

<b>Module Title:</b>	Systems Analysis and Testing
<b>Language of Instruction:</b>	English
<b>Credits:</b>	10
<b>NFQ Level:</b>	6
<b>Module Delivered In</b>	<a href="#">3 programme(s)</a>
<b>Teaching &amp; Learning Strategies:</b>	Mix of lectures, tutorials and workshops. Formative and summative assessments. A suitable case study will be selected and used throughout the course.
<b>Module Aim:</b>	To introduce students to the principles and practice of object-oriented systems analysis and testing.
<b>Learning Outcomes</b>	
<i>On successful completion of this module the learner should be able to:</i>	
LO1	Understand the principal software development processes.
LO2	Apply objected-oriented techniques to the analysis of a software system.
LO3	Use the principal UML diagramming techniques.
LO4	Understand and apply software testing techniques.
<b>Pre-requisite learning</b>	
<b>Module Recommendations</b>	
<i>This is prior learning (or a practical skill) that is recommended before enrolment in this module.</i>	
No recommendations listed	
<b>Incompatible Modules</b>	
<i>These are modules which have learning outcomes that are too similar to the learning outcomes of this module.</i>	
No incompatible modules listed	
<b>Co-requisite Modules</b>	
No Co-requisite modules listed	
<b>Requirements</b>	
<i>This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed.</i>	
No requirements listed	

## Module Content & Assessment

Indicative Content
<b>Approaches to Systems Analysis &amp; Design</b> Overview - structured, object-oriented, soft system, other.
<b>Object Oriented Concepts</b> Objects. Classes. Relationships - association, aggregation, inheritance. Encapsulation. Polymorphism.
<b>Software Development Processes</b> Waterfall vs Agile. Unified Process - iterative, evolutionary development; phases - inception, elaboration, construction, transition.
<b>Requirements</b> Fact-finding techniques - functional & non-functional requirements, FURPS+. Use Cases - brief and detailed formats. Use case diagrams. Tool (e.g. UMLet). System sequence diagrams. Collaboration diagrams. Domain Models - domain classes, attributes and associations. CRC cards. Animate using object diagrams.
<b>Implementation</b> System testing. Data conversion. User manuals. Training. Changeover strategies - direct, parallel, phased, pilot.
<b>Review &amp; Maintenance</b> System review. Evaluation Report. Amendment procedures.
<b>Testing</b> White-box and black-box methods. Levels of testing - unit, integration, system, acceptance. Test plans, schedules & reports. Testing tools.

Assessment Breakdown	%
Continuous Assessment	50.00%
End of Module Formal Examination	50.00%

Continuous Assessment				
Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Examination	Individual Written Exam	1,2	20.00	Week 5
Case Studies	Take-home Case Study Exercise	2,3,4	30.00	Week 10

No Project

No Practical

End of Module Formal Examination				
Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Formal Exam	Terminal Examination	1,2,3,4	50.00	End-of-Semester

SETU Carlow Campus reserves the right to alter the nature and timings of assessment

**Module Workload**

<b>Workload: Full Time</b>		
<i>Workload Type</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecturer Supervised Learning	12 Weeks per Stage	6.00
Estimated Learner Hours	15 Weeks per Stage	11.87
Total Hours		250.00

**Module Delivered In**

Programme Code	Programme	Semester	Delivery
CW_KCSOF_B	<a href="#">Bachelor of Science (Honours) in Software Development</a>	3	Mandatory
CW_KCSOF_D	<a href="#">Bachelor of Science in Software Development</a>	3	Mandatory
CW_KCCOM_C	<a href="#">Higher Certificate in Science in Computing Programming</a>	3	Mandatory