Hochschule Konstanz
University of Applied Sciences

Course number: SET 12150
Software Engineering +
Object-Oriented Programming
Study level: Bachelor / Undergraduate

Prof. Dr. Burkhard Lehner

Language of instruction: English
ECTS Credits: 5

Subject-specific competencies:

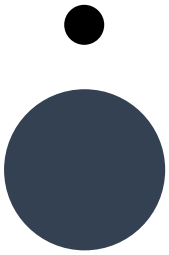
- Students are familiar with the basic software technologies.
- They are able to evaluate the use of software technologies in order to become self-sufficient in this rapidly evolving area.
- They are able to analyze and evaluate software engineering issues and problems.
- They are able to develop high quality software components for electrical engineering applications.
- They know and understand the three main principles of object-oriented programming for developing software.
- They can install software development tools on a computer.
- They know and can use an integrated development environment (IDE) to create object-oriented programs.
- They understand the concepts of event-driven graphical user interfaces (GUI), and can use tools for quickly designing such graphical user interfaces.

Methodological competencies:

- Students know the tasks, methods and tools of professional software development.
- They can act in the various roles of modern software development processes.
- They can transform a written problem description into a first draft of an object-oriented software design.
- They can translate a software design specified as a UML class diagram into an object-oriented program.
- They can use software development tools to analyze and optimize object-oriented programs and to find and remove bugs.
- They are able to write object-oriented programs with a well-structured error handling concept.

Personal competencies:

- Students can act in the various roles of modern software development processes.
- They can independently obtain information on specific issues and use it in a targeted manner.
- They can work and communicate in groups.
- They can judge their own software development skills.



Teaching Content:

Software Engineering

- Software development processes and quality management
- Requirement engineering (incl. Use-Case Diagram, Activity Diagram)
- Software architecture and design (incl. Class Diagram, Sequence Diagram, State Machine Diagram)
- Design patterns
- Software tests
- Working in teams (incl. issue tracker, version control, GitLab, Github Flow)
- Databases and data description languages

Object-Oriented Programming

- Objects and Classes
- Coop erating Objects
- Encapsulation
- Inheritance
- Polymorphism
- Abstract Classes
- Errors and Exceptions