Course introduction

Functional Programming 2019/20

Alejandro Serrano

Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Welcome to **Functional Programming 2019/2020**

Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

The team

Matthijs Vákár and Frank Staals (me) in the lectures

- You can find Matthijs in BBG-5.75 and me in BBG-4.15
- Lectures are held in English

8 teaching assistants in the labs

- Most of them are Dutch speakers

Guest lecture at the end of the course

Our aim

Teach you *functional programming* techniques!

Schedule

Lectures: twice per week

- Tuesday, 9.00 to 10.45
- Thursday, 13.15 to 15.00
- 15-minute break in the middle

Werkcolleges : Tuesday, 11.00 to 12.45

Labs : Thursday, 15.15 to 17.00

Schedule

Lectures: twice per week

- Tuesday, 9.00 to 10.45
- Thursday, 13.15 to 15.00
- 15-minute break in the middle

Werkcolleges : Tuesday, 11.00 to 12.45

Labs : Thursday, 15.15 to 17.00

- You are expected to *work at home/library/café/…*

Communication channels

`http://www.cs.uu.nl/docs/vakken/fp`

- All important information is found there
- Schedule, slides, assignments, exercises

E-mail for important news

- Check your UU-mail regularly

Resources

1. **Slides** contain most of the content
   - In some cases, supplemented by additional material
2. Pen-and-paper **exercises** !!!!!
   - There's more than programming in this course
   - Ask questions during werkcollege sessions !
   - Remember: there is *no compiler* at the exam
3. Recommended reading: *Programming in Haskell* (2nd edition) by Graham Hutton
   - The course follows it, except for chapters 13 and 17
   - Not mandatory, but good as extra help
   - More resources can be found in the website

Practical assignments

1. The first one helps you getting started
2. Three small ones with DOMJudge, one per week
3. One bigger project at the end

DOMJudge assignments

- Submissions are **individual**
  - Do not plagiarize!
- Graded mostly automatically, almost instant output
- Grading criteria:
  - Correctness
  - Style

Style checks

- Automatic checks for good style
  - Integrated in DOMJudge
  - Important part of the final project grade
- Some functions are **forbidden**
  - For example, `head`, `tail`, and `fromJust`
  - Using any of them = failing the assignment
- Ask TAs for advice, during labs
  - Off-labs advice on style restricted, and only on demand

Final project

Develop your own **game** in Haskell

- Work in **pairs**
- 80% of your grade for practicals
- Submission in two parts
  1. Preliminary design document
  2. Code of the project

Tools

- Haskell as a programming environment
  - We use GHC, the de facto standard compiler
  - More information later
- HLint to check style
- Two different systems for submission
  - *DOMJudge* for automatic grading
  - *Blackboard* for final project

Rooms for labs and werkcollege

Bring your own laptop policy

- Groups based on last name:
  - Group 1: A - F
  - Group 2: G - K
  - Group 3: L - S
  - Group 4: T - Z
- You can **switch** with somebody else
- After a few weeks, we shall reconsider the space
  - We might drop the group(s) far away.

Optional assignment

Learn and explain a Haskell library or language feature by means of a short video.

- Up to additional 0.5 points for the final grade
- Work in groups of at most three
- More details after mid-term exam

Grading

Linear combination of three grades

- *Theory* T = 0.3 × midterm + 0.7 × final
- *Practical* P = 0.2 × DOMJudge + 0.8 × final
- *Optional* assignment O

Final grade F = 0.5 × T + 0.5 × P + 0.05 × O

To pass the course, you need

- F >= 5.5, T >= 5, P >= 5
- Pass at least two DOMJudge assignments

All other cases are described in the website

If you did the course last year

- **Resubmit** your DOMJudge assignments
- Redo the **final project**
  - Using the same code as last year is *not* allowed
- Redo **all** the **exams**