Course Title	**Software Reuse**				
Course Code	**DLWSS553**				
Course Type	**Elective**				
Level	Master (2nd Cycle) – Distance Learning				
Year / Semester	2 / 3				
Teacher's Name	**Achilleas Achilleos, PhD**				
ECTS	10	Lectures / week	3	Laboratories/week	-
Course Purpose	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 to solve practical problems through the reuse of software systems.				
Learning Outcomes	Upon successful completion of the course students will be able to: • Understand the concepts, principles and methods of software reuse and argue on the importance of software reuse in building modern software systems. • Outline and describe the different levels of software reuse: object-oriented programming, component-based engineering, middleware, WS*-stack services, REST services and model-driven engineering. • Identify, analyse and reuse open-source software tools in practice and at different software reuse levels. • Gain theoretical knowledge and analytical skills to develop applications by employing reuse methods at code, component, design and models levels. • Distribute effectively the results of their work to other developers using software repositories in order to promote software reuse. • Describe and explain the concept of open-source software development and argue on the importance of software licensing.				
Prerequisites	**None.**		Corequisites	**None.**	
Course Content	This course consists of seven units that will be taught within twelve (12) weeks:				

	• **Unit 1 (Weeks 1-2) Introduction to Software Reuse:** Software Reuse Key Concepts. Levels and Types of Software Reuse. The Software Reuse Landscape. Software Reuse Approaches. Reuse Benefits, Issues and Economics. • **Unit 2 (Weeks 3-4) Object Oriented Programming and Component Based Software Engineering:** Revisiting key concepts of Object-Oriented Programming (OOP). Practical example of reuse through OOP. Reuse through the Java Collections Framework. Introduction to the principles and concepts of Component Based Software Engineering (CBSE). JavaBeans: Software Reuse at the level of CBSE. Practical example of reuse through JavaBeans. • **Unit 3 (Weeks 5-7) Design Patterns: Reusing Best Practices to Solve Common Design Problems:** Design Principles and Patterns. Design Patterns: Concepts and Types. Building Successful Mobile Applications using Design Patterns. • **Unit 4 (Week 8) Software Reuse via the notion of a Middleware:** Motivation, definition and the role of a middleware. Examining a simple middleware architecture: RPC. Challenges in middleware design. Example: HTML5 Context Middleware (H5CM). • **Unit 5 (Weeks 9-10) Service Reusability:** Motivation, History and Concepts. The Web Service Model. Web Service Standards - WS*-stack (WSDL, SOAP, XML, UDDI). RESTful Services. REST Motivation, Definition and Principles. REST Vocabulary and Concepts. REST Vs. WS*-stack. • **Unit 6 (Week 11) Model Driven Engineering:** 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. • **Unit 7 (Week 12) Software Repositories (1 Week):** Definition. Reusing Software Assets. Requirements 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. Open-Source Software Development. Software Licensing.
Teaching Methodology	**Mode of Delivery: Distance Learning** The course is designed to introduce and explain the material students are expected to learn through an on-line learning environment. The on-line environment provides an opportunity for receiving on-line feedback from the Course Instructor during their study. In addition, students will be encouraged to interact both with other students and the instructor so as to feel part of an on-line community of learners that belong to the University network. The course content will be delivered through online material/notes, recorded lectures and/or narrated presentations. Therefore, students may be asked to download and study notes, tutorials and numerical exercises as well as watch recorded lectures/demonstrations or narrated presentations posted on the web addressing the main concepts of a particular unit. Furthermore, the planned communication and the dynamic/online interaction activities between the course instructor and the students will include asynchronous communication tools (Discussion Forum) that students may be asked to participate, wherever appropriate, in an online forum posting their views on certain topics covered in a particular unit; and synchronous communication tools (instant messaging, such as Skype, chat rooms, video-

	conferencing, etc.), that students may discuss on-line with the Instructor (s) and/or other students specific issues covered in a given unit. Moreover, a number of case study readings are also considered, so as to demonstrate the relevance and practical applicability software reuse methods and systems covered in the various units of this course. Case-studies can illustrate that what students have studied in each unit is not just of academic or theoretical value but also has value in terms of improving real-life challenges.
Bibliography	**Compulsory Bibliography** 1. Michel Ezran, Maurizio Morisio, Colin Tully, "Practical Software Reuse" (Practitioner Series), Paperback: 216 pages, Publisher: Springer; 1st edition (April 2, 2002), Language: English, ISBN-10: 1852335025, ISBN-13: 978-1852335021. 2. John Vlissides, Ralph Johnson, Richard Helm, Erich Gamma, "Design Patterns: Elements of Reusable Object-Oriented Software", Publisher: Addison-Wesley Professional, Release Date: October 1994, ISBN: 0201633612. **Additional / Complimentary Bibliography** 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	The Students are assessed via continuous assessment throughout the duration of the Semester, which forms the Coursework grade and the final written exam. The coursework and the final exam grades are weighted 50% and 50%, respectively, and compose the final grade of the course. Various approaches are used for the continuous assessment of the students, such as dynamic online activities, online quizzes, group project design, implementation and presentation. The assessment weight, date and time of each type of continuous assessment is being set at the beginning of the semester via the course outline. An indicative weighted continuous assessment of the course is shown below: • **1st Online activity – Online Quiz** (10% of total marks for module) • **2nd Online activity – Research Papers Study & online quiz** (10% of total marks for module) • **Two marked assignments** (30% of total marks for module) • **One closed-book, 3-hour final exam** (50% of total marks for

	module) Students are prepared for final exam, by revision on the matter taught, problem solving and concept testing and are also trained to be able to deal with time constraints and revision timetable. The criteria considered for the assessment of each type of the continuous assessment and the final exam of the course are: (i) the comprehension of the fundamental concepts and theory of each topic, (ii) the application of the theory in solving related problems and (iii) the ability to apply the above knowledge in complex real-life problems. The final assessment of the students is formative and summative and is assured to comply with the subject's expected learning outcomes and the quality of the course.
Language	English.