

Course unit title:	Software Reuse			
Course unit code:	WSS553			
Type of course unit:	Elective			
Level of course unit:	Master (2nd Cycle)			
Year of study:	1			
Semester when the unit is delivered:	2 (Spring)			
Number of ECTS credits allocated :	10	Lectures:	3	Labs: 1
Name of lecturer(s):	Dr. Achilleas Achilleos			
Aim of the Course	<p>The aim of this course is to provide students with critical understanding of the technology, issues and challenges of software reuse at various levels. Specific focus in the course is dedicated to software reuse in web-based systems accessible via mobile devices. The course will enable students to practice software reuse at various levels, with different programming languages and on different platforms. In specific, the use of Java and HTML5 technologies will provide the capability to experience and practice software reuse on both desktop and mobile platforms, as well as at different levels such as object-oriented programming, component-based software development, middleware-based development, WS*-stack services, REST services and model-driven engineering. Finally, management of code repositories is introduced at the last week. In overall, the objective of the course is to enhance critical awareness, promote practical thinking and reasoning in order to solve practical problems through the reuse of software systems.</p>			
Learning outcomes of the course unit:	<p>Upon successful completion of the course students will be able to:</p> <ul style="list-style-type: none"> • Understand the concepts, principles and methods of software reuse. • Familiarise and understand the different levels of software reuse. • Make proper use of ready-made software components in practice. • Gain theoretical knowledge and analytical skills to develop applications by employing reuse methods at code, component, design and models levels. • Learn how to reuse Java and Web components. • Distribute effectively the results of their work to other developers through the use of software repositories. 			
Mode of delivery:	Face-to-face			
Prerequisites:	None	Co-requisites:	None	
Course contents:	<p>Unit 1. Introduction to Software Reuse and the different levels and types of reuse.</p> <ul style="list-style-type: none"> - Software Reuse Key Concepts. - Levels and Types of Software Reuse. - The Software Reuse Landscape. - Software Reuse Approaches. - Reuse Benefits, Issues and Economics. 			

	<p>Unit 2. Software Reuse through Object Oriented Programming (OOP) and Component Based Software Engineering (CBSE).</p> <ul style="list-style-type: none"> - Revisiting key concepts of OOP. - Practical example of reuse through OOP. - Reuse through the Java Collections Framework. - Introduction to the principles and concepts of CBSE. - JavaBeans: Software Reuse at the level of CBSE. - Practical example of reuse through JavaBeans. <p>Unit 3. Design Patterns: Reusing Best Practices to Solve Common Design Problems.</p> <ul style="list-style-type: none"> - Design Principles and Patterns. - Design Patterns: Concepts and Types. - Building Successful Mobile Applications using Design Patterns. <p>Unit 4. Software Reuse via the notion of a Middleware.</p> <ul style="list-style-type: none"> - Motivation, definition and the role of a middleware. - Examining a simple middleware architecture: RPC. - Challenges in middleware design. - Example: HTML5 Context Middleware (H5CM). <p>Unit 5. The WS*-stack.</p> <ul style="list-style-type: none"> - Motivation, History and Concepts. - The Web Service Model. - Web Service Standards - WS*-stack (WSDL, SOAP, XML, UDDI). <p>Unit 6. RESTful Services.</p> <ul style="list-style-type: none"> - REST Motivation, Definition and Principles. - REST Vocabulary and Concepts. - REST Vs. WS*-stack. <p>Unit 6. Model Driven Engineering.</p> <ul style="list-style-type: none"> - Introduction to the notion of models reuse. - Unified Modelling Language and Domain Specific Modelling. - Model-driven engineering and MDA architecture. - Models transformation and code generators. <p>Unit 8. Software Repositories for reusing Software Assets.</p> <ul style="list-style-type: none"> - Definition, Requirement and Advantages of a Software Repository - The Software Repository Model. - Main functions of a Software Repository. - Version Control Systems. - Creating and Managing a Software Repository.
Planned learning activities and teaching methods:	The methodology used to conduct the course is structured around lectures and laboratory exercises, so that students gain theoretical knowledge as well as practical skills. The taught part of course is delivered to the students with the help of computer presentations. Lecture notes and presentations are available through the web for students to use in combination with the textbooks. Furthermore theoretical principles are explained by means of specific examples and solution of

	<p>specific problems using practical examples.</p> <p>Lectures are supplemented with supervised and unsupervised computer laboratory. Laboratories will include demonstrations of taught concepts and experimentation with related technologies. Additionally, during laboratory sessions, students apply their gained knowledge and identify the principles taught in the lecture sessions by means of working on different tasks and problems. Students are also allocated exercises for homework, assignments and/or group project to improve both their individual skills and team work.</p>
Bibliography	
Textbooks:	<ol style="list-style-type: none"> 1. "Practical Software Reuse" (Practitioner Series), Michel Ezran, Maurizio Morisio, Colin Tully, Publisher: Springer, 2002. 2. "Head First Design Patterns, Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra, O'Reilly Media, 2004. 3. "Object-Oriented Design and Patterns", Cay Horstmann, Wiley, 2nd edition, 2006. 4. "Mobile Design Pattern Gallery, 2nd Edition, UI Patterns for Smartphone Apps", Theresa Neil, Publisher: O'Reilly Media, Final Release Date: April 2014, Pages: 408, Available Online: 5. "Service-Oriented Architecture (SOA): Concepts, Technology, and Design", T. Erl, Pearson Education, 2005.
References:	<ol style="list-style-type: none"> 1. "Why Software Reuse has Failed and How to Make It Work for You", Douglas C. Schmidt, Available Online: Link. 2. "Design patterns, the big picture, Part 1: Design pattern history and classification", Jeff Friesen, JavaWorld Nov 21, 2012, Available Online: Link. 3. "Design patterns, the big picture, Part 2: Gang-of-four classics revisited", Jeff Friesen, JavaWorld Dec 26, 2012, Available Online: Link. 4. "Design patterns, the big picture, Part 3: Beyond software design patterns", Jeff Friesen, JavaWorld Nov 21, 2012, Available Online: Link. 5. "Mobile UI Design Patterns – A Deeper Look At The Hottest Apps Today", Dominik Pacholczyk, UXPin, Available Online: Link.
Assessment methods and criteria:	<ul style="list-style-type: none"> • Project Work: 40% • Presentation: 10% • Final Exam: 50%
Language of instruction:	English
Work placement(s):	No