

# Declarative Programming and (Co)Induction

**Duration:** 20 hours

**Teachers:**

Davide Ancona – DIBRIS, University of Genova - [davide.ancona@unige.it](mailto:davide.ancona@unige.it)

Elena Zucca - DIBRIS, University of Genova - [elena.zucca@unige.it](mailto:elena.zucca@unige.it)

**When:** April 18-22, 2016

**Where:** DIBRIS - Via Dodecaneso, 35

**Abstract**

The course is a self-contained introduction to functional programming, logic programming, and the use of inductive and coinductive methods in programming languages. Both theoretical (inference systems, induction and co-induction, lambda calculus, type system and semantics) and practical (languages Haskell and Prolog) aspects will be covered concerning well-founded and non-well-founded entities.

**Program**

The course is organized in two modules.

- First module (10 hours): inductive definitions and proofs by induction, small step and big step semantics, lambda calculus, inductive type, system, soundness. Functional programming in Haskell.
- Second module (10 hours): induction and coinduction, lowest and greatest fixed points, proofs by induction and coinduction, abstract and operational semantics of Prolog and coprolog. Programming in Prolog and coProlog.

**References**

- Benjamin C. Pierce. Types and Programming Languages. The MIT Press.
- Leon S. Sterling and Ehud Y. Shapiro. The Art of Prolog, second edition. The MIT Press.
- Simon Peyton Jones. Haskell 98 language and libraries: the Revised Report. Cambridge University Press.