



## MODULE SPECIFICATION

Part 1: Information			
Module Title	Object Oriented Software Design and Development I		
Module Code	UFCFUM-15-2	Level	Level 5
For implementation from	2020-21		
UWE Credit Rating	15	ECTS Credit Rating	7.5
Faculty	Faculty of Environment & Technology	Field	Computer Science and Creative Technologies
Department	FET Dept of Computer Sci & Creative Tech		
Module type:	Standard		
Pre-requisites	None		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description
<p><b>Overview:</b> Students will learn the basic concepts of software design, data structures, programming, problem solving, programming logic, and fundamental software design techniques. This will include a review of traditional and contemporary software development methods including agile development. They will develop a holistic view of software engineering practice including gathering requirements, designing a solution, implementing a solution in a programming language, testing the completed application and deploying the solution to end users.</p> <p><b>Educational Aims:</b> The purpose of this topic is to introduce the students to the fundamental concepts of systems development through programming, computational thinking and data structures. They will analyse models of application development so that they can understand the key processes related to building functioning applications and appreciate the complexity of application development.</p> <p><b>Outline Syllabus:</b> Using an industry recognised language students will:</p> <p>Demonstrate an understanding of object-oriented concepts (e.g. classes, objects, inheritance, polymorphism, encapsulation)</p> <p>Perform object-oriented analysis and design. This will need to incorporate valid object-oriented designs (e.g. use case, user stories)</p>

## STUDENT AND ACADEMIC SERVICES

Object-oriented Program Development, to a defined business' requirement

Software artefacts

Apply good business practice in all areas of the development life cycle

The use of an appropriate object oriented testing facility and tracking to be able to debug created program code to understand and rectify problems within the code (e.g. white box,black box, unit testing)

**Teaching and Learning Methods:** Introductory lectures are supported by seminars, case studies, visits and practical workshops. In addition this module will be supported by interactive forums and learning tools.

Independent learning includes hours engaged with essential reading, case study preparation, assignment preparation and completion. Study time will be organised each week with a series of both essential and further readings and preparation for practical workshops.

This unit is practically based and designed to ensure that students understand and develop their skills in advanced programming techniques. Students will use the object-oriented facilities within C++ as a vehicle for this.

Scheduled learning will typically include lectures, seminars, supervision, external visits and an interactive forum.

All students are expected to attend a series of tutorials.

### Part 3: Assessment

This module is assessed by a combination of techniques: an examination (3 hours) and a practical build.

Component A – Exam

Students will be required to sit a 3-hour exam that will require knowledge of the following object-oriented techniques, classes, objects, inheritance, polymorphism and encapsulation.

Students will be required to perform object-oriented design using a taught methodology, against a business specification. They will need to employ various designs such as use case and user stories.

Component B – Practical Build

Students will be given a business specification from which they will produce a solution. They will need to design their systems and apply their knowledge of the development lifecycle models to create a sound system.

The task will include development, implementing, testing and debugging. The testing of the program will need to be robust and thorough, using techniques such as white and black box testing. The program must be fully object-oriented to ensure a sound understanding of the benefits that are associated with object-oriented programming. After the testing, Students will be required to modify their existing code base and apply new fixes or modules from their requirements and testing. The program will need to be fully documented and conform to industry standards.

Opportunities for formative assessment exist for the assessment strategy used. Verbal feedback and written feedback is given to all students providing a personal platform for improvement.

## STUDENT AND ACADEMIC SERVICES

First Sit Components	Final Assessment	Element weighting	Description
Examination - Component A	✓	50 %	Examination (3 hours)
Portfolio - Component B		50 %	Design, implement, test and correct a problem specification
Resit Components	Final Assessment	Element weighting	Description
Examination - Component A	✓	50 %	Examination (3 hours)
Portfolio - Component B		50 %	Design, implement, test and correct a problem specification

### Part 4: Teaching and Learning Methods

Learning Outcomes	On successful completion of this module students will achieve the following learning outcomes:	
	<b>Module Learning Outcomes</b>	<b>Reference</b>
	Describe the theory and concepts of the object-oriented paradigm	MO1
	Understand software design approaches and patterns and can interpret and implement a given design	MO2
	Analyse business and technical requirements and select appropriate solutions	MO3
	Create analysis artefacts, such as Use Cases and/or User Stories	MO4
	Implement, test, and debug software to meet a requirements specification	MO5
	Develop moderately complex software solutions and software modifications to specified requirements	MO6
	Debug own code and understand structure of programmes in order to identify and resolve issues	MO7
	Identify and apply best practices and standards	MO8
Contact Hours	<b>Independent Study Hours:</b>	
	Independent study/self-guided study	114
	<b>Total Independent Study Hours:</b>	114
	<b>Scheduled Learning and Teaching Hours:</b>	
	Face-to-face learning	36
	<b>Total Scheduled Learning and Teaching Hours:</b>	36
	<b>Hours to be allocated</b>	150
	<b>Allocated Hours</b>	150

## STUDENT AND ACADEMIC SERVICES

Reading List	<p><i>The reading list for this module can be accessed via the following link:</i></p> <p><a href="https://rl.talis.com/3/uwe/lists/191A5CA5-84FA-5EF6-E16C-CCD8C3E9D807.html">https://rl.talis.com/3/uwe/lists/191A5CA5-84FA-5EF6-E16C-CCD8C3E9D807.html</a></p>
--------------	---

### **Part 5: Contributes Towards**

This module contributes towards the following programmes of study: