UCL Université catholique de Louvain

LSINF2224

2013-2014

Programming methods

5.0 credits	30.0 h + 15.0 h	2q

Teacher(s) :	Pecheur Charles ;
Language :	Anglais
Place of the course	Louvain-la-Neuve
Inline resources:	> http://icampus.uclouvain.be/claroline/course/index.php?cid=LINF2224
Prerequisites :	-- basic notions in theory of programming, as covered in courses such as INGI1122 (Program conception methods), INGI1101 (Logic and discrete structures) and INGI1123 (Calculability)
Main themes :	-- Program foundations and properties, semantics, validity and proof. -- Deductive proof for programs: Hoare logic, weakest preconditions, verification conditions, invariants and variants. -- Automating proofs: loops, procedures and recursion, data structures, reactive program. -- State-based analysis: model-checking, temporal logic, abstraction.
Aims :	Students completing successfully this course will be able to -- define and formalize the principles of program analysis and verification introduced in bachelor courses. -- describe and apply the techniques that allow those principles to be automated on a computer. -- illustrate the potential and limits of such techniques using pratical examples. Students will have developed skills and operational methodology. In particular, they have developed their ability to -- formalize in mathematical form a given problem; -- write a brief technical report covering the main elements of an analysis; -- argue orally. *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".*
Evaluation methods :	3 assignments, 45% of the final grade. Theory: oral exam, 55% of the final grade. A list of questions is provided at the end of the course. The asssignments can only be presented during the quadrimester of the course. They cannot be represented in subsequent exam sessions.
Teaching methods :	The course combines -- lectures, -- exercises (modeling and analysis of programs), -- and assignments where students use automated verification software (ESC/Java, Java PathFinder) to prove properties of Java programs. The aims, methodology and tools used in each assignment are described in a document provided to the students. The program code to be analyzed is commented and a brief user's guide is included. The program verification and model-checking tools to be used are clearly indicated. To ease their grip by students, exercise sessions are planned. In addition, a consultancy is provided by the supervisors of the course in case of trouble. At the end of each mission, students write a brief report used as a basis for evaluation.
Content :	-- Demonstration -- Introduction -- Foundations

	-- Sequential Programs -- Verification Conditions -- Procedures -- Recursion -- Data Structures -- Reactive Programs -- State-Based Models -- Model Checking -- Abstraction
Bibliography :	Support material : -- copies from lecture slides Bibliography : -- B. Liskov, J. Guttag. Program Development in Java: Abstraction, Specification and Object-Oriented Design. Addison-Wesley, 2001. -- O.-J. Dahl. Verifiable Programming. Prentice Hall, 1992. -- K. R. Apt, E.-R. Olderog. Verification of Sequential and Concurrent Programs. Springer Verlag, 1991. -- J. Loeckx, K. Sieber. The Foundations of Program Verification (2nd Ed.) Wiley-Teubner, 1984. -- D. Gries, The Science of Computer Programming. Springer-Verlag, 1981.
Other infos :	
Cycle and year of study :	> Master [120] in Computer Science and Engineering > Master [120] in Computer Science > Master [120] in Chemistry and Bio-industries > Master [120] in Environmental Bioengineering > Master [120] in Forests and Natural Areas Engineering > Master [120] in Agricultural Bioengineering
Faculty or entity in charge:	INFO