Module Specification

Design and Analysis of Data Structures and Algorithms

Version: 2023-24, v4.0, 17 Mar 2023

Contents

Part 1: Information

Module title: Design and Analysis of Data Structures and Algorithms

Module code: UFCFW4-30-2

Level: Level 5

For implementation from: 2023-24

UWE credit rating: 30

ECTS credit rating: 15

Faculty: Faculty of Environment & Technology

Department: FET Dept of Computer Sci & Creative Tech

Partner institutions: None

Delivery locations: Not in use for Modules

Field: Computer Science and Creative Technologies

Module type: Module

Pre-requisites: Introduction to OO Systems Development 2022-23, Principles of Computing 2022-23

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: Not applicable

Features: Not applicable

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.

Part 3: Teaching and learning methods

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

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

MO1 Understand and use algorithm design strategies

MO2 Design programmes that use appropriate data structures

MO3 Implement data structures and the algorithms that maintain them, allowing for secure processing of the data

MO4 Use mathematical techniques in the analysis of algorithms for both correctness and efficiency

MO5 Critically compare and evaluate algorithms with respect to the appropriate problem domain

MO6 Analyse requirements and select appropriate solutions

MO7 Understand the legal framework for protecting data and its implications on developing relevant software

Hours to be allocated: 300

Contact hours:

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Total = 300

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://uwe.rl.talis.com/modules/ufcfw4-30-2.html

Part 4: Assessment

Assessment strategy: Assessment is via a piece of coursework. The coursework is designed to asses 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. This assessment will also require the students to consider the legal framework for protecting data and its implications for developers.

Assessment components:

Practical Skills Assessment (First Sit)
Description: Programming exercise(s) requiring the implementation of some of the key module concepts
Weighting: 100 %
Final assessment: No
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5, MO6, MO7

Practical Skills Assessment (Resit)
Description: Programming exercise(s) requiring the implementation of some of the key module concepts
Weighting: 100 %
Final assessment: No
Group work: No
Learning outcomes tested:

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Computer Science [Sep][FT][Villa][3yrs] - Not Running BSc (Hons) 2022-23

Computer Science [May][FT][Villa][3yrs] - Not Running BSc (Hons) 2022-23

Computer Science [Jan][FT][Villa][3yrs] - Not Running BSc (Hons) 2022-23

Software Engineering for Business [Frenchay] BSc (Hons) 2022-23

Software Engineering for Business {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2021-22

Software Engineering for Business {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2021-22