# Master of Applied Science & Engineering: Computer Science

## **Option: Software Languages and Software Engineering**

#### **Organised by the Software Languages Lab**



### Software Languages Lab (aka SOFT)



5 fulltime professors4 parttime professors8 post-docs25 pre-docs

#### located @ building F 10th



2

/RIJE

BRUSSEL

**INIVERSITEIT** 

## Who's Who @ SOFT



V. Jonckers



E. Gonzalez Boix



C. De Roover



W. De Meuter



D. Devriese



T. D'Hondt



J. Sartor



J. Nicolay



J. De Koster





#### What is this MA About?

The specialisation "Software Languages and Software Engineering" is about studying the Art, Science and Engineering of software construction.



## **List of Courses**

Name of the Course	Professor	Level	Sem	ECTS	pre/co-req
Higher Order Programming (HOP)	J. Nicolay	boosting	1	6	
Functional Programming (FP)	W. De Meuter	foundations	1	6	НОР
Meta Programming and Reflection	E. Gonzalez Boix	foundations	1	6	НОР
Cloud Computing and Big Data Processing	J. De Koster/ J. Nicolay	foundations	1	6	НОР
Multicore Programming	J. Sartor	broadening	2	6	НОР
Performance Analysis and Evaluation	J. Sartor	broadening	1	6	
Distributed and Mobile Programming Paradigms	E. Gonzalez Boix	broadening	2	6	НОР
Next Generation User Interfaces	B. Signer	broadening	1	6	
Interpretation of Computer Programs 2	W. De Meuter	broadening	2	6	НОР
Compilers	D. Vermeir	broadening	2	6	
Capita Selecta of SE	C. De Roover	broadening	1+2	6	
Software Quality Analysis	C. De Roover	deepening	2	6	
Capita Selecta of PL	D. Devriese	deepening	1+2	6	FP
Programming Language Engineering	T. D'Hondt	deepening	2	6	НОР
Fundamenten van Programmeertalen	C. Scholliers	deepening	1	6	FP

5 VRIJE UNIVERSITEIT BRUSSEL

#### **Higher-Order Programming**

This is a **boosting course** in functional programming that was especially designed for students who already know how to program well in some mainstream language with limited functional characteristics (e.g. Java, C#, Python,...). We advise all students with a non-VUB BA background to take it. It can be combined in the same semester with Functional Programming.

- Course requires manual registration
- Access denied for students with VUB-BA
- Strongly recommended for all other students







#### **Interpretation of Computer Programs 2**

This is a MA-level course that is **also accessible** for VUB students in their **3rd BA year**. Hence, some students may have already taken this course. The course studies advanced programming languages concepts using an **interpreter-based approach**. Topics include interpreters, continuations, type checking, actors, object models, classes, CPScompilers. Higher Order Programming is required.

- Course requires manual registration
- Access denied for VUB-BAs who already took it





#### **Toys @ SOFT 4 Students**









8

VRIJE UNIVERSITEIT BRUSSEL

### **List of Courses**

Name of the Course	Professor	Level	Sem	ECTS	pre/co-req
Higher Order Programming (HOP)	J. Nicolay	boosting	1	6	
Functional Programming (FP)	W. De Meuter	foundations	1	6	НОР
Meta Programming and Reflection	E. Gonzalez Boix	foundations	1	6	НОР
Cloud Computing and Big Data Processing	J. De Koster/ J. Nicolay	foundations	1	6	НОР
Multicore Programming	J. Sartor	broadening	2	6	НОР
Performance Analysis and Evaluation	J. Sartor	broadening	1	6	
Distributed and Mobile Programming Paradigms	E. Gonzalez Boix	broadening	2	6	НОР
Next Generation User Interfaces	B. Signer	broadening	1	6	
Interpretation of Computer Programs 2	W. De Meuter	broadening	2	6	НОР
Compilers	D. Vermeir	broadening	2	6	
Capita Selecta of SE	C. De Roover	broadening	1+2	6	
Software Quality Analysis	C. De Roover	deepening	2	6	
Capita Selecta of PL	D. Devriese	deepening	1+2	6	FP
Programming Language Engineering	T. D'Hondt	deepening	2	6	НОР
Fundamenten van Programmeertalen	C. Scholliers	deepening	1	6	FP

VRIJE

BRUSSEL

UNIVERSITEIT

#### **Capita Selecta of Software Engineering**

#### **New Content in 2018-2019**

This course is about **creating truly intelligent development tools** by applying advanced machine learning and data mining algorithms to software engineering data. Topics include predicting the location of defects in source code using classifiers such as Support Vector Machines, prioritizing test cases using Genetic Algorithms, and uncovering repetition in commit histories using Pattern Mining.





10

### Capita Selecta of Programming Languages

#### **New Content in 2018-2019**

A major recent scientific achievement in programming languages research is the development of the first realistic verified C compiler CompCert: a compiler that comes with a computer-verified rigorous correctness proof. In this course, we study basic concepts of verified compilers: both conceptually (what does it mean for a compiler to be correct) and practically (how does one build such a proof in practice). To do this, we first teach the basics of the main tool in this field: dependently-typed programming languages (in our case Agda). In addition to provably correct compilers, we also look at provably secure compilers. This course is taught in a hands-on way, with students learning to program and prove in Agda and studying and extending important compiler passes. Evaluation is done through a course project (with an oral defence) in which students apply the concepts and techniques learned. Functional Programming is a prerequisite for this course, but this course starts a bit later in the semester, so both courses can be followed in parallel.





11