

# Data Structures and Algorithms

## Module Information

2022.01, Approved

### Summary Information

Module Code	5133COMP
Formal Module Title	Data Structures and Algorithms
Owning School	Computer Science and Mathematics
Career	Undergraduate
Credits	20
Academic level	FHEQ Level 5
Grading Schema	40

### Teaching Responsibility

LJMU Schools involved in Delivery
Computer Science and Mathematics

### Learning Methods

Learning Method Type	Hours
Workshop	55

### Module Offering(s)

Display Name	Location	Start Month	Duration Number Duration Unit
SEP-CTY	CTY	September	12 Weeks

### Aims and Outcomes

Aims	This is a practical, applied Software Engineering module with the aim of introducing the student to the fundamentals of Abstract Data Types and complexity of operations on ADTs followed by an implementation-based exploration of common data structures and operations, their implementations and applications
------	---

**After completing the module the student should be able to:**

## Learning Outcomes

Code	Number	Description
MLO1	1	Explain a range of fundamental data structures and their operations
MLO2	2	Analyse fundamental algorithms' complexity as applied to a range of ADT implementations
MLO3	3	Evaluate data structures in a given problem domain
MLO4	4	Implement standard ADTs using both primitive language and library resources
MLO5	5	Synthesise appropriate algorithms and data structures to fulfil a problem specification

## Module Content

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) Graphs Algorithms for structure operations; insert, remove, retrieval Algorithms for structure navigation; search and sort 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 Using a program debugger to monitor program state, and halt/control execution as required. Use of a program debugger to inspect the call stack and stack frames
Module Overview	
Additional Information	This module is a technical, skills-focused module. It 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.

## Assessments

Assignment Category	Assessment Name	Weight	Exam/Test Length (hours)	Module Learning Outcome Mapping
Portfolio	Implementation of software	60	0	MLO3, MLO4, MLO5
Report	Report	40	0	MLO1, MLO2

## Module Contacts

### Module Leader

Contact Name	Applies to all offerings	Offerings
Sorren Hanvey	Yes	N/A

### Partner Module Team

Contact Name	Applies to all offerings	Offerings