



This learning unit is not open to incoming exchange students!

Teacher(s)	Mens Kim ;Nijssen Siegfried ;
Language :	French
Place of the course	Charleroi
Main themes	<ul style="list-style-type: none"> • Fundamental concepts of object-oriented programming; • Python programming language; • Analysis of a computer problem, design, specification and implementation of a solution; • Linear data structures.
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 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.I2 • S2.2, S2.4 <p>Students who have successfully completed this course will be able to :</p> <ul style="list-style-type: none"> • Apply the concepts, laws, reasonings to a disciplinary problem of squared complexity. • Describe adequate modeling and calculation tools to solve a disciplined disciplinary problem. • Model a problem and design one or more technical solutions that meet the specifications • Implement and test a solution in the form of a model, a prototype and / or a digital model. • Commit collectively to a work plan, a timetable (and roles to keep). • Communicate in graphical and schematic form interpret a diagram, present the results of a work, structure information. • Read, analyze and exploit technical documents (standards, plans, specifications, specifications, ...). • Write written summary documents taking into account the requirements of the missions (projects and problems). • Demonstrate a good understanding of the concepts and methodology of object-oriented programming. • Make good use of the elements of an object-oriented language such as Python.
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"> • course lectures, • practical exercise sessions with tutors, • programming exercises on the INGIInious platform. <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"> • Programs, source code and program execution • Identifiers, variables, values, types, assignment • Expressions, statements • Conditional structures and loops • Functions, parameters, calls, results, execution, variable scoping • Specifications and tests • Modules • Data structures, lists, strings and their operations • References and nested data structures • Nested lists, tuples, matrices, dictionnaires • Dichotomic search algorithms • File handling, input/output • Exception handling • Object-oriented programming and garbage collection • Classes, objects, constructors, methods • References to an object, self-references and self-calls • Class, instance and local variables, scope and visibility • Class composition, inheritance and encapsulation • Polymorphism, super calls and dynamic binding • Object equality • Linked data structures
Inline resources	All course material will be made available online: slides, syllabus, exercices, ...
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		