





6.0 credits	30.0 h + 30.0 h	1q
-------------	-----------------	----

Teacher(s) :	Bonaventure Olivier ; Pecheur Charles ;
Language :	Français
Place of the course	Louvain-la-Neuve
Inline resources:	<a href="http://moodleucl.uclouvain.be/course/view.php?id=7654">http://moodleucl.uclouvain.be/course/view.php?id=7654</a>
Main themes :	<p>Basics of programming</p> <p>--</p> <p>Values", variables, assignment</p> <p>--</p> <p>Primitive types, representation of numbers</p> <p>--</p> <p>Expressions and statements, conditions and loops</p> <p>--</p> <p>Methods, parameters, results and calls</p> <p>--</p> <p>specifications</p> <p>--</p> <p>Edition, compilation and execution of programs</p> <p>Object-oriented programming</p> <p>--</p> <p>Classes, objects, instances</p> <p>--</p> <p>Class variables and instance</p> <p>--</p> <p>Visibility, scope and lifetime of a variable</p> <p>--</p> <p>Interfaces and implementations</p> <p>--</p> <p>Extending a class and inheritance, polymorphism</p> <p>Functionality and data structures</p> <p>--</p> <p>Strings</p> <p>--</p> <p>Files and Input-Output</p> <p>--</p> <p>Exception mechanisms</p> <p>--</p> <p>Simple tables and multidimensional arrays</p> <p>--</p> <p>Working with Packages</p> <p>--</p> <p>Linked structures</p> <p>Programming methodology</p> <p>--</p> <p>Analysis of a computer problem</p> <p>--</p> <p>design, specification and implementation of a solution</p>
Aims :	<p>Given the learning outcomes of the "Bachelor in Engineering" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes:</p> <p>--</p> <p>S1.12</p> <p>--</p> <p>S2.2, S2.4</p> <p>Students completing this course successfully will be able to</p> <p>--</p> <p>demonstrate an in-depht understanding of the concepts and methodology of object-oriented programming ,</p> <p>--</p> <p>good use elements of a object-oriented language such as Java,</p> <p>--</p>

	<p>analyze a simple problem and propose a software solution to solve it and implement it in Java. Students will have developed skills and operational methodology . In particular, they have developed their ability to:</p> <ul style="list-style-type: none"> <li>--</li> <li>decompose a simple problem into sub-problems;</li> <li>--</li> <li>express the resolution of a problem as an algorithm ;</li> <li>--</li> <li>define and specify rigorously and precisely the features expected of a computer program ;</li> <li>--</li> <li>read and understand existing programs;</li> <li>--</li> <li>use of IT tools to support programming;</li> <li>--</li> <li>exploit the technique of computer language documentation;</li> <li>--</li> <li>make a small computer application as good quality Java program ;</li> <li>--</li> <li>identify test cases for the validation of a simple program.</li> </ul> <p><i>The contribution of this Teaching Unit to the development and command of the skills and learning outcomes of the programme(s) can be accessed at the end of this sheet, in the section entitled "Programmes/courses offering this Teaching Unit".</i></p>
<p><b>Evaluation methods :</b></p>	<p>A middle evaluation takes place in the middle of the first period. The note for this exam takes account of 1/3 of the final grade only if it is greater than the examination mark. The end of first period exam aims to test not only the knowledge, but also the ability to apply this knowledge to write simple Java programs, but correct</p>
<p><b>Teaching methods :</b></p>	<p>The methods used will encourage active student learning. The detailed arrangements for implementing an active student participation in learning are left to tenured, in accordance with the educational guidelines of the Faculty.</p>
<p><b>Content :</b></p>	<p>The first chapter of the course addresses the construction of algorithms. Primitive objects are given with primitive operations upon them. It is shown how arbitrarily complex algorithms acting on the objects can be built using sequential, conditional, and iterative composition of algorithms. The emphasis is put on specifications and on the use of assertions to derive correct code. The Java programming language is used in a controlled way as a tool to make the algorithms amenable to execution by a computer. The second chapter is concerned with the representation of data. Decimal, binary, octal and hexadecimal notations are reviewed as well as conversion algorithms from one notation to another. The two's complement representation for negative numbers is explained as well as the representation of floating point numbers. ASCII and Unicode representations for characters are described and discussed. In the third chapter, the CPU of a simple computer is described together with its machine language. Students are taught how to write programs in this simple machine language. The design of subprograms with standard conventions for subprogram calls and returns as well as parameter passing is discussed in details as it allows the student to better understand procedure calls and parameter passing methods in higher level programming languages. The fourth chapter addresses programming languages concepts in a more systematic way. The Java programming language is used to illustrate the concepts but no attempt is made to provide a global overview of Java. In contrast, a limited number of topics are studied with care and in details : primitive data types, literals, variables, arrays, the String class, static methods, expressions, a small subset of composed statements, primitive type conversions, method overloading. Classical algorithms for searching and sorting are build with this subset of Java.</p>
<p><b>Bibliography :</b></p>	<p>The course is based on the following reference book (in English): <span style="float: right;">D. Eck, Introduction to Programming Using Java,</span> Sixth Edition, 2013, <a href="http://math.hws.edu/javanotes/">http://math.hws.edu/javanotes/</a></p>
<p><b>Faculty or entity in charge:</b></p>	<p>INFO</p>

<b>Programmes / formations proposant cette unité d'enseignement (UE)</b>				
Intitulé du programme	Sigle	Credits	Prerequis	Acquis d'apprentissage
Bachelor in Computer Science	<a href="#">SINF1BA</a>	6	-	
Minor in Computer Sciences	<a href="#">LINFO100I</a>	5	-	
Master [120] in Anthropology	<a href="#">ANTR2M</a>	6	-	
Master [120] in Linguistics	<a href="#">LING2M</a>	5	-	
Additionnal module in Mathematics	<a href="#">LMATH100P</a>	6	-	