

Teacher(s)	Mens Kim ;Nijssen Siegfried ;
Language :	French
Place of the course	Charleroi
Learning outcomes	
Evaluation methods	<p>A programming assignment is due each week.</p> <p>A mid-term evaluation will be organised halfway throughout the quadrimester.</p> <p>The end-of-term exam aims to assess both the understanding of the course material and the capacity to apply it to write simple but correct Python programs.</p> <p>The final course mark takes into account the mid-term evaluation and assignments during the quadrimester, in addition to the mark of the end-term exam.</p> <p>The assignments and mid-term evaluation cannot be retaken for the June or September sessions.</p> <p>If the mark for the mid-term evaluation is higher than that for the end-term exam, it will count for 1/3 and the mark of the end-term exam for 2/3.</p> <p>If the mark for the mid-term evaluation is lower than that for the end-term exam, only the mark for the exam will be used to calculate the final course mark.</p> <p>A bonus of 1 point will be granted to students who have participated in and regularly submitted their programming assignments during the quadrimester.</p> <p>In case of plagiarism detection confirmed by a plagiarism detection tool the course teachers reserve the right to invite the student to pass an oral interrogation.</p>
Teaching methods	<p>The chosen teaching method relies on active student participation, through a mixture of :</p> <ul style="list-style-type: none"> <li>• course lectures,</li> <li>• practical exercise sessions with tutors,</li> <li>• programming exercises on the INGIInious platform.</li> </ul> <p>Even though preference will be given to face-to-face teaching sessions, depending on the health situation and the number of students enrolled, other forms of teaching and evaluation (online, co-modal or hybrid) may be considered.</p>
Content	<ul style="list-style-type: none"> <li>• Programs, source code and program execution</li> <li>• Identifiers, variables, values, types, assignment</li> <li>• Expressions, statements</li> <li>• Conditional structures and loops</li> <li>• Functions, parameters, calls, results, execution, variable scoping</li> <li>• Specifications and tests</li> <li>• Modules</li> <li>• Data structures, lists, strings and their operations</li> <li>• References and nested data structures</li> <li>• Nested lists, tuples, matrices, dictionaries</li> <li>• Dichotomic search algorithms</li> <li>• File handling, input/output</li> <li>• Exception handling</li> <li>• Object-oriented programming and garbage collection</li> <li>• Classes, objects, constructors, methods</li> <li>• References to an object, self-references and self-calls</li> <li>• Class, instance and local variables, scope and visibility</li> <li>• Class composition, inheritance and encapsulation</li> <li>• Polymorphism, super calls and dynamic binding</li> <li>• Object equality</li> <li>• Linked data structures</li> </ul>
Inline resources	All course material will be made available online: slides, syllabus, exercises, ...
Faculty or entity in charge	EPL

<b>Programmes containing this learning unit (UE)</b>				
Program title	Acronym	Credits	Prerequisite	Learning outcomes
Bachelor in Computer Science	<a href="#">SINC1BA</a>	5		