# Clean Code Development

Winter

(in German: Clean Code Development )

## Module-ID: FIN-INF-120432

| Abbreviation | Credit Points | Semester | Term | Duration | Language | Level |
|---|---|---|---|---|---|---|
| CCD | 6 | Winter | starting at 1. | 1 Semester | english | Master |

| | |
|---|---|
| **Link:** | LSF |
| **Limited accessability:** | true |
| **Responsibility:** | Prof. Dr. Frank Ortmeier |
| **Lecturer:** | Prof. Dr. Frank Ortmeier |
| **Classes:** | • Vorlesung *Clean Code Development*<br>• Übung *Clean Code Development* |
| **Applicability in curriculum:** | - M.Sc. INF: Informatik<br>- M.Sc. INF: Schlüssel- und Methodenkompetenzen<br>- M.Sc. INGINF: Informatik<br>- M.Sc. INGINF: Schlüssel- und Methodenkompetenzen<br>- M.Sc. DKE: Applied Data Science<br>- M.Sc. DE: Methoden des Digital Engineering<br>- M.Sc. DE: Methoden der Informatik<br>- M.Sc. DE: Fachliche Spezialisierung<br>- M.Sc. VC: Computer Science<br>- M.Sc. VC: Schlüssel- und Methodenkompetenzen |

**Intended learning outcomes:**
Students:

- know and understand the principles of Clean Code Development
- are able to apply the associated tools and practices in practice

- learn to independently select and adapt the concepts suitable for a project
- learn to distribute tasks within a team and collaborate in an organized manner

**Content:**

- Software engineering deals with the creation and development of software, the organization and modeling of related data structures, and the operation of software systems. It thus encompasses software design, implementation, and management.
- The introductory lecture *Software Engineering I* provided the foundational knowledge for designing and writing good software.
- This advanced lecture introduces modern techniques and methods that are commonly used in the development of large software systems.
- We align with the four core values of *Clean Code Development* – evolvability, correctness, production efficiency, and comprehensibility.
- To achieve these goals, a variety of programming principles and practices will be introduced.
- These include, among others, team formation and organization in software development, principles and tools of Clean Code Development, continuous integration and automated build systems, bug tracking, error localization and debugging, automated and model-based testing, code analysis and quality metrics, requirements engineering and tracing, as well as distributed and component-based software architectures.
- The course will offer a high degree of practical engagement with tools and software development techniques, especially in the exercises. The goal is not only to convey abstract knowledge but also to develop hands-on competencies.
- Participation in the exercises is mandatory and will be conducted in groups.

**Workload:**
56h Präsenszeit + 124h selbständige Arbeit am Projekt

| Pre-examination requirements: | Type of examination: | Teaching method / lecture hours per week (SWS): |
|---|---|---|
| Projekt | Mündliche Prüfung | • Vorlesung (2 SWS)<br>• Übung (2 SWS) |

| Prerequisites according to examination regulations: | Recommended prerequisites: |
|---|---|
| keine | Software Engineering |

| Media: | Literature: |
|---|---|
| | |

**Comments:**
Enrollment Restriction: Limited number of participants. The selection process will be announced during the first lecture.

---

Revision #1
Created 2026-10-01 07:34:15 UTC by Fakultät für Informatik
Updated 2026-10-01 07:34:15 UTC by Fakultät für Informatik