TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Module Handbook

TUK MODHB Homepage

Module INF-36-51-M-5

Functional Programming (M, 8.0 LP)

Module Identification

Module Number	Module Name	CP (Effort)
INF-36-51-M-5	*Functional Programming*	8.0 CP (240 h)

Basedata

CP, Effort	8.0 CP = 240 h
Position of the semester	1 Sem. in SuSe
Level	[5] Master (Entry Level)
Language	[DE/EN] German or English as required
Module Manager	Hinze, Ralf, Prof. Dr. (PROF | DEPT: INF)
Lecturers	Hinze, Ralf, Prof. Dr. (PROF | DEPT: INF)
Area of study	[INF-SE] Software-Engineering
Reference course of study	[INF-88.79-SG] M.Sc. Computer Science
Livecycle-State	[NORM] Active

Courses

Type/SWS	Course Number	Choice in Module-Part	SL	PL	CP	Sem.
4V+2U	**INF-36-51-K-5**	P	U-Schein	PL1	8.0	SuSe

- About **[INF-36-51-K-5]**: Title: "Functional Programming"; Presence-Time: 84 h; Self-Study: 156 h
- About **[INF-36-51-K-5]**: The study achievement **"[U-Schein] proof of successful participation in the exercise classes (ungraded)"** must be obtained.
  - It is a prerequisite for the examination for PL1.

Examination achievement PL1

- Form of examination: **written or oral examination**
- Examination number: 63651 ("Functional Programming")

Type of examination will be announced in the lecture. Duration of the examination: ref. examination regulations.

Evaluation of grades

The grade of the module examination is also the module grade.

Contents

From [INF-36-51-K-5] Functional Programming:

Functional programming is a style of programming that emphasises the use of immutable datatypes and pure functions. Functional programming has a simple mathematical basis that supports equational reasoning about properties of programs. As a consequence, functional programs are easier to develop and reason about than their imperative counterparts. The aim of these lectures is to illustrate these points using the standard functional language Haskell.

- Programming with expressions and values
- Types and polymorphism
- Lists and list-processing functions
- Algebraic datatypes
- Higher-order functions
- Type classes
- Equational Reasoning and calculations
- Evaluation orders
- Imperative Programming
- Applicative functors and monads
- Type and class system extensions
- Generic programming

Competencies / intended learning achievements

After successfully completing the module, students will be able to

- solve standard programming problems in Haskell,
- make educated assessments of the benefits of value-orientation,
- use and deploy concepts such as recursive datatypes, higher-order functions, polymorphism, and type classes,
- explain Haskell's approach to integrating side-effects via applicative functors and monads.

Literature

From [INF-36-51-K-5] Functional Programming:

- Lipovaca, Miran. Learn you a haskell for great good! A Beginner's Guide . No Starch Press, 2011.
- Bird, Richard. Thinking functionally with Haskell . Cambridge University Press, 2014.
- Hudak, Paul. The Haskell School of Expression: Learning Functional Programming through Multimedia. Cambridge University Press. 2000.
- Hutton, Graham. Programming in Haskell (2nd Edition). Cambridge University Press, 2016.
- O'Sullivan, Bryan, John Goerzen, and Donald Bruce Stewart. Real world haskell: Code you can believe in . Reilly Media, Inc., 2008.
- Simon Thompson, Haskell: The Craft of Functional Programming (3rd Edition). Addison-Wesley Professional, 2011.

Requirements for attendance of the module (informal)

None

- Notice: Some Courses have informal requirements for attendance:
  - #A: [INF-36-51-K-5] Functional Programming (4V+2U, 8.0 LP) (P: Obligatory)

Requirements for attendance of the module (formal)

None

References to Module / Module Number [INF-36-51-M-5]

Course of Study	Section	Choice/Obligation
[INF-88.79-SG] M.Sc. Computer Science	[Specialisation] Specialization 1	[WP] Compulsory Elective

Module-Pool	Name
[INF-SE_Ba_V-MPOOL-4]	Specialization Bachelor TA Software Engineering
[MV-MB-INF-2022-MPOOL-6]	Wahlpflichtmodule M.Sc. Maschinenbau mit angewandter Informatik 2022
[MV-MBINFO-MPOOL-6]	Wahlpflichtmodule Maschinenbau mit angewandter Informatik