

2019.HS

<b>Module Name: Software Engineering</b>	
Module Code	w.BA.XX.2SWEng.XX
Module Description	This module teaches the foundations of systematic and object-oriented programming. Students are taught to think in algorithmic and object-oriented terms. They not only acquire knowledge and a new way of thinking but also the ability to apply what they have learned in specific situations. The main goal of the module is the acquisition of know-how in procedural and object-orienting programming and in modeling software. These are specific IT competencies used in particular by software architects.
Program and Specialization	Business Information Technology
Legal Framework	Academic Regulations BSc dated 29.01.2009, Appendix to the Academic Regulations for the degree programs in Business Administration, Business Information Technology, and Business Law, first adopted on 12.05.2009
Module Category	<b>Module Type:</b> Compulsory
	<b>Program Phase:</b> First-Year Studies
ECTS	6
Organizational Unit	W Institut für Wirtschaftsinformatik Ltg
Module Coordinator	David Grünert (grud)
Deputy Module Coordinator	Alexandre de Spindler (desa)
Prerequisite Knowledge	none
Contribution to Program Learning Goals (Affected by Module)	<ul style="list-style-type: none"> <li>§ Professional Competence</li> <li>§ Methodological Competence</li> <li>§ Social Competence</li> <li>§ Self-Competence</li> </ul>
Contribution to Program Learning Objectives	Professional Competence <ul style="list-style-type: none"> <li>§ Knowing and Understanding Content of Theoretical and Practical Relevance</li> <li>§ Apply, Analyze, and Synthesize Content of Theoretical and Practical Relevance</li> <li>§ Evaluate Content of Theoretical and Practical Relevance</li> </ul> Methodological Competence <ul style="list-style-type: none"> <li>§ Problem-Solving &amp; Critical Thinking</li> <li>§ Scientific Methodology</li> <li>§ Work Methods, Techniques, and Procedures</li> <li>§ Information Literacy</li> <li>§ Creativity &amp; Innovation</li> </ul> Social Competence <ul style="list-style-type: none"> <li>§ Written Communication</li> <li>§ Oral Communication</li> <li>§ Intercultural Insight &amp; Ability to Change Perspective</li> </ul> Self-Competence <ul style="list-style-type: none"> <li>§ Self-Management &amp; Self-Reflection</li> <li>§ Learning &amp; Change</li> </ul>
Module Learning Objectives	Students... <ul style="list-style-type: none"> <li>§ understand the principles of object-oriented programming.</li> <li>§ understand the principles of object-oriented modeling.</li> <li>§ design algorithms and data models in accordance with requirements.</li> <li>§ translate requirements into an object-oriented model.</li> <li>§ develop their own programs in Java independently.</li> <li>§ understand the basic principles of procedural programming.</li> <li>§ evaluate the formal correctness of UML diagrams.</li> <li>§ use UML diagrams to formalize software requirements.</li> <li>§ read and interpret UML diagrams.</li> <li>§ are able to develop their own JAVA programming skills independently.</li> <li>§ are able to acquire new object-oriented programming languages with greater ease.</li> </ul>

Module Content	§ Modeling using UML: UML activity diagrams, UML class diagrams § Data structures: variables, lists, arrays, maps, files, JSON § Procedural programming: variables, assignment, operations, conditions, loops, methods, return values, arguments, recursion, exceptions, lambda expressions § Object-oriented programming: classes, class attributes, class methods, objects/instances, pointers, associations, inheritance, static elements § Programming of reactive software: command lines, Java as backend for web applications § Design and procedure: design patterns, test-driven development § Tool know-how: programming environment (Eclipse), version control (git), build management (Gradle) § Basic knowledge of the libraries used		
Links to other modules	The content of this module is linked to the following modules: w.BA.XX.2InfoM-WIN.XX w.BA.XX.2ITPM-WIN.XX w.BA.XX.2REng.XX w.BA.XX.2WEng-WIN.XX		
Methods of Instruction	§ Lecture § Interactive Instruction § Application Tasks § Exercises § Problem-Oriented Teaching § Project Work	<b>Social Settings Used:</b> § Individual Work § Pair Work	
Digital Resources	§ Teaching Videos § Teaching Materials § Practice and Application Exercises (with Key) § Multiple Choice Tests		
Type of Instruction	<b>Classroom Instruction</b>	<b>Guided Self-Study</b>	<b>Autonomous Self-Study</b>
Large Class	28 h	-	
Small Class	28 h	28 h	
Group Instruction	-	-	
Practical Work	-	-	
Seminar	-	-	
<b>Total</b>	<b>56 h</b>	<b>28 h</b>	<b>96 h</b>
Performance Assessment			
<b>End-of-module exam</b>	<b>Form</b>	<b>Length (min.)</b>	<b>Weighting</b>
Written exam	Specified documentation	90	100,00 %
<b>Permitted Resources</b>	No calculator	With dictionary	
<b>Others</b>	<b>Assessment</b>	<b>Length (min.)</b>	<b>Weighting</b>
All but two freely selectable exercises must be completed and submitted within the deadline. Alternatively, students may complete a project assignment.	Pass/Fail	-	-
Classroom Attendance Requirement	-		
Language of Instruction/Examination	German		
Compulsory Reading	§ Lecture notes on Moodle (slides) and literature in semester plan provided		
Recommended Reading	§ Mössenböck, H. (2014). Sprechen Sie Java. 5th edition. Heidelberg: dpunkt Verlag. ISBN 978-3-86490-099-0. § Kecher, C., Salvanos, A. & Hoffmann-Elbern, R. (2017). UML 2.5: Das umfassende Handbuch. 6th edition. Rheinwerk Verlag. ISBN 978-3-8362-6018-3.		
Comments	-		