



## MODULE SPECIFICATION

Part 1: Information			
Module Title	Operating Systems and Architecture		
Module Code	UFCFCU-30-1	Level	Level 4
For implementation from	2020-21		
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		
Module type:	Standard		
Pre-requisites	None		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description
<p><b>Overview:</b> This module introduces students to the foundations of computer systems architecture together with the integrated hardware and software components and subsystems that enable and allow data to be input, processed and output. Low level programming is used to illustrate the operation of the components. The module then explores the concepts of operating systems, virtualisation, hardware management and file systems.</p> <p>Students will learn the basic concepts of how data is represented in digital systems, logic and storage components, machine organisation and assembler programming. They will then build on this knowledge to see how levels of abstraction are introduced via operating systems and virtualisation to produce usable systems.</p> <p><b>Educational Aims:</b> This module contributes to the provision of underpinning technical understanding.</p> <p><b>Outline Syllabus:</b> The module covers:</p> <ul style="list-style-type: none"> <li>classical computer architectures</li> <li>virtualised architectures</li> </ul>

## STUDENT AND ACADEMIC SERVICES

digital logic, static and dynamic digital systems  
 machine level representation of data  
 assembly level machine organisation;  
 memory system organisation and architecture  
 interfacing and communication  
 operating System principles  
 concurrency and synchronisation  
 scheduling and dispatch  
 memory management  
 file systems  
 I/O system

**Teaching and Learning Methods:** Lecture sessions cover the technical knowledge required. Designated practical work is included to ensure that apprentices have absorbed and understood the key principles involved.

This module will be based on ensuring that student's practical skills and knowledge gained in the block release sessions are carried into the workplace to inform their employment and generation of evidence of competency.

### Part 3: Assessment

Students grasp of fundamental operating systems programming is tested in a timed programming task. The emphasis is very much on being able to understand assembler code, rather than on demonstrating great fluency in writing the code. The students are therefore given an assembler program to analyse and describe its function. They must use it to explain the relationship between hardware components and programs.

In component B of assessment, the students will report on a practical task that they have carried out. During classroom sessions, the students will build a system consisting of a hypervisor, virtual client and virtual server. They will establish a client/server system and test functionality. They will write a report illustrating how their system has met the design requirement how it operates with reference to detailed operating system functions. The students report on their work in order to demonstrate their design thinking. The emphasis is on design in order to help them avoid a "build it and then fix it" mentality.

First Sit Components	Final Assessment	Element weighting	Description
Examination - Component A		50 %	2 hour unseen exam
Report - Component B	✓	50 %	3000 word project report.
Resit Components	Final Assessment	Element weighting	Description
Examination - Component A		50 %	2 hour exam
Report - Component B	✓	50 %	300 word report, reworked from the main sit.

### 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>
	Build test and debug a digital system to a specification.	MO1

## STUDENT AND ACADEMIC SERVICES

	Describe and explain computer architecture, digital logic and machine level representation of data.	MO2	
	Explain the relationships between hardware components and the subsystems used in a computer system.	MO3	
	Implement simple programs in assembler language.	MO4	
	Describe and explain the purposes and implementation of operating systems.	MO5	
Contact Hours	<b>Independent Study Hours:</b>		
	Independent study/self-guided study	135	
	<b>Total Independent Study Hours:</b>	135	
	<b>Placement Study Hours:</b>		
	Placement	75	
	<b>Total Placement Study Hours:</b>	75	
	<b>Scheduled Learning and Teaching Hours:</b>		
	Face-to-face learning	90	
	<b>Total Scheduled Learning and Teaching Hours:</b>	90	
	<b>Hours to be allocated</b>	300	
	<b>Allocated Hours</b>	300	
	Reading List	<p>The reading list for this module can be accessed via the following link:</p> <p><a href="http://readinglists.uwe.ac.uk/lists/67FFF5AB-7CA1-FE34-C946-CF1622C8AAE2.html">http://readinglists.uwe.ac.uk/lists/67FFF5AB-7CA1-FE34-C946-CF1622C8AAE2.html</a></p>	

### Part 5: Contributes Towards

This module contributes towards the following programmes of study:

BSc (Hons) Cyber Security Technical Professional (integrated degree) BSc (Hons) 2020-21