Module: Secure Programming

Level	Master	**Short Name**	SPRG
Responsible Lecturers	Oliver Stecklina, Prof. Dr.		
Department, Facility	Electrical Engineering and Computer Science		
Course of Studies	Applied Information Technology, Master		
Compulsory/elective	Elective	**ECTS Credit Points**	5
Semester of Studies	(Unspecified)	**Semester Hours per Week**	4
Length (semesters)	1	**Workload (hours)**	150
Frequency	WiSe	**Presence Hours**	60
Teaching Language	German/English	**Self-Study Hours**	90
The following section is filled only if there is **exactly one** module-concluding exam.			
Exam Type	Oral Exam	**Exam Language**	German/English
Exam Length (minutes)	30	**Exam Grading System**	
Learning Outcomes	After successfully completing the course, students will be able to: • Explain and apply methods and technologies for writing safe programs • Develop programs using the Ada programming language • C programs comply with the MISTRA-C standard Analyze programs with regard to safety-critical program requirements		
Participation Prerequisites			
The previous section is filled only if there is **exactly one** module-concluding exam.			
Consideration of Gender and Diversity Issues	✔ Use of gender-neutral language (THL standard) ✘ Target group specific adjustment of didactic methods ✔ Making subject diversity visible (female researchers, cultures etc.)		
Applicability			
Remarks			

Module Course: Secure programming (Lecture)

(of Module: Secure Programming)

Course Type	Lecture	**Form of Learning**	Presence
Mandatory Attendance	no	**ECTS Credit Points**	3
Participation Limit		**Semester Hours per Week**	3
Group Size		**Workload (hours)**	90
Teaching Language	German/English	**Presence Hours**	45
Study Achievements ("Studienleistung", SL)		**Self-Study Hours**	45
SL Length (minutes)		**SL Grading System**	
The following section is filled only if there is a course-specific exam.			
Exam Type		**Exam Language**	
Exam Length (minutes)	120	**Exam Grading System**	
Learning Outcomes			
Participation Prerequisites			
The previous section is filled only if there is a course-specific exam.			
Contents	Autonomous and safety-critical systems increasingly determine our everyday life. They influence what we do or how we can do it. Errors in these systems often have a direct impact on us and our daily life. Errors can be made in the design but also in the implementation. However, hardware design must also be carried out with great care, especially when programming applications for safety-critical systems. In order to support the creation of the programs, rules for programming can be set up or secure programming languages can be used. The module "Secure Programming" gives the students an insight into the safe programming of applications for safety-critical (autonomous) systems. On the one hand, rules for unsafe languages, such as C, and on the other hand secure programming languages, e.g. ADA, considered. Practical examples investigate how rules and programming languages help to avoid errors when implementing a safety-critical application. Part 1 - Introduction to Safe Programming Part 1 - MISRA C Part 2 - Programming in ADA and SPARK		
Literature	[1] OWASP Secure Coding Pratices Quick Reference Guide, 2010 [2] John Barnes; Programming in Ada 2012, Cambridge University Press, 2014		

	[3] John W. McCormick, Building High Integrity Applications with SPARK, Cambridge University Press, 2015 [4] Brian W. Kernighan; The C Programming Language; Prentice Hall 2000 [5] MISRA C: 2012 Guidlines for the use of the C language in critical systems, 2012
Remarks	

Module Course: Secure Programming (Practical Training)

(of Module: Secure Programming)

Course Type		**Form of Learning**	Presence
Mandatory Attendance	no	**ECTS Credit Points**	2
Participation Limit		**Semester Hours per Week**	1
Group Size	12	**Workload (hours)**	60
Teaching Language	German/English	**Presence Hours**	15
Study Achievements ("Studienleistung", SL)	Practical Training	**Self-Study Hours**	45
SL Length (minutes)		**SL Grading System**	Pass

The following section is filled only if there is a course-specific exam.

Exam Type		**Exam Language**	
Exam Length (minutes)		**Exam Grading System**	
Learning Outcomes			
Participation Prerequisites			

The previous section is filled only if there is a course-specific exam.

Contents	The knowledge of the lecture is to be consolidated in practical examples: • Software analysis of sample programs, • Programming according to MISRA-C, and Programming in Ada.
Literature	
Remarks	