UCLouvain

# lepl1401

*2022*

# 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<br>• The Java programming language<br>• Problem analysis; specification and implementation of solutions<br>• Linear data structures, including dynamic implementations. |
| Learning outcomes | **At the end of this learning unit, the student is able to :**<br>Contribution of the course to the program objectives<br>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:<br>• LO 1.1, 1.2<br>• LO 2.4, 2.5<br>• LO 3.1<br>• AA 4.2, 4.3, 4.4<br>**More specifically, at the end of the course, the student will be able to:**<br>1 - Apply the concepts, laws, reasoning to a disciplinary problem of framed complexity.<br>- Describe appropriate modeling and calculation tools to solve a framed disciplinary problem.<br>- Model a problem and design one or more technical solutions that meet the specifications.<br>- Implement and test a solution in the form of a model, a prototype and/or a digital model.<br>- Commit collectively to a work plan, a timetable (and roles to play).<br>- Communicate in graphic and schematic form; interpret a diagram, present the results of work, structure information.<br>- Read, analyze and use technical documents (standards, plans, specifications, specifications, ...).<br>- Write summary written documents taking into account the requirements of the missions (projects and problems).<br>- Demonstrate a good understanding of the concepts and methodology of object-oriented programming.<br>- Use wisely the elements of an object-oriented language such as Python. |
| Evaluation methods | A programming assignment is due each week.<br>A mid-term evaluation will be organised halfway throughout the quadrimester.<br>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.<br>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.<br>The assignments and mid-term evaluation cannot be retaken for the June or September sessions.<br>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.<br>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.<br>A bonus of 1 point will be granted to students who have participated in and regularly submitted their programming assignments during the quadrimester.<br>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. |

| | |
|---|---|
| Teaching methods | The chosen teaching method relies on active student participation, through a mixture of :<br>• course lectures,<br>• partical exercice sessions with tutors,<br>• programming exercices on the INGInious platform?<br><br>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<br>• Identifiers, variables, values, types, assignment<br>• Expressions, statements<br>• Conditional structures and loops<br>• Functions, parameters, calls, results, execution, variable scoping<br>• Specifications and tests<br>• Modules<br>• Data structures, lists, strings and their operations<br>• References and nested data structures<br>• Nestsed lists, tuples, matrices, dictionnaries<br>• Dichotomic search algorithms<br>• File handling, input/output<br>• Exception handling<br>• Object-oriented programming and garbage collection<br>• Classes, objects, constructors, methods<br>• References to an object, self-references and self-calls<br>• Class, instance and local variables, scope and visibility<br>• Class composition, inheritance and encapsulation<br>• Polymorphism, super calls and dynamic binding<br>• Object equality<br>• 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 | | 🔍 |