Advanced Object-Oriented Programming

Prof. dr. Kris Luyten

Dries Cardinaels

Gilles Eerlings

What can you expect?

- You will become more efficient as a software developer
  - Better structuring/modeling pf code
  - Solving more complex problems with code
- You will master various programming techniques (closure, multi-threading, meta-programming,...)
- You will be able to learn new languages more quickly
- A fast pace with material that builds on solid programming knowledge
  - Being proficient in Python and C++ is a prerequisite

What do we expect?

- Enthusiasm for programming and experimentation
- Inclination for initiative and independent work
- Ability is as important as knowledge (but that doesn’t make knowledge less important)
- Being able to write elegant, robust, extensible, readable, safe and correct code without using an LLM

Programming requires "training", lots of training.

What should you do?

- *Be present* in classes
- *Take notes* during lessons
- *Practice, practice, practice* – also in other programming courses
- Refresh previous programming courses if necessary
- Self-reliance & critical reflection

What should you definitely not do?

- Leave everything until the end
- Ask questions *before searching for answers yourself*
- Think it’s not your fault, because it usually is…

Course overview: themes

- Programming and Programming Paradigms
- OO structures and patterns
- **Defensive programming**
- Multithreading
- Meta-programming
- Closure
- Patterns
- ...

Course overview: assessment

- Written exam: 60%
- "Challenges" (practicals): 40%

Course organization and Communication

- bb.uhasselt.be
- Individual Github account
  - for your exercises and all other work related to this course. Use specified file or folder names (e.g. **[Assignment_01_01]**).
  - Teaching team also has access and follows your progress.
  - Also maintain a backwards planning/burndown chart for all your study-related activities (cf. project skills).

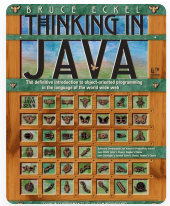
Course organization and Communication

- Discord channel (invite link coming soon):
  - Best to use a clear nick name.
  - This way we can maintain contact more easily outside of class moments.
  - Help each other, that often works better.
- Questions are *not asked via email*: via Discord or during class.
- There are lectures, but no scheduled tutorials (only when necessary).

Course materials

- Made available per lesson / via the web
- For external sources (such as websites), it will be clearly indicated what you need to know, be able to do, or know **and** do.
- Assignments are integrated in the slides.

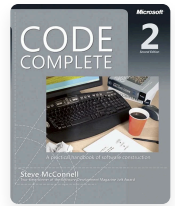
Course materials



Thinking in Java (3rd ed.!)

Bruce Eckel

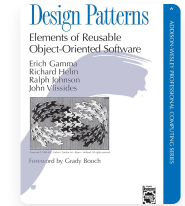
Pearson



Code Complete: A Practical Handbook of Software Construction (2nd ed.)

Steve McConnell

Microsoft Press



Design Patterns: Elements of Reusable Object-Oriented Software

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

Addison-Wesley

Questions?

Programming languages

- **Languages**: Postscript, Oberon, Tcl/Tk, Fortran, Prolog, Pascal, Delphi, Python, Cobol, Modula, Ada, Rexx, ISO C, C#, Javascript, ANSI/ISO C++, Ruby, Self, Eiffel, PHP, Perl, ML, Lisp, Objective-C, VB, Scheme, Haskell, Caml, Smalltalk, Miranda, PL/1, Simula, Java...
- **Main classes**: imperative, functional, object-oriented and logic

Programming languages

- **Languages**: Postscript, Oberon, Tcl/Tk, Fortran, **Prolog**, Pascal, Delphi, Python, Cobol, Modula, Ada, Rexx, ISO C, C#, Javascript, ANSI/ISO C++, Ruby, Self, Eiffel, PHP, Perl, ML, Lisp, Objective-C, VB, Scheme, Haskell, Caml, Smalltalk, Miranda, PL/1, Simula, Java...
- **Main classes**: imperative, functional, object-oriented and **logic**

Logic programming

- Programming based on logic
- Prolog
  - Predicate logic
  - Declarative (*what* not *how*)
  - Facts and clauses (rules)
  - Used in e.g. expert systems, data analysis and semantic web

Logic programming: Prolog

facts

```
parent(Frans, Eefje).
parent(Klaar, Eefje).
parent(Eefje, Salammbo).
parent(Eefje,Mattho).
parent(Gustave, Salammbo).
parent(Gustave,Mattho).
```

Clause defines a relation

```
grandparent(X,Y) :-
    parent(X,Z),
    parent(Z,Y).
```

Querying a Prolog program

```
? – parent(Eefje, Mattho).
yes
? – grandparent(Frans,Salammbo).
yes
? – grandparent(Klaar, X).
X = Salammbo.
X = Mattho.
```

Logic programming: Prolog

- Various interpreters and compilers: SWI-Prolog, GNU-Prolog, Sicstus Prolog, Amzi! Prolog,...
- [not required] Want to learn more yourself? https://www.krisluyten.net/teaching/programming-technologies/
- "Adventure in Prolog" is also an excellent resource to get started: http://www.amzi.com/AdventureInProlog/

Programming languages

- **Languages**: Postscript, Oberon, Tcl/Tk, Fortran, Prolog, Pascal, Delphi, Python, Cobol, Modula, Ada, Rexx, ISO C, C#, Javascript, ANSI/ISO C++, Ruby, Self, Eiffel, PHP, Perl, **ML**, **Lisp**, Objective-C, VB, **Scheme**, **Haskell**, Caml, Smalltalk, Miranda, PL/1, Simula, Java...
- **Main classes**: imperative, **functional**, object-oriented and logic

Functional programming

- Uses functions (but those based on Lambda calculus)
- But program has no *state*
- No use of "destructive updates" (e.g. variable updates) == *pure* functional
- Functional programming languages are particularly suitable for processing and manipulating *lists*
- [not required - Just FYI] Interested in more? https://www.krisluyten.net/teaching/programming-technologies/

Programming languages

- **Languages**: Postscript, Oberon, Tcl/Tk, Fortran, Prolog, **Pascal**, Delphi, Python, Cobol, Modula, **Ada**, Rexx, **ISO C**, C#, Javascript, ANSI/ISO C++, Ruby, Self, Eiffel, PHP, Perl, ML, Lisp, Objective-C, VB, Scheme, Haskell, Caml, Smalltalk, Miranda, PL/1, Simula, Java...
- **Main classes**: **imperative**, functional, object-oriented and logic

Programming languages

- **Languages**: Postscript, Oberon, Tcl/Tk, Fortran, Prolog, Pascal, Delphi, Python, Cobol, Modula, Ada, Rexx, ISO C, **C#**, Javascript, **ANSI/ISO C++**, Ruby, Self, **Eiffel**, PHP, Perl, ML, Lisp, **Objective-C**, VB, Scheme, Haskell, Caml, Smalltalk, Miranda, PL/1, Simula, **Java**...
- **Main classes**: imperative, functional, **object-oriented** and logic

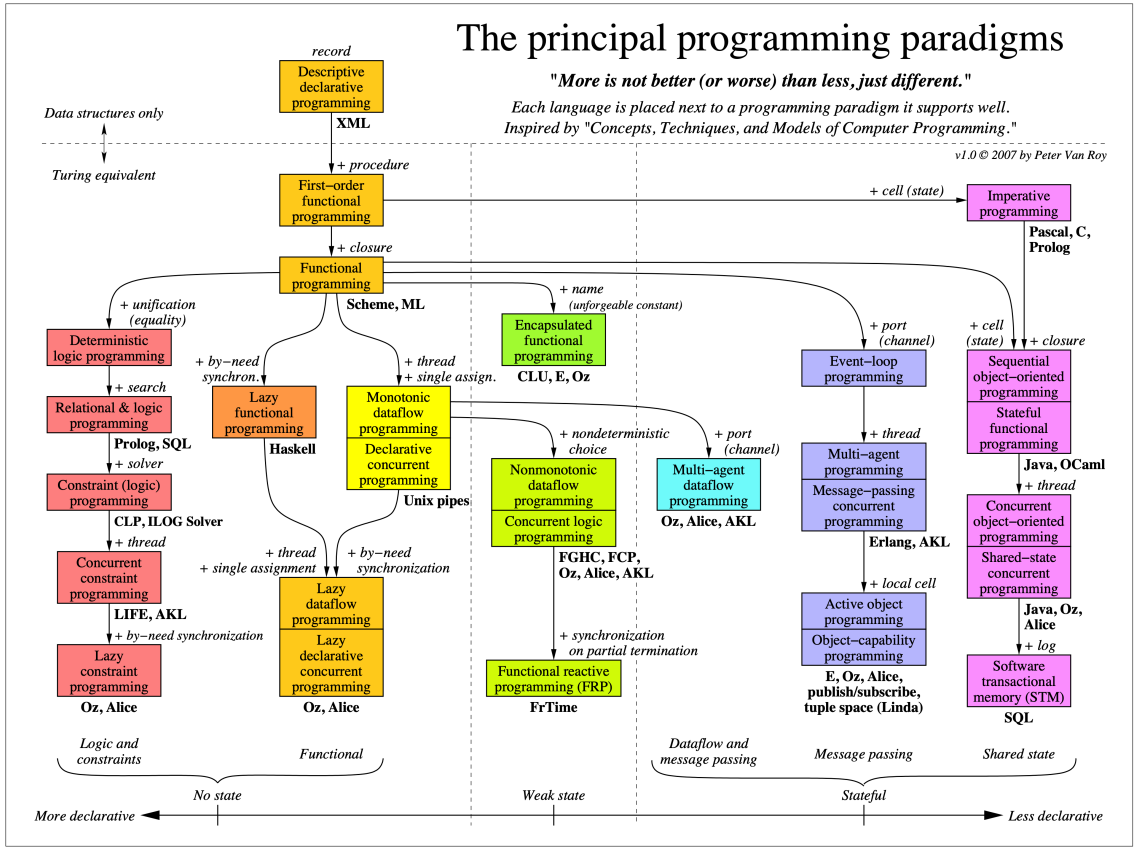
Programming languages

- **Assignment_01_01** What is a programming language? Describe what a programming language is based on your current knowledge!
- **Assignment_01_02** What is a programming paradigm? Style of programming that influences the way code execution happens and that is characterized by the elements used in the programming language (with/without variables, functions, objects,...)

Multi-Paradigm Programming Languages

- Programming languages that support multiple paradigms
- Combinations of procedural+OO (e.g. C++) and logic+functional (e.g. Mercury) are common
- Python

There is much, much more to say about Paradigms...



Peter Van Roy's Programming Language Paradigms Overview

Questions?