UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Programming Languages

2223-2-E3101Q108

Aims

The students will familiarize with various programming paradigms, in particular the logical and functional ones. They will learn the most common programming environments for the languages presented during the course. The students will acquire the ability to develop small and medium sized projects in (Common) Lisp and Prolog, by using the programming environments presented.

Contents

The course will provide students with a panoramic view of the main programming language paradigms: imperative, logic (declarative) and functional.

Detailed program

1. Programming languages paradigms: imperative, logical (declarative) and functional. Notions recall of “run-time”, program execution on an idealized stack architecture. Basic notions about programming environments for the different systems presented. Some historical perspectives.
2. The logic programming paradigm. Introduction to the Prolog programming language.
3. The functional programming paradigm. Introduction to the (Common) Lisp programming language. Note on Haskell and Julia
4. The imperative programming paradigm. Introduction to the C/C++ programming language.
5. Usage of the different paradigms in a variety of contexts.

Prerequisites

Basic concepts of recursive programming, mathematical logic and hardware/software architecture.

Teaching form

The course will be offered as a set of standard lectures during the term. “e-Learning” support will also be provided for the distribution of course material, exercises and project descriptions. Moreover there will be laboratory sessions where the students will learn to program with the programming languages introduced during the course.

Textbook and teaching resource

- Leon Sterling and Ehud Shapiro, The Art of Prolog Advanced Programming Techniques - 2nd Edition;
- Abelson, Sussman e Sussman, Structure and Interpretation of Computer Programs (SICP);
- Peter Seibel, Practical Common Lisp (PCL);
- Brian W. Kernigham & Dennis M. Ritchie, C Programming Language (2?? Edition), Prentice Hall, 1988;
- Robert Sedgewick, Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms (3rd Edition), Addison Wesley, 2001

Semester

First semester

Assessment method

The Programming Languages exam requires passing a written test and a programming project. At the discretion of the instructors an oral exam may be also required. The written exam and the project corresponding to the first roll-call are organized into two "partial" tests.

The partial tests are two written exams offered in November (Logic Programming and Prolog) and in January/February (Functional Programming and Lisp; Imperative Programming and C/C++), plus one or two projects, posted towards the end of October, to be electronically handed-in in January/February. The partial tests, as already said, comprise the first roll-call for the course and are not held over for the remaining roll-calls.

From the next roll-call (February/March) onward, the written exam will comprise all the course materiale. The same applies for the projects. The following roll-calls will be in June, July and September (in dates to be announced).

The written exam and the projects must be passed after the same roll-call; the final grade is the weighted average, at the discretion of the instructors, of the partial grades. In particular, a weighted arithmetic average equal to 18 does not guarantee the passing of the exam.

Projects can be done in groups (max 3 people), but must be handed-in individually. Whoever works on a project alone or in a pair, may have a bonus.

The final grade is in thirtieths, and expresses a comprehensive evaluation of everything that concurs to the achievement of the stated educational goals. That is, it is the result of a comprehensive evaluation of the tests. E.g., clarity, rigor, judgment abilities, argumentation choices and ability to illustrate them in an efficacious way.

Final note: every exam result will be recorded in the system: grade, withdraw, absent, insufficient, etc.

Office hours

Prof. Marco Antoniotti: TBD or by appointment via email.

Prof.ssa Gabriella Pasi: by appointment via email.

Prof. Fabio Sartori: by appointment via email.

Sustainable Development Goals