


5 credits

30.0 h + 15.0 h

Q2

Teacher(s)	Riviere Etienne ;
Language :	English
Place of the course	Louvain-la-Neuve
Main themes	<p>This course treats a specific advanced topic or selection of topics of current research interest in the area of software engineering.</p> <p>The actual topic(s) may vary from year to year, and will be chosen from a variety of software engineering domains such as data-intensive computing, software analytics, development and analysis of large evolving software systems, big data techniques, software repository mining, software recommendation systems, software visualization, novel programming technologies, software requirements and analysis, model-driven software engineering, software configuration management, software engineering processes, software engineering tools and methods, software testing and quality aspects, etc.</p>
Aims	<p>Given the learning outcomes of the "Master in Computer Science and Engineering" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes:</p> <ul style="list-style-type: none"> • INFO1.1 • INFO3.1 • INFO6.3 <p>1 Given the learning outcomes of the "Master [120] in Computer Science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes:</p> <ul style="list-style-type: none"> • SINF1.M3 • SINF3.1 • SINF6.3 <p>The students shall acquire advanced theoretical knowledge and technical competences about the topics covered in the course.</p> <p>-----</p> <p><i>The contribution of this Teaching Unit to the development and command of the skills and learning outcomes of the programme(s) can be accessed at the end of this sheet, in the section entitled "Programmes/courses offering this Teaching Unit".</i></p>
Evaluation methods	<p>Evaluation methods:</p> <ul style="list-style-type: none"> • Projects (40% of the final mark) • Exam (60% of the final mark) <p>The project work is mandatory and cannot be repeated for the second examination session. Project reports will be evaluated only once before the first session and cannot be presented for the second session.</p>
Teaching methods	<ul style="list-style-type: none"> • Short lectures • Guided and interactive exercises in class • Readings and/or videos at home • Practical sessions
Content	<p>This course focuses on parallel programming with shared memory objects, when the accesses by multiple thread must be synchronized for coherence.</p> <p>It targets modern multicore CPUs.</p> <p>It introduces the use of dedicated mechanisms and instructions for synchronization in these CPUs to build scalable shared data structures and algorithms.</p> <p>If presents performance considerations, including those arising from non-uniform memory access and multi-level memory hierarchies.</p>
Bibliography	<p>The Art of Multiprocessor Programming, Maurice Herlihy and Nir Shavit, Morgan Kaufmann. ISBN 978-0-12-370591-4.</p> <p>UCL/INGI library reference 135-08-02</p>

Other infos	All relevant course material and slides as well as practical information related to the course will be accessible on Moodle, which will also be the primary means of communication between the teacher(s) and the students. Background: <ul style="list-style-type: none">• Introduction to Algorithmics (LSINF1103)• Algorithmics and data structures (LSINF1121)• Performance measurements (LFSAB1402)• Language constructs for concurrent programming (LINGI1131)
Faculty or entity in charge	INFO

Programmes containing this learning unit (UE)				
Program title	Acronym	Credits	Prerequisite	Aims
Master [120] in Computer Science and Engineering	INFO2M	5		
Master [120] in Computer Science	SINF2M	5		