midterm exam**
  - At the end of week 5 (upload it before lecture 6)
  - One week to upload it
  - Work done in labs 1-5
  - Keep your compiler up-to-date



Midterm Exam

- Important: for the **1st exam**, detailed feedback will be given to the student, but a working implementation **will not be provided**
  - Thus, the student must fix/finalize his/her language design and implementation
  - Otherwise, it will make it difficult to continue with the following labs, and hence the second exam

Assessment

- The features of the language to be designed and implemented are described in **lab 02**
  - **Minimum features** are worth 70% tops
  - **Additional features** allow higher marks (up to 100%)
  - In each lab, you could start with the minimum features and then try to add the additional ones

Lecture attendance

- Assignments are based on the labs, so
- Do I really need to attend/watch the lectures?
  - Yes, **attend/watch the lectures before starting the lab**, otherwise you will not be able to do the labs
- Remember, **keep your compiler up-to-date!** (week after week)

Questions

Any questions about the module?