

## Liverpool John Moores University

Title: Data Structures and Algorithms  
Status: Definitive  
Code: **5217COMP** (127993)  
Version Start Date: 01-08-2021

Owning School/Faculty: Computer Science and Mathematics  
Teaching School/Faculty: Computer Science and Mathematics

Team	Leader
David Lamb	Y
Glyn Hughes	

**Academic Level:** FHEQ5      **Credit Value:** 20      **Total Delivered Hours:** 46  
**Total Learning Hours:** 200      **Private Study:** 154

### Delivery Options

Course typically offered: Semester 1

Component	Contact Hours
Workshop	44

**Grading Basis:** 40 %

### Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Artefacts	AS1	Design, implementation, and evaluation of software versions	40	
Exam	AS2	Examination	60	2

### Aims

*To introduce the student to the fundamentals of Abstract Data Types (ADTs) and complexity of operations on ADTs followed by an implementation-based exploration of common data structures and operations, their implementations and applications. To expose students to the development and optimisation of software based on both theoretical and applied evaluations.*

## Learning Outcomes

After completing the module the student should be able to:

- 1 Explain a range of fundamental data structures and their operations
- 2 Analyse the complexity of various algorithms' as applied to ADTs
- 3 Implement standard ADTs using both primitive language constructs and extant resources, and evaluate their performance for a specified problem domain
- 4 Synthesise algorithms and data structures to fulfil a problem specification

## Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Software versions	3	4
Examination	1	2

## Outline Syllabus

*Abstract Data Types and common implementation strategies:*

*Linear ADTs: Lists (Arrays, Linked Lists) Stacks, Queues*

*Non-Linear ADTs: Trees, Binary Trees, BSTs*

*Maps (ListMaps, BSTMaps, HashMaps)*

*Algorithms for structural operations; insert, remove, retrieval*

*Algorithms for structure navigation; traversal, search*

*Sorting Algorithms*

*Algorithm types: iterative and recursive*

*Relationship between ADTs and computing fundamentals (e.g. Stack, Queue)*

*Use of Big O notation to specify time complexity for simple algorithms*

*Use of a program debugger to inspect the call stack and stack frames*

*Use of software instrumentation to measure performance and inspect root/causal method calls*

## Learning Activities

Didactic, lecture-based theory and applied examples.

Workshop activities exploring the implementation and usage of core structures and algorithms.

Problem-based learning centred on coursework assignment tasks.

## Notes

This module is a technical, skills-focused module. It is an applied study of Data Structures and Algorithms; as such will require previous experience in programming. It will build on existing programming-based skills such as problem / functional decomposition and the use of an IDE to develop and test programs. Basic

operational familiarity with a debugger will be assumed but reinforced and built on during this module.