STUDENT AND ACADEMIC SERVICES

MODULE SPECIFICATION

Part 1: Information			
Module Title	Object Oriented Software Design and Development Ii		
Module Code	UFCFYM-15-2	Level	Level 5
For implementation from	2018-19		
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		
Contributes towards			
Module type:	Standard		
Pre-requisites	None		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description

Overview: The purpose of this topic is to introduce the apprentices 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.

Educational Aims: Apprentices 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.

Outline Syllabus: The syllabus includes:

Demonstrate an understanding of object-oriented concepts

Outline the general trends in software development, and identify the perceived advantages of object-oriented techniques (e.g. modularity, encapsulation, reuse, iterative development, interactivity, greater client involvement in design. Identification of objects, classification, inheritance, polymorphism)

Perform object-oriented analysis and design

Develop modelling techniques appropriate to object-oriented design (e.g. object diagrams, class diagrams, use cases, state diagrams, scenarios, sequence diagrams, collaboration diagrams, CRC cards and appropriate use of data dictionary)

Emphasis will be placed on UML and the use of a Case Tool

Identify and select the most appropriate paradigm for a business case study, explaining why that paradigm will be the most relevant (e.g. Event Driven, Procedural, Object-Oriented)

Create a comprehensive test plan that utilises unit tests, ensuring errors are analysed to help correct errors

Develop complex software using two cohesive languages while utilising at least two different paradigms

Design appropriate usability study

Adhere to industry standards (e.g. Code Documentation, Naming Conventions)

Object-oriented Program Development Code features

Style and structure

Syntax and semantics

Prepare code for classes to be re-used in other applications

Object-oriented Program Evaluation Conduct tests

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.

150 hours study time of which 36 hours will represent scheduled learning.

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

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

Contact Hours:

36 hours scheduled learning
114 hours research, independent study and preparation for assessment work

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

All apprentices 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
Apprentices will be required to sit a 3-hour exam that will require knowledge of the both object-oriented and procedural programming techniques.

Apprentices will be required to perform object-oriented design using a taught methodology, against a business specification. They will need to employ data models and designs they have been taught such as sequence and state diagrams.

Apprentices will be required to create a structurally sound program using their own selected paradigm (event driven, procedural or object-orientated) to a set of business requirements.

Component B – Practical Build
Apprentices 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 program should consist of at least two different languages, across two different paradigms, working cohesively to create a robust program. The testing of the program will need to be robust and thorough, using unit testing. Apprentices will be required to track the errors found by unit testing, and evaluate, applying corrections as required. 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 apprentices providing a personal platform for improvement.

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

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will be able to:

	Module Learning Outcomes
MO1	Apply procedural and/or object-oriented programming techniques.
MO2	Identify, explain, and use appropriately business and technical requirements and select appropriate solutions.
MO3	Select the relevant paradigm (for example Object-Oriented, Event Driven or Procedural) for a given set of business requirements.
MO4	Create data models and software designs to effectively communicate understanding of the programme.
MO5	Design, implement, test, and debug complex software solutions to meet a requirements specification.
MO6	Write good quality code (logic) with sound syntax in at least two languages with different paradigms.
MO7	Develop complex software solutions and software modifications to specified requirements.
MO8	Test code and analyse results to correct errors found using unit testing.
MO9	Apply underlying concepts and the principles of best practices and standards.

Contact Hours

Contact Hours	
Independent Study Hours:	
Independent study/self-guided study	114
Total Independent Study Hours:	114
Scheduled Learning and Teaching Hours:	
Face-to-face learning	36
Total Scheduled Learning and Teaching Hours:	36
Hours to be allocated	150
Allocated Hours	150

Reading List

The reading list for this module can be accessed via the following link:

https://uwe.rl.talis.com/index.html