TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Module Handbook

TUK MODHB Homepage

Module INF-02-40-M-2

Programming 1 (M, 14.0 LP)

Module Identification

Module Number	Module Name	CP (Effort)
INF-02-40-M-2	*Programming 1*	14.0 CP (420 h)

Basedata

CP, Effort	14.0 CP = 420 h
Position of the semester	1 Sem. in WiSe/SuSe
Level	[2] Bachelor (Fundamentals)
Language	[DE] German
Module Manager	Liggesmeyer, Peter, Prof. Dr. (PROF | DEPT: INF, GS)
Lecturers	Hinze, Ralf, Prof. Dr. (PROF | DEPT: INF) Liggesmeyer, Peter, Prof. Dr. (PROF | DEPT: INF, GS)
Area of study	[INF-PFL] Mandatory Modules
Reference course of study	[INF-82.B16-SG] B.Sc. Socioinformatics
Livecycle-State	[NORM] Active

Courses

Type/SWS	Course Number	Choice in Module-Part	SL	PL	CP	Sem.
4V+4U	INF-02-01-K-2	P	U-Schein	PL1	10.0	WiSe
2V+1U	INF-02-02-K-2	P	UK-Schein	no	4.0	SuSe

- About **[INF-02-01-K-2]**: Title: "Foundations of Programming"; Presence-Time: 112 h; Self-Study: 188 h
- About **[INF-02-01-K-2]**: 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.
- About **[INF-02-02-K-2]**: Title: "Modelling of Software Systems"; Presence-Time: 42 h; Self-Study: 78 h
- About **[INF-02-02-K-2]**: The study achievement **"[UK-Schein] proof of successful participation in the exercise classes (incl. written examination, ungraded)"** must be obtained.

Examination achievement PL1

- Form of examination: **written exam (Klausur) (150-180 Min.)**
- Examination Frequency: each semester
- Examination number: 60201 ("Foundations of Programming")

Evaluation of grades

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

Contents

From [INF-02-01-K-2] Foundations of Programming:

Syntax of programming languages:

- Concrete and abstract syntax
- Static and dynamic semantics
- Truth trees
- Regular expressions and grammars
- Lexical analysis and syntax analysis

Functional programming concepts:

- Primitive data types
- Records and variants
- Declarations
- Higher order functions
- Parametric polymorphism

Imperative programming concepts:

- Input and output
- Control structures
- References
- Exception handling
- Basics of memory management

Object-oriented programming:

- Objects and classes
- Encapsulation and access control
- Inheritance
- Subtype polymorphism
- Modularisation

Algorithms:

- Basic search and sort algorithms
- Data structures: lists, arrays and trees
- Algorithmic problem solving

Correctness and termination:

- Testing
- Specification: invariants, pre- and post conditions
- Induction

From [INF-02-02-K-2] Modelling of Software Systems:

Students learn basic modeling techniques about the software life cycle. The focus is on engineering techniques, such as UML modeling for object-oriented procedures and functionally decomposing models in analysis and design.

- UML modeling in analysis and design (class and object diagrams, communication and sequence diagrams and others)
- Functional decomposing models (Structured Analysis, Real Time Analysis, Structured Design)
- Modeling of non-functional properties
- Virtualization on the basis of models
- Traditional process models of software development (waterfall, V-model, prototypes, evolutionary, incremental and concurrent models)
- Project management models (network plan, Gantt chart, effort calculations)
- Models in quality assurance (especially model-based testing)

Competencies / intended learning achievements

Mit erfolgreichem Abschluss des Moduls werden die Studierenden in der Lage sein,

- die Grundbegriffe der Programmierung und Modellierung zu benennen,
- die Merkmale verschiedener Programmierparadigmen zu erläutern,
- kleine bis mittelgroße Programme in einer Programmiersprache idiomatisch zu modellieren, zu implementieren und zu testen,
- fortgeschrittene funktionale, imperative und objektorientierte Programmierkonzepte und -techniken einzusetzen
- elementare Algorithmen und Datentypen zu implementieren und bei der Problemlösung zu verwenden,
- Sachverhalte in geeigneten Modellen abzubilden und zu visualisieren,
- bestehende Modelle zu prüfen und die Konsistenz mit zugrundeliegenden Sachverhalten zu bewerten,
- Inhalte zwischen verschiedenen Modellen zu überführen (z.B. aus UML-Modellen der Analyse in UML-Modelle des Entwurfs).

Literature

From [INF-02-01-K-2] Foundations of Programming:

- FANCHER, Dave. The Book of F#: Breaking Free with Managed Functional Programming. No Starch Press, 2014.
- HANSEN, Michael R.; RISCHEL, Hans. Functional programming using F. Cambridge University Press, 2013.
- PICKERING, Robert; DE LA MAZA, Michael. Beginning F♯. Apress, 2009.

From [INF-02-02-K-2] Modelling of Software Systems:

- T. Ottmann, P. Widmayer: Algorithmen und Datenstrukturen.
- Mehlhorn K., Datenstrukturen und effiziente Algorithmen. Band 1 Sortieren und Suchen. Teubner, 1988.
- G. Goos: Vorlesung über Informatik. Band 1 und 2.
- M. Broy: Informatik. Eine grundlegende Einführung.
- Poetzsch-Heffter: Konzepte objektorientierter Programmierung.
- G. Krüger: Handbuch der Java-Programmierung.
- Liskov: Program Development in Java.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Pattern: Elements of Reusable Object-Oriented Software.
- W. Zuser, S. Biffl, T. Grechenig, M. Köhle: Software Engineering mit UML und dem Unified Process.
- Züllighoven H., Object-Oriented Construction Handbook, dpunkt-Verlag 2005.
- Booch G., Rumbaugh J., Jacobson I., The Unified Modeling Language User Guide, Addison-Wesley 1998.
- DeMarco T., Structured Analysis and System Specification, Englewood Cliffs: Prentice Hall, 1985.
- Liggesmeyer P., Software-Qualität, Spektrum-Verlag Heidelberg, 2002.

Requirements for attendance of the module (informal)

None

- Notice: Some Courses have informal requirements for attendance:
  - #A: [INF-02-02-K-2] Modelling of Software Systems (2V+1U, 4.0 LP) (P: Obligatory)

Requirements for attendance of the module (formal)

None

References to Module / Module Number [INF-02-40-M-2]

Course of Study	Section	Choice/Obligation
[INF-82.B16-SG] B.Sc. Socioinformatics	[Compulsory Modules] Computer Science	[P] Compulsory