STUDENT AND ACADEMIC SERVICES

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	2019-20		
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	Introduction to OO Systems Development 2018-19, Principles of Computing 2018-19		
Excluded Combinations	None		
Co-requisites	None		
Module Entry Requirements	None		
PSRB Requirements	None		

Part 2: Description

Educational Aims: See Learning Outcomes.

Outline Syllabus: Review of methods of analysis:

Mathematical modelling of complexity: space v time.

Upper and lower bounds: techniques of analysis: correctness, efficiency; analytical strategies, order notation, design strategies.

Hard problems: the classes P, NP and NP-complete; significance for computing.

Classic Algorithms:

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.

Selection algorithms: representing rankings using posets; finding maximum and second maximum values; lower bound considerations; finding maximum and minimum; finding i-th 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 deques.

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.

Data protection legislation and the GDPR.

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.

Developing algorithms for secure data processing; reactive and proactive data integrity.

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

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**
Examination - Component A	✓	50 %	Exam (3 hours)
Practical Skills Assessment - Component B		50 %	Programming exercise(s) requiring the implementation of some of the key module concepts
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Examination - Component A	✓	50 %	Exam (3 hours)
Practical Skills Assessment - Component B		50 %	Programming exercise(s) requiring the implementation of some of the key module concepts

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**
Understand and use algorithm design strategies	MO1
Design programmes that use appropriate data structures	MO2
Implement data structures and the algorithms that maintain them, allowing for secure processing of the data	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
Understand the legal framework for protecting data and its implications on developing relevant software	MO7

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/ufcfw4-30-2.html	

Part 5: Contributes Towards

This module contributes towards the following programmes of study:

Computer Science [Sep][SW][Frenchay][4yrs] BSc (Hons) 2018-19

Software Engineering for Business [Sep][SW][Frenchay][4yrs] BSc (Hons) 2018-19

Computer Science [May][FT][Villa][3yrs] BSc (Hons) 2018-19

Computer Science [Jan][FT][Villa][3yrs] BSc (Hons) 2018-19

Computer Science [Sep][FT][Villa][3yrs] BSc (Hons) 2018-19

Computer Science [Sep][FT][Frenchay][3yrs] BSc (Hons) 2018-19

Software Engineering for Business [Sep][FT][Frenchay][3yrs] BSc (Hons) 2018-19