STUDENT AND ACADEMIC SERVICES

MODULE SPECIFICATION

Part 1: Information			
Module Title	Software Design and Development		
Module Code	UFCFPE-30-1	Level	Level 4
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	Applied Computing [Sep][PT][UCW][3yrs] FdSc 2018-19 Applied Computing [Sep][FT][UCW][2yrs] FdSc 2018-19		
Module type:	Standard		
Pre-requisites	None		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description
Overview: This module will be based on ensuring that students practical skills are developed in programming. Every session will incorporate designated practical work to complete in order to ensure that students understand and implement principles of good practice.   **Educational Aims:** On successful completion of this module students will be able to use C++ programming language to: (see Learning Outcomes).   **Outline Syllabus:** Design methodologies, e.g. pseudo code, step-wise refinement, structure diagrams and flow charts.   Basics of programming languages, e.g. procedural, event-driven, Introduction to object oriented techniques…etc.   Data storage: Files, variables, constants, literals, pre-defined and user defined data types,

program elements.

Software constructs: Sequence, selection and iteration.

Good programming practice, e.g. Modularisation, divide and conquer, use and re-use of modules, pre-defined and user defined functions and the attributes of a 'good' program.

Documentation requirements: Internal (e.g. variable names, comments) and external documentation (e.g. user guide).

Program testing: Types of error, test plans and testing methodologies.

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.

300 hours study time of which 108 hours will represent 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, which will be group sessions for academic content.

Independent learning includes hours engaged with essential reading, assignment and completion. Student study time will be organised each week with a series of both essential and further readings.

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.

Practical Assessment: Students will be given a specification from which they will produce two solutions using a taught methodology. Students will implement one of the solutions and justify their choice.

Fully Documented and Implemented System: A team assessment to replicate a workplace situation, in which a number of staff collaborate in a larger software development task. The task will include practical software design, development, testing, evaluation and supporting documentation of a business problem. This will be accompanied by a demonstration of the product by the teams of students. Each student will take responsibility for a section of the work which will be produced and assessed individually but marks will be allocated to the team for their audit trail documentation e.g minutes of meetings, peer evaluation.

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
Project - Component B		60 %	Fully documented (eg technical, user guide, algorithms) and implemented system)
Practical Skills Assessment - Component A	✓	40 %	Practical Assessment (in–class) (3 hours)

Resit Components	Final Assessment	Element weighting	Description
Project - Component B		60 %	Fully documented (eg technical, user guide, algorithms) and implemented system
Practical Skills Assessment - Component A	✓	40 %	Practical Assessment (in–class) (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	Demonstrate an understanding of a range of recognised design methodologies for solving a problem specification
MO2	Develop, implement and document a practical solution according to a design using structured programming methods
MO3	Apply techniques of good programming practice to optimise the practical solution implemented
MO4	Produce appropriate program documentation
MO5	Create a test plan incorporating a range of appropriate tests to confirm the validity of the designed practical solution
MO6	Evaluate the success of the application produced
MO7	Understand the basics of object oriented techniques, e.g. structures and terminology

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