UCLouvain

# lsinc1402

*2022*

# Computer Science 2

| 5.00 credits | 30.0 h + 30.0 h | Q1 |
|---|---|---|

**This learning unit is not open to incoming exchange students!**

| | |
|---|---|
| Teacher(s) | Jodogne Sébastien ;Sadre Ramin ; |
| Language : | French |
| Place of the course | Charleroi |
| Prerequisites | *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.*<br>*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.* |
| Main themes | - Introduction to Java: compilation, byte-code, virtual machine, primitive type, strings, tables<br>- Abstract data types;<br>- Linear and tree structures, and their applications;<br>- recursive solution formulation and recursive algorithms;<br>- reasoning technique: preconditions, postconditions, invariants<br>- Notions of computational complexity and analysis of the temporal and spatial complexity of an algorithm;<br>- Functional programming and higher order programming<br>- Object-oriented modeling (inheritance, composition, reuse, polymorphism, class invariant);<br>- Introduction to design patterns;<br>- Program testing and validation methods. |
| Learning outcomes | **At the end of this learning unit, the student is able to :**<br>AA 1.2, 1.5-AA 2.4 - AA 4.3, 4.4, 4.5 - AA 5.1, 5.3<br>With regard to the AA reference of the program "Bachelor in Engineering Sciences, orientation civil engineer", this course contributes to the development, acquisition and evaluation of learning outcomes.<br>following:<br>- AA 1.2, 1.5<br>- AA 2.4<br>- AA 4.3, 4.4, 4.5<br>- AA 5.1, 5.3<br><u>More specifically, at the end of the course, the student will be able to:</u><br>- make a justified choice between several representations of information and several algorithms to process them,<br>- design (fragments of) programs in a functional style,<br>- reasoning about (fragments of) programs: complexity of algorithms and efficiency of programs implementing them, recursive reasoning,<br>- apply object-oriented modeling principles,<br>- design and apply methods for testing a program,<br>- design a simple parallel program<br>Students will have developed methodological and operational skills. In particular, they will have developed their ability to:<br>- Analyze a medium-sized problem, propose a computer solution to solve it and implement it in the Java language. |
| Evaluation methods | Oral examination at the end of the quadrimester. The purpose of the exam in January is to test not only knowledge of the subject, but also the ability to apply the knowledge acquired to write programs.<br>An optional quiz is held at the mid of the quadrimester as a part of the continuing assessment.<br>The final grade is computed as follows: final_grade_over_20 = max(quiz_over_2 + exam_over_18, exam_over_20).<br>The points obtained for the quiz will be kept for the August session. |

| | |
|---|---|
| Teaching methods | Flipped classroom method:<br>• Theoretical lectures are available online as videos.<br>• Practical exercises are accessible online at any time on the INGInious platform.<br>• Lab sessions consist in asking support online to tutors or teaching assistants.<br>• On-site or remote restructuration sessions allow students to ask their questions and/or to witness the resolution of selected exercises.<br>The learning is the responsibility of each student. To pass the exam, it is imperative that the student programs regularly and using IntelliJ. |
| Content | This teaching unit focuses on:<br>- Introduction to Java: compilation, byte-code, virtual machine, primitive type, strings, tables<br>- Abstract data types;<br>- Linear and tree structures, and their applications;<br>- recursive solution formulation and recursive algorithms;<br>- reasoning technique: preconditions, postconditions, invariants<br>- Notions of computational complexity and analysis of the temporal and spatial complexity of an algorithm;<br>- Functional programming and higher order programming<br>- Object-oriented modeling (inheritance, composition, reuse, polymorphism, class invariant);<br>- Introduction to design patterns;<br>- Program testing and validation methods;<br>- Introduction to parallelization: notion of thread and synchronization mechanisms.<br>Students who have successfully completed this course will be able to<br>• to design Java programs<br>• analyze programs according to their performance<br>• to prove correctness of programs using invariants<br>• apply the principles of object-oriented programming such as genericity, abstraction, composition and reuse<br>• design and implement variants of the algorithms studied in high quality Java programs.<br>• design and manipulate simple linear and tree and recursive structures<br>• design tests for programs<br>• design functional programming approaches to solve small algorithmic problems<br>• use design patterns<br>• design simple parallel programs with synchronization mechanisms |
| Inline resources | https://lepl1402.readthedocs.io/ (links to the slides and other course contents)<br>https://moodleucl.uclouvain.be/course/view.php?id=12884 Moodle et/ou Teams (communication with the students)<br>https://inginious.info.ucl.ac.be/course/LEPL1402 for the exercises |
| Other infos | Background: LSINC1101 or a similar course. |
| 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 | LSINC1101 | 🔍 |