LSINF1212

2014-2015

Computer science deepening project

5.0 credits	7.5 h + 45.0 h	2q

Teacher(s) :	Saerens Marco ; Gonzalez Montesinos Sebastian Andres (compensates Saerens Marco) ;
Language :	Français
Place of the course	Louvain-la-Neuve
Inline resources:	> http://icampus.uclouvain.be/claroline/course/index.php?cid=SINF2125
Prerequisites :	p { margin-bottom: 0.08in; } LSINF1102
Main themes :	The main course themes will be -- specification of user requirements, -- design of 3-tier architectures, including -- design of user interfaces, -- business model design, -- data structures and persistence; -- software quality control.
Aims :	p { margin-bottom: 0.08in; } The students who will succeed this course will be able to undertake the construction of a non-trivial information system, following elementary software engineering practices. More specifically, the students will develop their capacity to -- analyse user requirements to build an information system, -- systematically specify user requirements, -- devise a system architecture under the light of the specified requirements, -- implement the designed system in an object-oriented language, and -- control the quality of the system through functional tests. Additionally, the students will develop methodological and operational skills, such as -- team work: divide and coordinate development tasks, such that each team member is able to defend the project when confronted to a quality evaluation committee; -- convincingly demonstrate the system to potentially interested users; -- effectively perform a technical debreifing for future developers of the system. *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 :	p { margin-bottom: 0.08in; } The evaluation of the project will be based on the quality of -- analysis and design of the system (30%), including conformance to specified user requirements, software architecture, data structures and persistence, and the user interface; -- system implementation (30%), including maintainability (readability, documentation, tests) and efficiency; -- the final project report (10%). The project will be drawn to a close through an oral defence as follows: -- a first part demonstrating the functionality of the system (10%), -- a second part presenting the technical internals of the system (10%), and -- a round of questions and answers (10%). The students defending in second session will have to add functionality extensions that are commensurate with the additional time they will have with respect to the first session. The evaluation criteria will remain unchanged.
Teaching methods :	p { margin-bottom: 0.08in; } The project will be based on concepts introduced during practical sessions, as well as other concepts seen in parallel or previous courses. The progress of the project will be monitored through practical sessions carried out in a computer room, in which a first part will be dedicated to introduce base needed concepts in the form of a plenary lecture, and a second part will be practical, such that students will be able to ask questions and show the issues they have encountered so far. The project will be carried out in groups of 2 or 3 students. The choice of teammates will be open. The students of this course will be encouraged to submit and present their result in English, so as to improve their communication proficiency in this language when it comes to a technical matter. p { margin-bottom: 0.08in; }
Content :	p { margin-bottom: 0.08in; } The students will work in small groups to design, develop, demonstrate and make a technical debriefing of their system. A number of methodologies and tools will be introduced according the project needs, in particular: -- Object-oriented programming and asynchronous programming in JavaScript, -- Verifiable user requirements specification in Cucumber, -- Dynamic web interface design in HTML, CSS and JavaScript, -- Web service programming in node.js, -- Use of non-relational databases in MongoDB, -- Quality control through unit testing and source code static analysis. These contents are aimed at being complementary to those of other courses followed in parallel and previously. The goal is to widen the conceptual and technical spectrum that students can bring into play during their professional lives, in particular when confronted with the development of web-based applications.
Bibliography :	p { margin-bottom: 0.08in; }
Other infos :	Background : LSINF1101 basics of object-oriented programming LSINF1103 basics of algorithmics
Cycle and year of study :	> Bachelor in Computer Science > Master [120] in Linguistics
Faculty or entity in charge:	INFO