Advanced Concepts in Programming Languages

Prerequisites	Programming Languages (DM509)
Form	Individual Study Activity (reading course with practical projects)
Credit	5 ECTS
Evaluation	Pass/Fail based on participation and projects
Textbooks	none – articles, tutorials, and notes

Advanced Concepts in Programming Languages

Prerequisites	Programming Languages (DM509)
Form	Individual Study Activity (reading course with practical projects)
Credit	5 ECTS
Evaluation	Pass/Fail based on participation and projects
Textbooks	none – articles, tutorials, and notes
Lecturer	Peter Schneider-Kamp (new at IMADA from January 1, 2009)

Advanced Concepts in Programming Languages

Advanced Concepts in Imperative Programming

Advanced Concepts in Programming Languages

Advanced Concepts in Imperative Programming

- functional constructs in python

  use map, filter, reduce, lambda, partial to write better Python code

  ```
  2*reduce(lambda x,y: x+y, map(int, '1 2 3 4 5 6'.split()))
  ```

Advanced Concepts in Programming Languages

Advanced Concepts in Imperative Programming

- functional constructs in python™

  use map, filter, reduce, lambda, partial to write better Python code

  2*reduce(lambda x,y: x+y, map(int, ['1', '2', '3', '4', '5', '6']))

Advanced Concepts in Programming Languages

Advanced Concepts in Imperative Programming

- functional constructs in python™

  use map, filter, reduce, lambda, partial to write better Python code

  2*reduce(lambda x,y: x+y, [1, 2, 3, 4, 5, 6,])

Advanced Concepts in Programming Languages

Advanced Concepts in Imperative Programming

- functional constructs in python™

  use map, filter, reduce, lambda, partial to write better Python code

  2*(1+2+3+4+5+6)

Advanced Concepts in Programming Languages

Advanced Concepts in Imperative Programming

- functional constructs in python™

  use map, filter, reduce, lambda, partial to write better Python code

  42

Advanced Concepts in Programming Languages

Advanced Concepts in Imperative Programming

- functional constructs in python™

  use map, filter, reduce, lambda, partial to write better Python code

  2*reduce(lambda x,y: x+y, map(int, '1 2 3 4 5 6'.split()))

- Single Assignment C SAC www.sac-home.org

  functional array programming for high-performance computing

  matrixProduct = {[i,j] -> sum(A[[i,.]] * B[[.,j]])};

Advanced Concepts in Programming Languages

Advanced Concepts in Imperative Programming

- functional constructs in python

  use map, filter, reduce, lambda, partial to write better Python code

  ```
  2*reduce(lambda x,y: x+y, map(int, ’1 2 3 4 5 6’.split()))
  ```

- Single Assignment C

  functional array programming for high-performance computing

  ```
  matrixProduct = {[i,j] -> sum(A[[i,.]] * B[[.,j]])};
  ```

- generic types in Java

  polymorphic data types in mainstream Java

  ```
  public class Pair<T,U> { public T x; public U y; }
  ```

Advanced Concepts in Programming Languages

Advanced Concepts in Logic Programming

- foreign language interface for Prolog

  interface Prolog with Java using InterProlog

- meta programming in Prolog

  assert, retract, and clause for implementing expert systems

- database queries using Datalog

  restricted Prolog as a query language for databases

- writing parsers in Prolog

  declarative clause grammars

Advanced Concepts in Programming Languages

Advanced Concepts in Functional Programming

- combined functional and logic languages

  the functional logic language Curry as an extension of Haskell

- parallel programming in Haskell

  exploit multiple processors/machines using parallel/distributed Haskell

- web interfaces using Haskell

  use the cgi and xhtml modules to create web interfaces

- graphical user interface in Haskell

  use Java Swing and the LambdaVM

Research Topics

Possible Areas for Bachelor or Master Theses

- verification of Java, Haskell, and Prolog programs
  - termination analysis
  - correctness, liveness, and safety
- programming languages
  - optimizing compilers
  - virtual machines
- software development tools
  - push-button verification tools
  - integrating verification tools into IDEs
- constraint solving
  - satisfiability solving and optimization
  - constraint solving over finite domains

Advanced Concepts in Programming Languages

Prerequisites	Programming Languages (DM509)
Form	Individual Study Activity (reading course with practical projects)
Credit	5 ECTS
Evaluation	Pass/Fail based on participation and projects
Textbooks	none – articles, tutorials, and notes
Lecturer	Peter Schneider-Kamp (new at IMADA from January 1, 2009)

Advanced Concepts in Programming Languages

Prerequisites	Programming Languages (DM509)
Form	Individual Study Activity (reading course with practical projects)
Credit	5 ECTS
Evaluation	Pass/Fail based on participation and projects
Textbooks	none – articles, tutorials, and notes
Lecturer	Peter Schneider-Kamp (new at IMADA from January 1, 2009)