Content

Course Code	Course Name	Semester	Theory	Practice	Lab	Credit	ECTS
INF481	Software Engineering and Object Oriented Design	8	4	0	0	4	5

Prerequisites	
Admission Requirements	

Language of Instruction	French
Course Type	Compulsory
Course Level	Bachelor Degree
Objective	In this course, tools and techniques that can be used in object oriented design process are introduced and students are given the experience of applying them. These tools help design both visually and textually. Students gain the ability to use methods and tools that can increase productivity in any process that can be encountered in the life cycle of a software project. Students learn the place of software engineering in computer engineering. They understands the necessity of software design and then object oriented design. They can use UML, a visual design language accepted as world standard. They can design different software problems using UML language. Students will be able to describe the software development process and life cycle in detail and compare the software development processes used in the market. They can test the developed software and calculate its expected cost/effort.
Content	1. Introduction to software engineering and design, software demand analysis – 1 2. Software want analysis – 2, software design – 1 3. Software design – 2, software design – 3 4. Software design principles - 1, software design principles - 2 5. Software architectures 6. Design patterns 7. Design patterns, conceptual design presentations 8. UML class examples 9. Midterm exam 10. Software development models, agile software development methods 11. Software testing techniques 13. Estimating in software projects 14. Software quality 15. Term project presentations
References	1. Software Engineering, Ian Sommerville, Addison-Wesley, 10th Edition, 2015. 2. Introduction to Software Engineering Design, Processes, Principles, and Patterns with UML2, Christophe Fox, Addison-Wesley, 2006.

Theory Topics

Week	Weekly Contents
1	Introduction to software engineering and design
2	Software requirement analysis
3	Software design
4	Software design principles
5	Software architectures
6	Design patterns

Week	Weekly Contents
7	Conceptual design presentations
8	UML class examples
9	Midterm
10	Software development models
11	agile software development methods
12	Software testing techniques
13	Software quality
14	Term project presentations