MODULE SPECIFICATION

Part 1: Information			
Module Title	Computing Practice		
Module Code	UFMFSJ-15-1	Level	Level 4
For implementation from	2020-21		
UWE Credit Rating	15	ECTS Credit Rating	7.5
Faculty	Faculty of Environment & Technology	Field	Engineering, Design and Mathematics
Department	FET Dept of Engin Design & Mathematics		
Module type:	Project		
Pre-requisites	None		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description

Overview: This module concentrates on learning basic syntax and use of a structured, stepwise-refinement approach to design and implementation with exposure to structure charts and finite state diagrams.

Educational Aims: This module equips students with the theory and practice of basic syntax of ISO90 C to support a structured approach to program development using procedural abstractions - program and control structures - basic data types - reuse of basic functions for I/O, string and mathematical manipulation.

Outline Syllabus: Typical topics covered include:

Structured types - arrays, vectors and classes as records. Algorithm design. Simple file processing.

The use of functions and parameters. Global and local variables.

Bit-wise and logical operators.

Using pointers for accessing data and evoking functions.

Problem analysis and design using a structured, step-wise refinement approach.

Structure charts as effective documentary aids for HLL programs.

Teaching and Learning Methods: This module will typically involve 3 hours contact time peweek. The time will be more or less equally divided between lecture sessions, laboratory sessions.

The module will be taught with a very strong emphasis on practical work and the development of understanding by numerous demonstrations and simple, progressive exercises.

The course will concentrate on teaching basic syntax and use of a structured, stepwise-refinement approach to design and implementation with exposure to structure charts and finite state diagrams.

An extended case-study, supported by focussed laboratory based workshops, will allow the students to follow through an example application from design to implementation, and appreciate the relevance of all the component parts of the module syllabus.

Lectures will be used to introduce concepts, syntax and design methods. Laboratory sessions (workshops) will be used to practice and reinforce the students understanding of these. Students will be expected to work for an equivalent amount of their own time independently on the workshop material, and to independently read their reference book.

Scheduled learning includes lectures and workshops.
Independent learning includes hours engaged with essential reading, assignment preparation and completion etc.

Part 3: Assessment

Assessment will be by practical exercises This strategy has been chosen to develop the student's knowledge of the theory that they require in order to be successful at the coursework, thus reducing the impact of issues such as collusion.

The coursework will be in the form of a portfolio of work with several practical exercises to be completed throughout the module run. There will be multiple opportunities for formative feedback. All associated learning outcomes will be assessed. A significant percentage of the marks will be awarded for the students demonstrating and explaining their work.

The resit assessment will take the same format as the first sit assessment.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Portfolio - Component A	✓	100 %	The coursework will be in the form of a portfolio of work with several practical exercises to be completed throughout the module run.
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Portfolio - Component A	✓	100 %	The coursework will be in the form of a portfolio of work containing several practical exercises.

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will achieve the following learning outcomes:

Module Learning Outcomes	Reference
Generate correctly syntaxed code for programming in C	MO1
Demonstrate an understanding of finite state design	MO2
Apply structural design theory and use relevant approaches in the development of software programmes	MO3
Analyse and decipher code written in C	MO4

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/modules/ufcff6-30-1.html

Part 5: Contributes Towards

This module contributes towards the following programmes of study: