UCL Université catholique de Louvain

LSINF1161

2010-2011

Algorithmics and programming Part2

6.0 credits	30.0 h + 30.0 h	1q

Teacher(s) :	Le Charlier Baudouin ;
Language :	Français
Place of the course	Louvain-la-Neuve
Inline resources:	> http://www.icampus.ucl.ac.be/claroline/course/index.php?cid=SINF1150
Prerequisites :	-- Basic knowledge of programming in a high-level language -- Ability to base programming in high-level language on a rigorous analysis of the underlying concepts. These prerequisites are the aims of the course SINF1160
Main themes :	-- Input/output operations at the machine language level and its relational to multiprogramming -- Sequential files : binary files and text files -- Recursion -- Java classes to build graphical interfaces (AWT) -- Object oriented concepts illustrated with Java -- Logical and physical data structures -- Recursive data structures -- Linked data structures
Aims :	Students completing successfully this course will be able to -- build small programs that are correct right from the start without relying to a trial and error process, -- describe and justify the use of basic programming concepts such as files, graphical interfaces, linked and recursive data structures, -- describe basic fundamental concepts of object oriented programming, and provide example from Java. Students will have developed skills and operational methodology. In particular, they have developed their ability to -- have a critical look at their achievements and justify the steps of the reasoning that led them there *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".*
Evaluation methods :	-- Written exam -- Projects and homeworks are evaluated and this evaluation contributes to 30% of the final examination.
Teaching methods :	Teaching is organized with lectures, exercises and pratical work. The practical work consists of one part in sessions in classrooms where simple exercises are solved with the help of an assistant and, on the other hand, individual homeworks and / or small projects committed by groups of 2 students. The homeworks are to complete the exercises solved during classroom sessions up their implementation (including some comments and speficiations in the code). Projects are evaluated and this evaluation contributes to 30% of the final examination.
Content :	The material addressed in the lectures is as follows (presented in that order). -- Input/output operations at the machine language level are considered. Their study motivates the introduction of multiprogramming and consequently of a kernel operating system including a file system. -- Two data models for handling files in Java are proposed. Standard Java classes for files are too complicated to be used in this introductory course. Thus simpler classes based on a data model similar to Pascal files are used. They are completely specified. Only two classes are defined. The first one allows the students to manipulate text files while the other makes it possible to handle arbitrary binary files. -- Recursion is introduced as a reasoning method to build algorithms following a different approach (than previously). The role of local variables (including parameters) in this context is emphasized. -- The basic concept of a "class defined by the user" is introduced (for Java). This includes the following notions : instance variables, non static methods, constructors, accessibility of classes and class members. -- Inheritance is introduced as a most fundamental oriented object concept aiming at making it possible to reuse code in a simple and correct way. -- The underlying principles of standard libraries for building graphical interfaces are explained. This is an excellent topic to illustrate the use of inheritance. A very small number of AWT classes are described. But they are sufficient to build simple graphical interfaces and to understand the overall principles of such tools. The event model used by AWT is elicited in full. -- More "advanced" concepts of object-oriented programming (in Java) are explained : interfaces, abstract classes, abstract methods, dynamic method calls (how they work exactly). The handling of exceptions is also presented. -- The use of abstract classes to implement recursive data structures is illustrated using binary trees as an exemple. -- Java interfaces are related to the general notion of logical data type. -- Several definitions of lists are presented and discussed to illustrate and contrast notions such as logical versus physical data structures and their relations to interfaces and implementation classes. It is shown how the limited "multiple inheritance" mechanism

	of interfaces in Java allows one to use or reuse a physical data structure (a class) to implement a given logical data structure (an interface).
Bibliography :	-- transparents disponibles en ligne
Other infos :	
Cycle and year of study :	> Bachelor in Mathematics > Bachelor in Computer Science
Faculty or entity in charge:	INFO