# Programming Paradigms

Course introduction

# Course structure

| Week | Date | Topics | | Test | Homework Assignment | Materials | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Lecture | Lab | | | Lecture | Lab | Self-study | Links |
| 1 | 2021.10.18 | Programming paradigm. Declarative vs imperative | Lambda calculus recap | — | | TaPL 5.1,5.3 | TaPL 5.2 | Racket Essentials 4.1–4.2 | |
| | 2021.10.20 | Functional programming. Scheme. Expressions, pairs and lists | Recursion over lists in Racket | 2% | | SICP 1.1.5, RE 4.3–4.4 | — | SICP 1.2, ex 1.11, 1.14, 1.16, 1.26 | |
| 2 | 2021.10.25 | Higher-order functions. Mapping and folding lists | Practice with lists, nested lists and trees | 2% | HA #1 | SICP 2.1–2.2 | TBA | TBA | https://racket-lang.org |
| | 2021.10.27 | Functional programming in other languages | More practice with Racket | 2% | | TBA | TBA | TBA | |
| 3 | 2021.11.01 | Haskell. Static types and pure functions | Basic Haskell with pictures | 4% | | TBA | TBA | TBA | |
| | 2021.11.03 | Algebraic data types. Parametric polymorphism | Practice with simple ADTs | 2% | | TBA | TBA | TBA | |
| 4 | 2021.11.08 | Input and output in Haskell | Practice on separating pure functions from IO | 2% | | TBA | TBA | TBA | https://code.world/haskell |
| | 2021.11.10 | Lazy evaluation in Haskell. Laziness and ADT | Practice on wholemeal programming with lazy lists | 2% | HA #2 | TBA | TBA | TBA | |
| 5 | 2021.11.15 | Typed functional programming in other languages | More practice with Haskell | 2% | | TBA | TBA | TBA | |
| | 2021.11.17 | Prolog. Clauses, programs, queries. Unification | Recusion and lists in Prolog | 4% | | LPN 1–4 | TBA | TBA | |
| 6 | 2021.11.22 | Backtracking, cuts and negation | Practice cuts and negation in Prolog | 2% | HA #3 | LPN 10 | TBA | TBA | http://www.let.rug.nl/bos/lpn/<br>https://swish.swi-prolog.org |
| | 2021.11.24 | Logic programming in other languages. List comprehension | More practice with Prolog | 2% | | TBA | TBA | TBA | |
| 7 | 2021.11.29 | Extra topic 1. Pure object-oriented programming. Smalltalk. Io. | Practice | 4% | | TBA | TBA | TBA | https://iolanguage.org |
| | 2021.12.01 | Extra topic 2. Array programming. APL | Practice | | | TBA | TBA | TBA | https://tryapl.org |
| 8 | 2021.12.06 | Extra topic 3. Differentiable programming. Julia | Practice | | | TBA | TBA | TBA | https://juliapackages.com/p/zygote |
| | | | | | | | | | |
| | 2021.12.10 | Exam (written admission test + oral) — 80% | | | | | | | |

| Abbreviation | Full title | |
|---|---|---|
| TaPL | Types and Programming Languages | (has Russian translation) |
| SICP | Structure and Interpretation of Computer Programs | (has Russian translation) |
| RG | Racket Guide | |
| LYAH | Learn You a Haskell for Great Good! | (has Russian translation) |
| LPN | Learn Prolog Now! | |

https://docs.google.com/spreadsheets/d/1VPd8rdu_5SfPqgZrTYCrGrIKd2mOKsxOI_J7EIrarHQ/edit?usp=sharing

# Course outline

- Course introduction and lambda calculus recap
- Functional programming in Racket (Lisp/Scheme dialect)
- Typed functional programming in Haskell
- Logic programming in Prolog
- Extra topics

# Course outline

- Course introduction and lambda calculus recap
- Functional programming in Racket (Lisp/Scheme dialect)
- Typed functional programming in Haskell
- Logic programming in Prolog
- Extra topics

# Course grading scheme

The following contributes to your final grade:

- Tests (at the beginning of every lecture) — 30%
- Final Exam — 80%
- TA bonus points — 5%

Grading policy:

| | |
|---|---|
| **A** | **≥ 85 %** |
| **B** | **≥ 70 %** |
| **C** | **≥ 55 %** |
| **D** | **< 55 %** |