Programming Languages and Object-Oriented Programming (LPO)

DIBRIS - Università di Genova

Anno accademico 2022-2023

Instructors

Davide Ancona office n.**328**
e-mail **davide.ancona@unige.it**

Riccardo Bianchini office n.**309**
e-mail **riccardo.bianchini@edu.unige.it**

Objectives

- Learning the object-oriented and functional programming methodology, both at the conceptual and practical level
- Getting acquainted with object-oriented design and programming patterns
- Acquiring some basic concepts on programming language design and implementation

Motivations

Importance of software productivity, reuse and maintenance

Motivations

- What is the best programming language?
  - The question hardly makes sense in general
  - Many considerations play an important role in the choice of the language:

    application domain, project requirements, developers' skills, software interoperability, compatibility issues, market requests, etc.
- Importance of mastering different computational models, several languages are in fact multi-paradigm

Motivations

- Compilers and interpreters are very basic tools: every computer professional and graduate in computer science/engineering should have a rough idea of how they are designed and implemented
- Domain specific languages (DSL) enhance software productivity, but require skills in programming language implementation to be developed

Motivations

Prepare your future career!

- Demand for developers is high:
  - https://www.zdnet.com/article/want-to-get-hired-in-tech-these-programming-languages-will-get-you-the-most-interviews/
- The Top Programming Languages 2022
  - www.tiobe.com/tiobe-index
  - https://spectrum.ieee.org/top-programming-languages-2022
  - https://www.zdnet.com/article/programming-languages-python-dominates-but-developers-are-adding-new-skills-to-stand-out/

Exams

- Written test (solutions of past exams available on AulaWeb)
- Java project (2-4 students per project) + individual interview
- Partial written test during the Winter term (typically February)
- Written test: 5 exams
- All rules available on AulaWeb
- Requirement: IP exam must be passed first
- Suggested: ASD exam should be passed first

Aulaweb


LINGUAGGI E PROGRAMMAZIONE ORIENTATA AGLI OGGETTI - 61799
Corso
Partecipanti
Valutazioni
Competenze
Introduzione
Minimizza tutto
Orario delle lezioni
aula 506
lunedì 14.15-16.00
giovedì 9.15-11.00
venerdì 14.15-16.00
FORUM
Annunci
FORUM
Forum per discussioni su questioni tecniche od organizzative
PAGINA
Testi consigliati

Schedule

First semester: 4 hours of classes + 2 hours of class or lab per week

	Monday	Tuesday	Wednesday	Thursday	Friday
9-11				class	
11-13					
14-16	class				class/lab
16-18					

Second semester: 2 hours of class or lab per week

Lab

- Activities:
  - 2 OCaml labs
  - 8 Java labs
- Tools:
  - JDK (Java Development Kit)
  - Eclipse/IntelliJ IDEA Integrated Development Environments (IDEs)
  - OCaml interpreter
  - GitHub Classroom
- Groups:
  - groups to be defined asap by using the AulaWeb Wiki

Remarks: the more advanced Java labs will be taken in the second semester

Lab

How to successfully manage the Java project

There are just three very simple rules to follow:

Lab

How to successfully manage the Java project

There are just three very simple rules to follow:

1. **attend the Java labs and complete the home assignments**

Lab

How to successfully manage the Java project

There are just three very simple rules to follow:

1. **attend the Java labs and complete the home assignments**
2. **attend the Java labs and complete the home assignments**

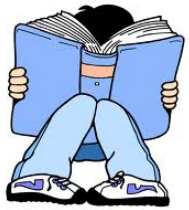
Lab

How to successfully manage the Java project

There are just three very simple rules to follow:

1. **attend the Java labs and complete the home assignments**
2. **attend the Java labs and complete the home assignments**
3. **attend the Java labs and complete the home assignments**

Suggested readings



Basic

- D. Ancona, G. Lagorio, E. Zucca. Linguaggi di Programmazione. Città Studi Edizioni.
- R. Gallardo, S. Hommel, S. Kannan, J. Gordon, S. Biocca Zakhour. The Java Tutorial: A Short Course on the Basics (Sixth Edition), Oracle
- E. Chailloux, P. Manoury, B. Pagano. Developing Applications With Objective Caml. (freely available on the Web)

Advanced

- J. Bloch. Effective Java (third edition). Addison-Wesley Professional
- Y. Minsky, A. Madhavapeddy, J. Hickey. Real World OCaml – Functional programming for the masses (2nd edition in progress). O'Reilly Media
- E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

Technical (useful for consultation only)

- J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley. The Java Language Specification (Java SE, latest edition). Oracle. (freely available on the Web)

AulaWeb feedback for students

Useful to improve the course organization

Sondaggio per gli studenti

Modalità: Anonimo

1. Quante volte ho frequentato LPO
- ○ sto frequentando LPO per la prima volta
- ○ ho già frequentato LPO una volta sola
- ○ ho già frequentato LPO più di una volta

2. Corso di laurea/curriculum
- ○ Informatica (Scuola di Scienze MFN), curriculum propedeutico
- ○ Informatica (Scuola di Scienze MFN), curriculum tecnologico
- ○ Ingegneria Biomedica
- ○ Ingegneria Elettronica
- ○ Ingegneria Informatica
- ○ Matematica
- ○ SMID
- ○ Altri corsi

3. Ho già passato i seguenti esami
- ☐ IP
- ☐ ASD
- ☐ Algebra e Logica per Informatica
- ☐ Esami di programmazione di altri corsi di laurea

4. Ho familiarità con i seguenti linguaggi di programmazione
- ☐ C
- ☐ C++
- ☐ C#
- ☐ Java

Syllabus

A general outline

- Programming language principles – part 1 (semester 1)
  - Syntax: regular expressions, and context-free grammars
- Functional programming in OCaml (semester 1)
- Object-oriented programming and Java (semesters 1 and 2)
- Programming language principles – part 2 (semester 2)
  - Parser implementation
  - Static and dynamic semantics and their implementation

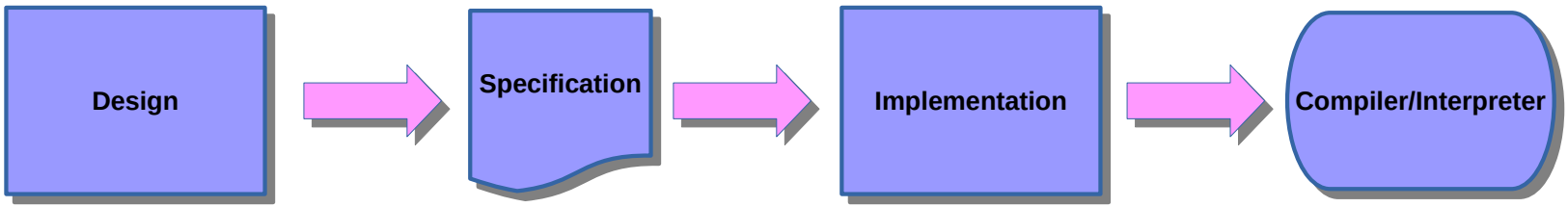
A more detailed description can be found on AulaWeb

Principles of Programming Languages

Goal

Design a programming language and implement it

Remark: here *"programming language"* means higher-level language
no machine code, assembly or intermediate language



Language specification

- language specification = definition of the language
- desiderata: clear, precise, unambiguous
- possible solution: use tools for formal methods

Formal Specification of Programming Languages

Main parts of a programming language specification

- syntax
- static semantics (optional)
- dynamic semantics