



MODULE SPECIFICATION

Part 1: Information			
Module Title	Design and Analysis of Data Structures and Algorithms		
Module Code	UFCFW4-30-2	Level	Level 5
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	Principles of Computing 2020-21		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description
<p>Educational Aims: See Learning Outcomes.</p> <p>Outline Syllabus: Review of methods of analysis:</p> <p>Mathematical modelling of complexity: space v time.</p> <p>Upper and lower bounds: techniques of analysis: correctness, efficiency; analytical strategies, order notation, design strategies.</p> <p>Hard problems: the classes P, NP and NP-complete; significance for computing.</p> <p>Classic Algorithms:</p> <p>Searching algorithms: linear search; jump searches; worst and average case upper bound determination - probabilistic approaches; randomised searches; binary search; using decision trees to determine the lower bound on search - worst and average cases.</p> <p>Selection algorithms: representing rankings using posets; finding maximum and second maximum values; lower bound considerations; finding maximum and minimum; finding i-th</p>

STUDENT AND ACADEMIC SERVICES

largest, randomised and non-randomised approaches.

Sorting algorithms: strategies for sorting; select sorts; insert sorts; merge sorts; split sorts; analysis of lower bounds and average case.

Numerical algorithms: exponentials; largest common factor; prime numbers; cryptography; fast Fourier transform.

Data Structures and the algorithms that support and maintain them:

Collections (sets, dictionaries, hash-tables).

General sequences (lists).

Stacks; queues and dequeues.

Trees (binary, multi-way and some variants such as 2-3-trees, B-trees, KD-Trees and Sphere Trees).

Graph algorithms: operations on structures; topological sort; depth- and breadth-first search; spanning trees; cheapest paths, travelling salesman problem.

Applications:

Use of language libraries or, where appropriate, handcrafted data structures in application problems such as, parsing arithmetic expressions, tracking the position of mobile units in computer games, providing predictive text and discrete event-based simulation.

Teaching and Learning Methods: This module will principally be delivered as an equal balance between lectures and practical sessions with some occasional tutorials and seminars. Students are expected to attend all scheduled classes. The lectures will explain theoretical concepts and students will be expected to read from the set text (or other directed readings) in preparation for the lectures.

The theory will be underpinned by practical sessions during which the students will write and experiment with programme code to illustrate and consolidate the concepts introduced by the lecture series.

The module will be supported by Blackboard which will be used as a repository for course materials, a forum for discussion and, from time to time, tests and/or quizzes to enable the students to self-test their knowledge.

Contact Hours:

This module will involve 6 hours contact time per fortnight. The time will be divided between lecture sessions and laboratory sessions:

Module contact time: 72 hours

Over the course of the academic year students should expect to spend approximately:

Activity:

Contact time: 72 hours

Assimilation and development of knowledge: 148 hours

Exam preparation: 40 hours

Coursework preparation: 40 hours

Total study time: 300 hours

STUDENT AND ACADEMIC SERVICES

Part 3: Assessment			
The module is assessed by a three hour exam which will be taken at the end of the course. In addition, students will complete a piece of coursework. The coursework is designed to test the students' capacity to implement the ideas presented in software and will require them to write a number of programmes in an object oriented language.			
Students should expect to spend approximately 40 hours completing the coursework.			
First Sit Components	Final Assessment	Element weighting	Description
Practical Skills Assessment - Component B		50 %	Programming exercise(s) requiring the implementation of some of the key module concepts
Examination (Online) - Component A	✓	50 %	Online Exam (3 hours) 24 hour window
Resit Components	Final Assessment	Element weighting	Description
Practical Skills Assessment - Component B		50 %	Programming exercise(s) requiring the implementation of some of the key module concepts
Examination (Online) - Component A	✓	50 %	Online Exam (3 hours) 24-hour window

Part 4: Teaching and Learning Methods															
Learning Outcomes	<p>On successful completion of this module students will achieve the following learning outcomes:</p> <table border="1"> <thead> <tr> <th>Module Learning Outcomes</th> <th>Reference</th> </tr> </thead> <tbody> <tr> <td>Understand and use algorithm design strategies</td> <td>MO1</td> </tr> <tr> <td>Design programmes that use appropriate data structures</td> <td>MO2</td> </tr> <tr> <td>Implement data structures and the algorithms that maintain them</td> <td>MO3</td> </tr> <tr> <td>Use mathematical techniques in the analysis of algorithms for both correctness and efficiency</td> <td>MO4</td> </tr> <tr> <td>Critically compare and evaluate algorithms with respect to the appropriate problem domain</td> <td>MO5</td> </tr> <tr> <td>Analyse requirements and select appropriate solutions</td> <td>MO6</td> </tr> </tbody> </table>	Module Learning Outcomes	Reference	Understand and use algorithm design strategies	MO1	Design programmes that use appropriate data structures	MO2	Implement data structures and the algorithms that maintain them	MO3	Use mathematical techniques in the analysis of algorithms for both correctness and efficiency	MO4	Critically compare and evaluate algorithms with respect to the appropriate problem domain	MO5	Analyse requirements and select appropriate solutions	MO6
Module Learning Outcomes	Reference														
Understand and use algorithm design strategies	MO1														
Design programmes that use appropriate data structures	MO2														
Implement data structures and the algorithms that maintain them	MO3														
Use mathematical techniques in the analysis of algorithms for both correctness and efficiency	MO4														
Critically compare and evaluate algorithms with respect to the appropriate problem domain	MO5														
Analyse requirements and select appropriate solutions	MO6														
Contact Hours	<table border="1"> <thead> <tr> <th colspan="2">Independent Study Hours:</th> </tr> </thead> <tbody> <tr> <td>Independent study/self-guided study</td> <td>228</td> </tr> <tr> <td>Total Independent Study Hours:</td> <td>228</td> </tr> <tr> <th colspan="2">Scheduled Learning and Teaching Hours:</th> </tr> <tr> <td>Face-to-face learning</td> <td>72</td> </tr> </tbody> </table>	Independent Study Hours:		Independent study/self-guided study	228	Total Independent Study Hours:	228	Scheduled Learning and Teaching Hours:		Face-to-face learning	72				
Independent Study Hours:															
Independent study/self-guided study	228														
Total Independent Study Hours:	228														
Scheduled Learning and Teaching Hours:															
Face-to-face learning	72														

STUDENT AND ACADEMIC SERVICES

	Total Scheduled Learning and Teaching Hours:	72
	Hours to be allocated	300
	Allocated Hours	300
Reading List	<p><i>The reading list for this module can be accessed via the following link:</i></p> <p>https://uwe.rl.talis.com/modules/ufcfw4-30-2.html</p>	

Part 5: Contributes Towards	
<p>This module contributes towards the following programmes of study:</p> <p>Software Engineering for Business [Sep][FT][Frenchay][3yrs] BSc (Hons) 2019-20</p> <p>Software Engineering for Business [Sep][SW][Frenchay][4yrs] BSc (Hons) 2019-20</p> <p>Computer Science {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19</p> <p>Computer Science {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19</p> <p>Software Engineering for Business {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19</p> <p>Software Engineering for Business {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19</p>	