MODULE SPECIFICATION

Part 1: Information			
Module Title	Computational Thinking and Practice		
Module Code	UFCFQN-30-0	Level	Level 3
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	Audio and Music Technology {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Audio and Music Technology {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Software Engineering for Business {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Computing {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Forensic Computing and Security {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Forensic Computing and Security {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Computing {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Broadcast Audio and Music Technology {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Broadcast Audio and Music Technology {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Computer Science {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Computer Science {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Games Technology {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Business Computing {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Business Computing {Foundation} {Apprenticeship} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Digital Media {Foundation}[Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Digital Media {Foundation}[Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Software Engineering for Business {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Games Technology {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Business Computing {Foundation} [Feb][FT][GCET][4yrs] BSc (Hons) 2018-19		

	Business Computing {Foundation} [Oct][FT][GCET][4yrs] BSc (Hons) 2018-19
Module type:	Standard

Pre-requisites	None
Excluded Combinations	None
Co- requisites	None
Module Entry requirements	None

Part 2: Description

Educational Aims: In this module, you will acquire the skills necessary for identifying and modelling a problem into a computational solution. You will then practice on how to transform a computational solution into a computing program using a programming language.

Outline Syllabus: The module will cover:

What is computational thinking: abstract thinking, algorithmic thinking, logical thinking, scalable thinking

Computational thinking for everyone and everywhere

Process of problem solving:
-formulating a problem that could be solved through computation
-logically organizing and analysing data
-representing data through models and simulations
-generalizing to a wide variety of problems
-searching for the most efficient and effective solution

Design and implement software through algorithmic thinking
-approaches and methods for eliciting what is required by the user
-software design techniques
-programming language constructs and data types
-strategy, methods and techniques for developing testing

Teaching and Learning Methods: The module will be delivered via a combination of lectorial/workshop and lab sessions, with face-to-face and online help provided by tutors. Online resources such as Lynda.com are also available.

The first half of the module will focus upon computational thinking and the second half will focus more on programming. The transition from computational thinking to programming will be interleaved and iterated.

For the computational thinking part, there is an emphasis on group discussions and collaborations to re-enforce each student's understanding. In the lectorial/workshop sessions, students will be working in groups of 3 or 4 on the tasks set by the tutors. Students from different groups will be given opportunities to share their insights with the rest of the class. The assessment for this part will be group-based. Examples of tasks may be to consider the algorithmic aspects of work on a piece of music; or to examine an artwork from an aesthetic perspective and from the viewpoint of digitising it.

For the programming part, the learning is more focused on developing the individual student's ability to program. The assessment for the programming part will be individual in-class tests. In

the workshop/lab sessions, students will use their computational thinking skills to derive solutions and to implement the solutions using a programming language.

Part 3: Assessment

Assessment will comprise two components: a portfolio (Component B); and a series of programming tests under controlled conditions (Component A)

The assessment regime has been designed to ensure that students' understanding and skills are developed incrementally in a supported way, and that the students' experience is developed in a social and collaborative atmosphere, in accordance with the programme aims.

The portfolio in Component B will consist of outputs from group tasks. These tasks will be introduced to the students incrementally during the first half of the module in the workshops. Each group will produce an output for each task. The output must address how they have formulated and modelled problems, and what strategies were applied to derive the solution. Members from the same group will get the same mark for each output. However mark adjustment may take place where there is an evidence of significant unbalanced contributions from the group members.

Component A is used to assess the programming aspect. This is done as two in-class tests in a phased manner.

This phased approach will enable both the tutors and the students to evaluate the students' understanding after each phase. This form of assessment will provide opportunities for students to learn in an incremental way.

During each phase students will be given formative feedback leading to the test at the end of that phase. Feedback from the summative assessment will help to inform students' learning in the next phase.

For resit, component A will be an individual 2 hour programming test. Component B will be an individual task. The nature of the work will be same as in the main run. The size of the task will be scaled appropriately.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Portfolio - Component B		50 %	Portfolio of group outputs
In-class test - Component A		25 %	Individual in-class programming test (1 hour)
In-class test - Component A	✓	25 %	Individual in-class programming test (1 hour)
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Written Assignment - Component B		50 %	Individual coursework
Examination - Component A	✓	50 %	Individual programming test (2 hours)

Part 4: Teaching and Learning Methods

Learning Outcomes

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

	Module Learning Outcomes
MO1	Formulate and model problems in various task domains, identifying significant features and how to apply an appropriate strategy to finding solutions.
MO2	Translate solution to a problem into a computable form, and evaluate its effectiveness.
MO3	Explore methods to represent solutions as a formal process, such as instructions, algorithms or pseudocode.
MO4	Develop and evaluate ways of representing solutions as code, using basic programming constructs, data types and test methods.

Contact Hours

Contact Hours	
Independent Study Hours:	
Independent study/self-guided study	228
Total Independent Study Hours:	228
Scheduled Learning and Teaching Hours:	
Face-to-face learning	72
Total Scheduled Learning and Teaching Hours:	72
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/modules/ufcfqn-30-0.html