UCLouvain

lepl1401

2024

Informatics 1

5.00 credits	30.0 h + 30.0 h	Q1

Teacher(s)	. SOMEBODY ;Mens Kim ;Nijssen Siegfried ;Pecheur Charles ;
Language :	French
Place of the course	Louvain-la-Neuve
Main themes	• Basic concepts of object-oriented programming • The Java programming language • Problem analysis; specification and implementation of solutions • Linear data structures, including dynamic implementations.
Learning outcomes	**At the end of this learning unit, the student is able to :** Contribution of the course to the program objectives Regarding the learning outcomes of the program of Bachelor in Engineering, this course contributes to the development and the acquisition of the following learning outcomes: • LO 1.1, 1.2 • LO 2.4, 2.5 • LO 3.1 • AA 4.2, 4.3, 4.4 **More specifically, at the end of the course, the student will be able to:** 1 - Apply the concepts, laws, reasoning to a disciplinary problem of framed complexity. - Describe appropriate modeling and calculation tools to solve a framed 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 play). - Communicate in graphic and schematic form; interpret a diagram, present the results of work, structure information. - Read, analyze and use technical documents (standards, plans, specifications, specifications, ...). - Write summary written 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. - Use wisely the elements of an object-oriented language such as Python.
Evaluation methods	A programming assignment is due each week. A mid-term evaluation will be organised halfway throughout the quadrimester. 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. The exam is done on a computer without access to external resources. 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. The assignments and mid-term evaluation cannot be retaken for the June or September sessions. 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. 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. A bonus of 1 point will be granted to students who have participated in an autonomous manner and regularly submitted their programming assignments during the quadrimester. The use of external help or AI tools in the assignments is not desired. In case of confirmed plagiarism during the exam, the course teachers reserve the right to invite the student to pass an oral interrogation.

Teaching methods	The chosen teaching method relies on active student participation, through a mixture of : • course lectures, • partical exercice sessions with tutors, • programming exercices on the INGInious platform? 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.
Content	• 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 • Nestsed lists, tuples, matrices, dictionnaries • 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	BTCI

Programmes containing this learning unit (UE)				
Program title	Acronym	Credits	Prerequisite	Learning outcomes
Bachelor in Engineering	FSA1BA	5		🔍