




5.00 credits

30.0 h + 15.0 h

Q1

Teacher(s)	Pecheur Charles ;
Language :	English > French-friendly
Place of the course	Louvain-la-Neuve
Main themes	This course studies the principles, formalisms and tools used to model and analyse concurrent computer systems. <ul style="list-style-type: none"> • Models of Concurrent Systems • Semantics of Concurrent Systems • Properties of Concurrent Systems • Verification of Concurrent Systems
Learning outcomes	<p>At the end of this learning unit, the student is able to :</p> <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"> • INFO1.1-3 • INFO2.2-3, INFO2.5 • INFO5.2, INFO5.5 • INFO6.1, INFO6.4 <p>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.M1, SINF1.M2 • SINF2.2-3, SINF2.5 • SINF5.2, SINF5.5 • SINF6.1, SINF6.4 <p>Students completing successfully this course will be able to</p> <ul style="list-style-type: none"> • master the mathematic concepts and tools which enable modelling and analysis of concurrent computer systems; • model an actual concurrent system in a appropriate abstract formalism; • use automated verification techniques to analyse concurrent systems. <p>Students will have developed skills and operational methodology. In particular, they have developed their ability to</p> <ul style="list-style-type: none"> • model existing software ; • thinking with abstractions and use a formalized approach to infer properties of an existing system .
Evaluation methods	<ul style="list-style-type: none"> • 3 assignments, 30% of the final grade. • Exercises: test at the end of term, 30% of the final grade. • Theory: oral exam, 40% of the final grade. A list of questions is provided at the end of the course. <p>The assignments can only be presented during the quadrimester of the course. They cannot be represented in subsequent exam sessions; the grade remains acquired for subsequent sessions.</p> <p>According to circumstances, the exam may be organized remotely.</p>
Teaching methods	<ul style="list-style-type: none"> • Lectures • Exercises (theoretical exercises to master the concepts, followed by computer room session to apply these on concurrent systems) • Assignments (performed conjointly by two students) <p>The exercise sessions are closely related to the assignments and help student using similar models and tools on other concurrent systems.</p> <p>Due to circumstances, all or part of the lectures and exercises may be streamed and recorded for distance learning.</p>
Content	<ul style="list-style-type: none"> • Models of Concurrent Systems: processes and actions, conditions and choices, concurrency, synchronization, process algebras.

	<ul style="list-style-type: none"> • Semantics of Concurrent Systems: state machines and transition systems, finite and infinite traces, concurrency by interleaving, equivalences and minimization. • Properties of Concurrent Systems: invariants, safety and liveness properties, temporal logic, refinement relations. • Verification of Concurrent Systems: model checking, equivalence checking.
Inline resources	http://icampus.uclouvain.be/claroline/course/index.php?cid=ingi2143
Bibliography	<p>Livre de référence (obligatoire)</p> <ul style="list-style-type: none"> • J Magee and J Kramer, Concurrency: State Models and Java Programming (2nd Ed.), Wiley, 2006. <p>Autres références</p> <ul style="list-style-type: none"> • H Bowman and R Gomez, Concurrency Theory: Calculi and Automata for Modelling Untimed and Timed Concurrent Systems, Springer, 2006. • AW Roscoe, The Theory and Practice of Concurrency, Prentice Hall, 1998 (http://web.comlab.ox.ac.uk/oucl/work/bill.roscoe/publications/68b.pdf). • E Clarke, O Grumberg and D Peled, Model Checking, MIT Press, 1999. • B Bérard et al., Systems and Software Verification, Springer, 2001.
Faculty or entity in charge	INFO

Programmes containing this learning unit (UE)				
Program title	Acronym	Credits	Prerequisite	Learning outcomes
Master [120] in Computer Science and Engineering	INFO2M	5		
Master [120] in Computer Science	SINF2M	5		
Master [120] in Data Science Engineering	DATE2M	5		
Master [120] in Data Science: Information Technology	DATI2M	5		