STUDENT AND ACADEMIC SERVICES

UWE Bristol | University of the West of England

MODULE SPECIFICATION

Part 1: Information			
Module Title	Object Oriented Software Design and Development		
Module Code	UFCFME-30-2	Level	Level 5
For implementation from	2018-19		
UWE Credit Rating	30	ECTS Credit Rating	15
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	Software Design and Development 2018-19, Systems Analysis and Databases 2018-19		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description
Educational Aims: See Learning Outcomes **Outline Syllabus:** Using C++ programming language at an advanced level: 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, re-use, 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.

Develop and document a test plan for an object-oriented system:
Select and describe an appropriate O/O testing strategy.
Produce a detailed test plan and supporting documentation.
Design appropriate usability study.

Object-oriented Program Development:
Code features.
Style and structure.
Syntax and semantics.
Control structures.
Class development.
Prepare code for classes to be re-used in other applications.

Object-oriented Program Evaluation:
Conduct tests.
Complete test logs.
Document classes and code.

Maintain version control.

Teaching and Learning Methods: 108 hours scheduled learning.
192 hours research, independent study and preparation for assessment work.

Scheduled learning will typically include lectures, seminars, supervision and an interactive forum.
All students are expected to attend a series of tutorials.

Independent learning includes hours engaged with essential reading, case study preparation, assignment preparation and completion. Student 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.

Part 3: Assessment

A range of assessment techniques will be employed to ensure that learners can meet the breadth of learning outcomes presented in this module alongside the ability to demonstrate transferable skills e.g. communication skills.

Examination: Unseen examination questions on object-oriented concepts.

Object-Oriented Software Development: Based on the design created for component A, students will be expected to implement a practical solution in C++, test, evaluate and document the outcome.

Opportunities for formative assessment exist for the assessment strategy used. Verbal feedback is given and all students will engage with personalised tutorials setting SMART targets as part of the programme design.

First Sit Components	Final Assessment	Element weighting	Description
Practical Skills Assessment - Component B		50 %	Design, implement, test and evaluate a problem specification
Examination - Component A	✓	50 %	Examination (2 hours)
Resit Components	Final Assessment	Element weighting	Description

STUDENT AND ACADEMIC SERVICES

Practical Skills Assessment - Component B		50 %	Design, implement, test and evaluate a problem specification
Examination - Component A	✓	50 %	Exam on the object-oriented concepts

Part 4: Teaching and Learning Methods	
Learning Outcomes	On successful completion of this module students will be able to:

	Module Learning Outcomes
MO1	Demonstrate an understanding of what constitutes a good object-oriented system
MO2	Use an object-oriented methodology, e.g. UML, to design a practical solution to a business-related problem
MO3	Apply OOP techniques to implement the practical solution designed
MO4	Show proficiency in code development using a suitable OOP language and relevant software tools, e.g. C++, to include appropriate tools and effective programming techniques to optimize the execution
MO5	Test and document the complete object oriented application produced

Contact Hours	**Contact Hours**	
	Independent Study Hours:	
	Independent study/self-guided study	192
	Total Independent Study Hours:	192
	Scheduled Learning and Teaching Hours:	
	Face-to-face learning	108
	Total Scheduled Learning and Teaching Hours:	108
	Hours to be allocated	300
	Allocated Hours	300
Reading List	*The reading list for this module can be accessed via the following link:* https://uwe.rl.talis.com/index.html	