

Module Title	Data Structures and Algorithms
Level	4
Reference No.	CSI_4_DSA
Credits	20
Student Study Hours	Total: 200 Contact hours: 52 Student managed learning hours: 148
Pre-Requisites	Software Development 1: Fundamentals of Software Development
Co-requisites	None
Excluded combinations	None
Module coordinator	TBC
Division	Division of Computer Science and Informatics
Short Description	This module teaches the definition of data structures, attributes, arrays, records, linked lists, binary trees and hash tables, using the fundamental elements of programming languages to construct them (for example using pointers). Also, the derivation of algorithms, problem solving techniques, sequences, selections, and repetitions, sorting and searching. The relationship between abstract data types and object-oriented classes is introduced and the ready-made implementation of common structures such as hash tables in software libraries is explored.
Aims	To build upon the fundamental knowledge acquired in the previous software development module to develop problem analysis and solving skills.
Learning Outcomes	LO1: Knowledge and Understanding - Describe different algorithms and the means used to measure their performance. (Maps to: BCS 2.2.1 a1-a4, LO2: Intellectual Skills - Analyse programming problems and identify appropriate algorithmic solutions. (Maps to: BCS 2.2.1 a5-a9) LO3: Practical Skills - Develop software to solve relatively complex problems and assess alternative solutions. (Maps to: BCS 2.2.1 b1-b4; 2.2.3 a4-a6) LO4: Transferable Skills - Apply critical and analytical reasoning to tasks. (Maps to: BCS 2.2.1 c1, c2)
Employability	Software development skills are in demand in themselves and understanding software development is an important skill in many non-development employment roles as it allows critical appraisal of solutions offered by others.
Teaching and Learning Pattern	This module emphasises learning through practical exercises and the development of actual software artefacts. Short lectures will be used to inform the laboratory activities and provide a forum for discussion of issues students have encountered in the practical work. The lab sessions will occupy the majority of the contact time and will involve much independent working. Students are required to keep a clear record of the work they have done and are encouraged to experiment and investigate beyond the basic material being taught.

Indicative Content	• Data Structures composed of primitive types • Records, Collections, Linked Lists, Binary Trees, Hash Tables • Sequences, selections, repetitions, sorting and searching • Iterative and recursive algorithms • Big O notation and measurement of algorithm performance • The implementation of data structures and collections in object-oriented languages
Assessment Elements & weightings	COURSEWORK 100% Summative Assessment Coursework: Expected to consist of individual assessments typically lab-based development exercises and analytical reports including code development. (LO1 - LO4) (Maps to: BCS 2.2.1 a1-a9, b1-b4, c1-c2; 2.2.3 a4-a6) Formative Assessment Skills for the summative assessment will be embedded throughout formative opportunities in Lectures and Workshops. Formative assessment will take different forms, such as: • a logbook used to aid discussions with the tutor • verbal feedback on tutorial activities • observation and questioning to provide instant feedback as the student takes part in learning activities
Indicative Sources (Reading lists)	Core: There is no core textbook defined for this module. Students are expected to refer to the indicative sources below. Optional: • Samoylov, N., (2018) <i>Introduction to programming: learn to program in Java data structures algorithms and logic</i>. ISBN 1788839129 • Goodrich, M.T., Tamassia, R. (2015) <i>Data Structures and Algorithms in Java</i>. John Wiley & Sons; ISBN 1118808576 • Weiss, M. (2013) <i>Data Structures and Algorithm Analysis in C++</i>. Pearson; ISBN 0273769383