


This learning unit is not open to incoming exchange students!

Teacher(s)	Deville Yves ;
Language :	French
Place of the course	Charleroi
Prerequisites	<p>This course assumes that the student acquired programming skills, algorithmic and programming language targeted in course LEPL1402 and discrete mathematics as seen in courses LINFO1114 or LEPL1108</p> <p><i>The prerequisite(s) for this Teaching Unit (Unité d'enseignement – UE) for the programmes/courses that offer this Teaching Unit are specified at the end of this sheet.</i></p>
Main themes	<ul style="list-style-type: none"> • Theory of computability: problems and algorithms, computable and non-computable functions, reduction, undecidable problem classes (Rice's theorem), fixed point theorem, Church-Turing thesis • Logic: logic of propositions and logic of predicates (syntax, semantics, proof, quantifiers, model checking, resolution) • Computability Models: Turing Machine • Theory of complexity: complexity classes, NP-completeness, Cook's theorem, NP-complete problem solving.
Learning outcomes	<p>At the end of this learning unit, the student is able to :</p> <p>Given the learning outcomes of the "Bachelor in Engineering" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes:</p> <ul style="list-style-type: none"> • AA1.1, AA1.2 • AA2.4 <p>Given the learning outcomes of the "Bachelor 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"> • S1.I3, S1.G1 • S2.2 <p>Students who have successfully completed this course will be able to</p> <ul style="list-style-type: none"> • recognize, explain and identify the limitations of information processing by a computer; • explain and make good use of the main computability models by explaining their bases, differences and similarities; • convert current language assertions into logical expressions using the syntax and semantics of the logic of propositions or predicates • recognize, identify and apprehend non-calculable problems as well as intrinsically complex problems. <p>Students will have developed methodological and operational skills. In particular, they will have developed their capacity to</p> <ul style="list-style-type: none"> • take a critical look at the performance and capacity of computer systems
Evaluation methods	<p>Different modes of evaluation can be organized: continuous assessment, graded work, participation, exam. The exam will be written, but in case of doubt on the part of the teacher as to the grade to be given to a student, the student may be questioned orally. Depending on the number of students, the September exam can be an oral exam.</p>
Teaching methods	<p>This course can be given in a variety of face-to-face and distance modalities. These may include lectures, readings, preparations, exercises, as well as individual or group work.</p>
Content	<ul style="list-style-type: none"> • Introduction • Enumerable sets • Computability: fundamental results • Models of computability • Propositional logic • Introduction to algorithmic complexity • Complexity classes

Faculty or entity in charge	SINC
-----------------------------	------

Programmes containing this learning unit (UE)				
Program title	Acronym	Credits	Prerequisite	Learning outcomes
Bachelor in Computer Science	SINC1BA	5		