

<b>Module title</b>		<b>Abbreviation</b>
Network and Concurrent Programming		10-GE-NPP-162-m01
<b>Module coordinator</b>		<b>Module offered by</b>
holder of the Chair of Computer Science IX		Institute of Computer Science
<b>ECTS</b>	<b>Method of grading</b>	<b>Only after succ. compl. of module(s)</b>
5	numerical grade	--
<b>Duration</b>	<b>Module level</b>	<b>Other prerequisites</b>
1 semester	undergraduate	--
<b>Contents</b>		
<p>This module will give the students the opportunity to learn and practice the skills essential to the development of networked and multithreaded applications. This module will give an overview of networking protocols and related APIs (application programmer interfaces), and familiarize the students with concurrent and distributed programming paradigms, focusing in particular on the realtime interactive systems (RIS) domain (such as video games, virtual reality or mixed reality applications). Issues faced when developing a concurrent or distributed application will be tackled, including synchronization and security issues. Examples of abstractions will be studied, including concurrency design patterns, distributed objects models and architectures. Classical and innovative architectures and deployment will be studied. Students will be given the opportunity to experiment and practice with the issues studied through the use of suitable libraries and middleware (e.g. game engine) during the exercise sessions.</p>		
<b>Intended learning outcomes</b>		
<p>The students possess an solid understanding of computer network systems, classical networking protocols and communication models on private networks and Internet, and of the issues faced when developing distributed applications with strong realtime interactive requirements such as digital games, virtual reality or mixed reality applications.</p> <p>The students are able to to design and develop concurrent and networked applications through the use of adequate design patterns and communication models and have an overview of different concurrent programming models, such as threads and processes, and the different communication models they can support</p>		
<b>Courses</b> (type, number of weekly contact hours, language – if other than German)		
V (2) + Ü (2) Module taught in: German or English		
<b>Method of assessment</b> (type, scope, language – if other than German, examination offered – if not every semester, information on whether module is creditable for bonus)		
a) written examination (approx. 60 to 120 minutes) or b) presentation of project results (approx. 20 minutes) Language of assessment: German and/or English creditable for bonus		
<b>Allocation of places</b>		
--		
<b>Additional information</b>		
--		
<b>Workload</b>		
150 h		
<b>Teaching cycle</b>		
--		
<b>Referred to in LPO I</b> (examination regulations for teaching-degree programmes)		
--		
<b>Module appears in</b>		
Bachelor' degree (1 major) Games Engineering (2016)		



Bachelor' degree (1 major) Games Engineering (2017)