

## H8OOS: Object Oriented Software Engineering

<b>Module Code:</b>	H8OOS
<b>Long Title</b>	Object Oriented Software Engineering <b>APPROVED</b>
<b>Title</b>	Object Oriented Software Engineering
<b>Module Level:</b>	LEVEL 8
<b>EQF Level:</b>	6
<b>EHEA Level:</b>	First Cycle
<b>Credits:</b>	5
<b>Module Coordinator:</b>	ANTHONY PAUL STYNES
<b>Module Author:</b>	ANTHONY PAUL STYNES
<b>Departments:</b>	School of Computing
<b>Specifications of the qualifications and experience required of staff</b>	Master's degree in computing or cognate discipline. Proposed lecturer: Dr Paul Stynes
<b>Learning Outcomes</b>	
<i>On successful completion of this module the learner will be able to:</i>	
<b>#</b>	<b>Learning Outcome Description</b>
LO1	Demonstrate the conceptual, practical and technical skills of planning and monitoring a project plan using an appropriate CASE tool
LO2	Describe in detail the theory, concepts and methods pertaining to Software Engineering such as Agile and UML.
LO3	Create requirements using use case modelling concepts.
LO4	Demonstrate conceptual and technical skills in the analysis, design, implementation and test of a software development solution individually or as part of a team.
LO5	Employ tools and techniques for Software Engineering,
<b>Dependencies</b>	
<b>Module Recommendations</b>	
No recommendations listed	
<b>Co-requisite Modules</b>	
No Co-requisite modules listed	
<b>Entry requirements</b>	See section 4.2 Entry procedures and criteria for the programme including procedures recognition of prior learning

# H8OOS: Object Oriented Software Engineering

Module Content & Assessment			
<b>Indicative Content</b>			
<b>Software Engineering</b> • Principles of Software Engineering. • Software Product. • Software Process.			
<b>Software Project Management</b> • Project Management Concepts • Process, Project and Product Metrics • Software Project Organization • Recruiting and Staffing projects • Project roles and the project team • Project planning methods • Risk analysis and risk management • Project scheduling and Tracking • Project inspections and walkthroughs • Coping with change – change models • Software Quality Assurance			
<b>Requirements Engineering</b> • Use case modelling. • Actors Use Cases. • Anatomy of use cases. • Advanced Use Case Concepts.			
<b>Structural Diagrams</b> • Class Diagram. • Component Diagram. • Package Diagram. • Deployment Diagram.			
<b>Behavioral Diagrams</b> • Sequence Diagram. • Communication Diagram. • Statechart.			
<b>Design Patterns</b> • Expert Pattern. • Creator Pattern. • Controller Pattern. • Three tier Architectures. • Multi-tiered Architectures. • Model view separation pattern. • Mapping to code.			
<b>Testing</b> • Software testing strategies. • Software Testing Techniques. • Unit Testing.			
<b>Test Driven Development</b> • Test Driven Development (TDD) Approach.			
<b>Agile Process Models</b> • Agility Agile Process. • Agile Process Models. • Overview of the SCRUM process.			
<b>Assessment Breakdown</b>			<b>%</b>
Coursework			100.00%
<b>Assessments</b>			
<b>Full Time</b>			
<b>Coursework</b>			
<b>Assessment Type:</b>	Project	<b>% of total:</b>	50
<b>Assessment Date:</b>	Week 8	<b>Outcome addressed:</b>	1,2,3,4,5
<b>Non-Marked:</b>	No		
<b>Assessment Description:</b> Team project requiring learners to project plan a project and apply UML diagrams to the resolution of a real life problem.			
<b>Assessment Type:</b>	Project	<b>% of total:</b>	50
<b>Assessment Date:</b>	Sem 1 End	<b>Outcome addressed:</b>	2,4,5
<b>Non-Marked:</b>	No		
<b>Assessment Description:</b> Team project requiring learners to apply agile methods such as TDD and SCRUM to the resolution of a real life problem.			
No End of Module Assessment			
No Workplace Assessment			
<b>Reassessment Requirement</b>			
<b>Coursework Only</b> <i>This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.</i>			
<b>Reassessment Description</b> This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination. Learners who fail this module will be required to sit a repeat module assessment where all learning outcomes will be examined.			

## H8OOS: Object Oriented Software Engineering

Module Workload				
Module Target Workload Hours 0 Hours				
Workload: Full Time				
Workload Type	Workload Description	Hours	Frequency	Average Weekly Learner Workload
Lecture	No Description	24	Per Semester	2.00
Tutorial	No Description	12	Per Semester	1.00
Independent Learning	No Description	89	Per Semester	7.42
Total Weekly Contact Hours				3.00
Workload: Part Time				
Workload Type	Workload Description	Hours	Frequency	Average Weekly Learner Workload
Lecture	No Description	24	Per Semester	2.00
Tutorial	No Description	12	Per Semester	1.00
Independent Learning Time	No Description	89	Per Semester	7.42
Total Weekly Contact Hours				3.00

Module Resources	
<i>Recommended Book Resources</i>	
<p>Alan Dennis, Barbara Haley Wixom, David Tegarden. (2015), Systems Analysis and Design with UML, 5. Wiley.</p> <p>Shore, J., and Warden, S. (2014), The Art of Agile Development, 1. O'Reilly Media.</p> <p>Kent Beck. (2003), Test-driven Development, Addison-Wesley Professional, p.220, [ISBN: 9780321146533].</p>	
<i>Supplementary Book Resources</i>	
<p>Axel van Lamsweerde. (2009), requirements Engineering: From System Goals to UML Models to Software Specifications, Wiley.</p> <p>Dragan Milicev. (2009), Model-Driven Development with Executable UML, Wiley.</p> <p>Eric Baude, Michael Bernstein. (2011), Software Engineering Modern Approaches, 2. Wiley.</p> <p>Roger S. Pressman. (2010), Software Engineering: A practitioners approach, 7. McGraw and Hill.</p>	
<i>This module does not have any article/paper resources</i>	
<i>Other Resources</i>	
<p>[Website], <a href="http://www-01.ibm.com/software/rational/uml/">http://www-01.ibm.com/software/rational/uml/</a>.</p>	
Discussion Note:	