

<b>Department</b>	07 Computer Science and Mathematics
<b>Course title</b>	<b>Program Verification</b>
<b>Hours per week (SWS)</b>	4
<b>Number of ECTS credits</b>	5
<b>Course objective</b>	Understanding different approaches to program verification. Practical experience using different verification tools.
<b>Prerequisites</b>	Logic
<b>Recommended reading</b>	
<b>Teaching methods</b>	Lecture (2 SWS) + practical exercise (2 SWS)
<b>Assessment methods</b>	oral or written exam
<b>Language of instruction</b>	English
<b>Name of lecturer</b>	Prof. Matthias Guedemann
<b>Email</b>	<a href="mailto:matthias.guedemann@hm.edu">matthias.guedemann@hm.edu</a>
<b>Link</b>	
<b>Course content</b>	<p>For systems whose failure endangers human life, health or very high assets (safety-critical systems), reliance on successful tests is not sufficient. Verification is the mathematical proof that a system possesses certain properties essential to its safety --- provided that program properties are formally specified.</p> <p>On a common theoretical basis, both classical program verification based on the Hoare calculus and abstract interpretation and model checking are considered</p> <p>Fundamentals of logic, lattices and fixed points, Axiomatic verification according to Hoare, Abstract interpretation according to Cousot, Software model checking using bit-precise SMT solvers, Application of modern verification tools used in industry, Accompanied by practical exercises</p>
<b>Remarks</b>	