Module Details	
Module Title	Data Structures and Algorithms
Module Code	COS5021-B
Academic Year	2024/5
Credits	20
School	School of Computer Science, AI and Electronics
FHEQ Level	FHEQ Level 5

Contact Hours	
Type	Hours
Lectures	24
Laboratories	24
Directed Study	152

Availability	
Occurrence	Location / Period
BDA	University of Bradford / Semester 1

Module Aims

To introduce computational thinking for problem solving as core computer science knowledge that all computing practitioners are expected to know and use: to explore computational aspects of data representation, data-based optimisation, theoretical analysis and to develop familiarity with well-established and proven computing solutions to problems that occur frequently throughout computing and practical computing applications.

Outline Syllabus

Pseudocode. Estimating algorithm running time: complexity analysis; asymptotic order notation. Abstract Data Types. ADTs, implementations and running times for linear structures (stacks, queues, lists, hash tables), binary trees (ordered, balanced, complete), multi-way trees, graphs. Searching and sorting methods.

Learning Outcomes

Outcome Number	Description
O1	Recognise common data structures and fundamental algorithms and be familiar with the associated terminology.
O2	Define Abstract Data Types in terms of their data structures.
O3	Identify algorithms and demonstrate their operation and effect in solving problems.
O4	Assess efficiency of various implementations and justify choice of implementation for any application.
O5	Construct and test simple algorithms using suitable data structures, analyse the complexity and running times for algorithms.

Learning, Teaching and Assessment Strategy

As fundamental part of Computer Science, Data Structures and Algorithms are introduced for each topic by a set of exercises designed to build knowledge, promote understanding and develop problem-solving skill. For each topic lectures expose concepts and introduce material and terminology in context. All the learning materials have been shared on Canvas. In particular, the basic linear data structure will be introduced in the first 2 weeks and tree structure will be given in weeks 3-8 while graphs will be discussed in weeks 9-12. The feedback session has been achieved by questionaries and the recommend reading list is given based on the pre-lecture session. Lab design is implemented using Java and Python, while the online visualisation tool is introduced to help the student for concept understanding. In addition, to support the students? lab remotely, we also have a parallel Canvas conference session if the student has urgent enquiry for the module delivery.

Assessment is aligned to the learning outcomes. In particular, the coursework 1 with computing exercises (min-heap, AVL tree, Harsh table, etc.) will reflect the learning outcomes: LO1 and LO2 by programming. The concept understanding and complexity analysis will be reflected using coursework 2 (Graph, sorting algorithm, multi-way tree, etc.) which covers all the rest LOs including LO3, LO4 and LO5. The coursework has been designed using the key values, which are generated by the students? UB number. In other words, all the students will have individual coursework with the unique solution.

Mode of Assessment

Type	Method	Description	Weighting
Summative	Coursework - Written	Course work (6 Computing exercises)	50%
Summative	Coursework - Written	Report (4 Computing exercises)	50%
Formative		Following the given example on the slides, the student would have random classroom-based test with some simply exercises in order to better understand the concepts	N/A

Reading List

To access the reading list for this module, please visit https://bradford.rl.talis.com/index.html