UCLouvain

lsinf2335

2020

Programming paradigms

Due to the COVID-19 crisis, the information below is subject to change, in particular that concerning the teaching mode (presential, distance or in a comodal or hybrid format).

5 credits	30.0 h + 15.0 h	Q2

Teacher(s)	Mens Kim ;
Language :	English
Place of the course	Louvain-la-Neuve
Main themes	In the course of a career, a computer scientist or software engineer will be confronted with many different programming languages and paradigms. To make informed design choices when selecting a particular language, he or she must understand the principles underlying how programming language features are defined, implemented and used. This course will examine, from a historical perspective, the guiding principles of the major programming paradigms, starting from the earliest programming languages until the most recent ones. As such it will highlight the major principles, strengths and differences of different programming languages and paradigms.
Aims	Given the learning outcomes of the "Master in Computer Science and Engineering" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: • INFO1.2 • INFO2.4-5 • INFO6.3-4 Given the learning outcomes of the "Master [120] in Computer Science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: • SINF1.M2-3 • SINF2.4-5 • SINF6.3-4 1 Students completing this course successfully will be able to: • describe and differentiate the main programming paradigms (including procedural programming, functional programming, logic programming, object-oriented programming, concurrent programming, as well as more recent programming paradigms) • determine what paradigm a programming language belongs to; • identify and discuss the design principles of a given language or paradigm; • choose a language or paradigm suitable for solving a particular problem and argue this choice; • write small programs in a selection of the different languages and paradigms seen in the course; • place a programming language in relation to others from a historical perspective; • compare different programming languages from the point of view of their underlying design principles; • understand the impact of different language design choices (syntax, parameter passing, scoping, abstraction, '). - - - - *The contribution of this Teaching Unit to the development and command of the skills and learning outcomes of the programme(s) can be accessed at the end of this sheet, in the section entitled "Programmes/courses offering this Teaching Unit".*
Evaluation methods	**Due to the COVID-19 crisis, the information in this section is particularly likely to change.** Throughout the year, in parallel with the theory and lab sessions, the students will study in detail (either individually or in pairs) several of the languages seen in the course, by carrying out three programming missions in three different languages. These missions will be evaluated through interviews and presentations to the professor and the course assistant. This evaluation replaces the traditional course exam. In case of doubt about the final grade, the teacher reserves the right to ask a student to pass a complementary oral exam.
Teaching methods	**Due to the COVID-19 crisis, the information in this section is particularly likely to change.** The course will consist of traditional theory sessions in which the characteristics and guiding principles of different programming languages and paradigms are explored in detail. The practical sessions complement these more theoretical course sessions with hands-on programming exercises in a selection of programming languages and paradigms seen in the theory course.

Inline resources	Moodle course website The course slides as well as other relevant and practical information related to the course will be accessible on Moodle. The same platform will also be the means of communication between the teacher(s) and the students.
Bibliography	**Références** Comme les langages étudies peuvent varier d'un année à un autre, les références conseillés pour ce cours pourront varier également. Néanmoins, une référence très utile qui couvre un large éventail de langages de programmation reste : o "Principles of Programming Languages - Design, Evaluation and Implementation" par Bruce J. MacLennan. **References** As the programming languages studied in this course may vary from year to year, the recommended references for this course may also vary. Nevertheless, a very useful reference which covers a wide range of programming languages remains: o "Principles of Programming Languages - Design, Evaluation and Implementation" by Bruce J. MacLennan.
Other infos	Background : • Having a healthy interest in programming language concepts, such as for example seen in the courses LINFO1104 and LINFO1131. • The more different programming languages a student has been confronted with before, the more he or she will appreciate this course.
Faculty or entity in charge	INFO

Programmes containing this learning unit (UE)				
Program title	Acronym	Credits	Prerequisite	Aims
Master [120] in Computer Science and Engineering	INFO2M	5		
Master [120] in Computer Science	SINF2M	5		