

Course Title	Software Methodology				
Course Code	ACSC359				
Course Type	Compulsory				
Level	BSc (Level 1)				
Year / Semester	3 <sup>rd</sup> (Spring)				
Teacher's Name	Dr. Andreas Constantinides				
ECTS	6	Lectures / week	2	Laboratories/week	2
Learning Outcomes	<p>Upon successful completion of the course students will be able to:</p> <ol style="list-style-type: none"> <li>1. Describe modern practices in the development of software and introduce the principles of software engineering.</li> <li>2. Explain the key aspects for developing robust software including modularity, object orientation, reusability and human interface design.</li> <li>3. Explain and critically evaluate the key process models for constructing software (waterfall, spiral, prototyping, agile).</li> <li>4. Examine project management and feasibility analysis issues related to software projects.</li> <li>5. Explain and show the ability to implement concepts related to the design and utilization of database management systems using SQL.</li> <li>6. Demonstrate basic knowledge in developing Java Object-oriented applications.</li> <li>7. Demonstrate basic knowledge on the integration of Java and SQL using JDBC.</li> </ol>				
Prerequisites	ACSC183		Co-requisites	None	
Course Content	<ul style="list-style-type: none"> <li>• <b>Principles of Software Engineering: Design</b> – User Interface (Menus and GUIs), Design Principles and Guidelines Interface Design. <b>Modularity</b> – Components type, size and complexity, Information hiding, Coupling and cohesion, OO Programming. <b>Structured Programming</b> – Selecting Control Structures. <b>Data Flow Design</b> – Identifying data flows, Creation of structure chart. Data Structure Design. OO Design. <b>Process Models</b> – The Waterfall Model. The spiral model. Prototyping. Incremental Development. Open-Source. Extreme Programming. The unified Process. <b>Project Management</b> – Teams. Software Metrics and quality assurance. Management.</li> <li>• <b>Introduction to Databases:</b> Databases and Database Users, Database System Concepts and Architecture, Data Modeling Using the Entity-Relationship (ER) Model, Conceptual Model, The Enhanced Entity-Relationship (EER) Model, The Relational Data Model and Relational Database Constraints, Relational Database Design by ER, EER-to-Relational Mapping, Normalization for Relational Databases.</li> <li>• <b>Introduction to Structured Query Language (SQL):</b> Queries ( Select, From, Where clause, Like clause, Having clause, Order By clause, Count, Group By,</li> </ul>				

	<p>Join Inner/Right, Outer/Left Join). <b>DDL</b>: Create Table, Alter Table, Truncate Table, Drop Table. <b>DML</b>: Insert Into statement, Update statement, Delete From statement.</p> <ul style="list-style-type: none"> <li>• <b>Java Programming Language</b>: History, Introduction to Java, Selecting a Development Tool (JDK, SDK etc.), JDK Contents (java, javac, javadoc etc) Compile and run from command line. NetBeans. Encapsulation. Library Classes. Inheritance. Polymorphism. Packages. Abstraction and Interfaces. Exceptions and exception handlers. Input/output Mechanisms. Objects and Instances. Constructors. Setters and Getters.</li> <li>• <b>Java Database Connectivity (JDBC)</b>: Connecting to JDBC data source. Querying JDBC data source. Manipulating databases with JDBC. Example Application.</li> <li>• <b>Laboratory Work</b>: Design and develop fully functioning and well-behaved database systems using SQL, applications using Java Object-oriented programming language as well as their integration using JDBC.</li> </ul>
Teaching Methodology	<p>The course is structured around lectures, assignments, laboratory exercises and individual work. During the lectures, students are encouraged to participate in discussions enabling the exchange of ideas and examples. Laboratory exercises are handed to students and their solutions are discussed at laboratory periods. Additional tutorial time at the end of each lecture is provided to students as well as additional notes for each section of the course and worksheets, which process in the lab or as homework. Students are expected to demonstrate the necessary effort to become confident with the different concepts and topics of the course.</p>
Bibliography	<p><u>Textbooks:</u></p> <ul style="list-style-type: none"> <li>• Wu, C. Thomas, An Introduction to Object-Oriented Programming with Java, 5th edition, McGraw-Hill, 2010.</li> <li>• Thomas M. Connolly, Carolyn E. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition, Addison-Wesley Publishing, 2015, ISBN: 978-0132943260.</li> </ul> <p><u>References:</u></p> <ul style="list-style-type: none"> <li>• SQL Database Documentation</li> <li>• Java Object-Oriented Programming Documentation</li> </ul>
Assessment	<ul style="list-style-type: none"> <li>• Assignments 10%</li> <li>• Tests: 20%</li> <li>• Laboratory Work: 10%</li> <li>• Final Exam 60%</li> </ul>
Language	English