Prifysgol Wrecsam
Wrexham University

Module specification

When printed this becomes an uncontrolled document. Please access the Module Directory for the most up to date version by clicking on the following link: Module directory

Module Code	COM571
Module Title	Data Structures and Algorithms
Level	5
Credit value	20
Faculty	FACE
HECoS Code	100956
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
BSc (Hons) Computer Science	Core
BSc (Hons) Computer Science with industrial placement	Core
Delivery as standalone or part of CPD package	Option
BSc (Hons) Software Engineering	Core
BSc (Hons) Software Engineering with Industrial Placement	Core

Pre-requisites

None

Breakdown of module hours

Learning and teaching hours	15 hrs
Placement tutor support	0 hrs
Supervised learning e.g. practical classes, workshops	15 hrs
Project supervision (level 6 projects and dissertation modules only)	0 hrs
Total active learning and teaching hours	**30** hrs
Placement / work based learning	0 hrs

Learning and teaching hours	15 hrs
Guided independent study	170 hrs
Module duration (total hours)	**200** hrs

For office use only	
Initial approval date	08/11/2023
With effect from date	Sept 2025
Date and details of revision	07/05/2025 addition of BSc (Hons) Software Engineering programme titles from Sept 25
Version number	2

Module aims

This module will introduce students to the concepts and principles of programming data structures and algorithms. The module focus is developing a comprehensive understanding of commonly used data structures such as arrays, linked lists, stacks, queues, trees, graphs, and hash tables. Gain knowledge of various algorithmic techniques including sorting, searching, graph traversal, and dynamic programming. Compare and contrast different algorithms to select the most appropriate solution for a given problem, considering efficiency and resource utilization. Enhance analytical thinking abilities by breaking down complex problems into smaller sub-problems and designing algorithmic solutions

Module Learning Outcomes - at the end of this module, students will be able to:

1	Demonstrate proficiency in the implementation, manipulation, and optimization of complex data structures.
2	Apply various algorithms and techniques for sorting and searching.
3	Analyse and evaluate the time and space complexity of algorithms, including Big O notation.
4	Use critical thinking and analytical skills through the analysis and comparison of different algorithms and data structures.

Assessment

Indicative Assessment Tasks:
This section outlines the type of assessment task the student will be expected to complete as part of the module. More details will be made available in the relevant academic year module handbook.

The assessment will consist of multiple tasks, which include analysing, designing and implementing solutions to programming problems. Students will work individually on the assessment problems, which could include using data structures such as linked lists and

binary trees. Also, students may be required to analyse the properties, strengths, weaknesses, and appropriate use cases of different data structures.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1, 2, 3, 4	Coursework	100%

Derogations

None

Learning and Teaching Strategies

In line with the Active Learning Framework, this module will be blended digitally with both a VLE and online community. Content will be available for students to access synchronously and asynchronously and may indicatively include first and third-party tutorials and videos, supporting files, online activities any additional content that supports their learning.
As this module progresses, the strategies will change to best support a diverse learning environment. Initially, the module will start with a heavier reliance on engaging tutor-led lectures, demonstrations, and workshops to ensure that the students get the relevant threshold concepts. As the module continues experiential and peer learning strategies will be encouraged as the students' progress with their portfolio work.
Assessment will occur throughout the module to build student confidence and self-efficacy in relation to applied mathematical and technical programming concepts.

Indicative Syllabus Outline

Yearly content will be updated to represent the most appropriate content for current industry technologies, but a list of indicative topics could include:

- Arrays and Linked Lists
- Stacks and Queues
- Trees
- Graphs
- Sorting and Searching Algorithms
- Algorithm Analysis
- Hash Tables
- Advanced Data Structures
- Greedy Algorithms
- Divide and Conquer Algorithms
- Advanced Topics in Data Structures
- String Algorithms
- Algorithm Design
- Big O notation and its applications

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update.

Essential Reads

J. Wengrow, *A Common-Sense Guide to Data Structures and Algorithms, Second Edition: Level Up Your Core Programming Skills,* Pragmatic Bookshelf, 2020.

Other indicative reading

N. Karumanchi, *Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles* 5th ed, CareerMonk Publications, 2016.
K. D. Lee, *Data Structures and Algorithms with Python (Undergraduate Topics in Computer Science)*, Springer, 2015.