

# Programming in Haskell

127252



## Lecturer in Charge



Izv. prof. dr. sc.  
Jan Šnajder

## Course Description

This is an introductory course to Haskell, an advanced purely-functional programming language. Functional programming languages allow for the development of robust, concise programs in a short amount of time. The key advantages are higher-order functions as an abstraction mechanism and an advanced type system for safety and reusability. The course teaches basic functional programming in Haskell and the basic functional programming concepts. The course starts with language constructs and data types, moves to higher-order functions and algebraic data types, and concludes with IO data streams and monads. Students gain practical experience by solving numerous programming tasks given during the lectures and as take-home assignments. The goal of the course is not only to teach a new programming language, but also to teach a new programming paradigm and a radically different and mathematically sophisticated approach to programming.

## Learning Outcomes

On successful completion of the course, students will be able to:

1. Define the basic concepts of functional programming in Haskell
2. Explain the syntax and semantics of a Haskell program
3. Use Haskell to solve simple practical problems
4. Apply functional idioms and functional design patterns
5. Use available programming libraries to solve complex problems
6. Compare alternative Haskell programs to determine which are better according to selected criteria
7. Design Haskell programs

## General Competencies

Students will learn the basic concepts of functional programming in Haskell, know and understand the basics of syntax and semantics of Haskell, they will be able to recognize and apply functional idioms and functional design patterns, use Haskell to solve simple practical problems, find and use available programming libraries to solve complex problems, compare alternative Haskell programs to determine which are better according to selected criteria, and they will learn the basics of software development in Haskell.

## Forms of Teaching

- » Lectures
  - » Three hours lecture per week for 15 weeks.

ECTS Credits 4

English Level L3

E-learning Level L1

Study Hours

Lecturers 30

Laboratory exercises 15

### Grading

This is a pass/fail course.

This course is not graded. To pass the course students must attend all lectures and practical sessions and hand in all assignments on time.

CEA

EEST

EPE

ECE

EL

IP

TI

WT

SEIS

CE

CS

- » Laboratory Work
  - » Each two-hour lecture is intertwined with one-hour practical session.
- » Programming Exercises
  - » Students will receive homework programming assignments, which they must demonstrate to the instructor or lab assistant.

## Grading System

Type	Continuous Assessment		Exam	
	Threshold	Percent of Grade	Threshold	Percent of Grade
Laboratory Exercises	50 %	70 %	0 %	0 %
Seminar/Project	50 %	30 %	0 %	0 %
Mid Term Exam: Written	50 %	0 %	0 %	

## Comment:

The midterm does not contribute to the total score, but students need to solve at least 50% of the midterm correctly to be able to proceed to the second half of the course.

## Week by Week Schedule

1. Introduction to functional programming. GHC compiler and other on-line resources.
2. Basic language constructs. Tuples and lists.
3. Types and type classes.
4. Syntax of functions.
5. Recursive functions. Corecursive functions.
6. Higher-order functions 1.
7. Higher-order functions 2.
8. Custom data types 1.
9. Custom data types 2. Modules.
10. Custom data types 3. Standard library.
11. Input-output operations 1.
12. Input-output operations 2. Random number generator.
13. Monads and monadic programming 1.
14. Monads and monadic programming 2.
15. Strictness. Code documentation. Packages.

## Literature



Miran Lipovača (2011). *Learn You a Haskell for Great Good!: A Beginner's Guide*, No Starch Press



Graham Hutton (2007). *Programming in Haskell*, Cambridge University Press



Bryan O'Sullivan, Don Stewart, John Goerzen (2008). *Real World Haskell*, O'Reilly Media



Simon Thompson (1999). *Haskell: The Craft of Functional Programming*, Addison Wesley



Paul Hudak (2000). *The Haskell School of Expression: Learning Functional Programming through Multimedia*, Cambridge University Press

## Similar Courses

- » Functional Programming, Chalmers University
- » Functional Programming, Stanford
- » Functional Programming, Oxford
- » Functional Programming, Lund University