H8DS: Data Structures

Module Code:	H8DS
Long Title	Data Structures APPROVED
Title	Data Structures
Module Level:	LEVEL 8
EQF Level:	6
EHEA Level:	First Cycle
Credits:	10
Module Coordinator:	
Module Author:	Isabel O'Connor
Departments:	School of Computing
Specifications of the qualifications and experience required of staff	Master's and/or PhD degree in computing or cognate discipline. May have industry experience also.

Learning Outcomes

On successful completion of this module the learner will be able to:

#	Learning Outcome Description
LO1	Explain the theory, concepts and principles of linear and non-linear data structures used in computer science.
LO2	Use object-oriented concepts and generics to develop abstract data types and algorithms.
LO3	Identify, evaluate and implement software solutions that use complex data structures and related algorithms to solve a number of fundamental real-world computational problems.

Dependencies

Module Recommendations

No recommendations listed

Co-requisite Modules

No Co-requisite modules listed

Entry requirements	See section 4.2 Entry procedures and criteria for the programme including procedures recognition of prior learning

H8DS: Data Structures

Module Content & Assessment

Indicative Content
Introduction Review of Software Testing; writing a unit test; unit testing will be used throughout the module, where appropriate, when implementing data structures and their corresponding operations.. Introduction to Data Structures (e.g. linear and non-linear)
Recursion The concept of recursion. Fibonacci sequence. Example of real-life problems using recursion
Linear Data Structures ArrayLists. Stacks. Queues. Priority Queues. Single Linked Lists. Double Linked Lists. Operations on linear structures
Binary Trees Tree Data Structure – characteristics. Tree organization. Tree traversal. Balancing a tree
Search Trees Tree search characteristics and applicability. Binary search trees. Balanced binary search. Implementing a binary search tree
Graphs Representation of graphs. Types of graphs (e.g. undirected, directed, and weighted graphs). Graph traversals algorithms (i.e. depth- and breadth-first traversals). Common graph algorithms – shortest-path algorithm (e.g. Dijkstra's algorithm)

Assessment Breakdown	%
Coursework	50.00%
End of Module Assessment	50.00%

Assessments

Full Time

Coursework

Assessment Type:	Formative Assessment	**% of total:**	Non-Marked
Assessment Date:	n/a	**Outcome addressed:**	1,2,3
Non-Marked:	Yes		

Assessment Description:
The formative assessment will consist of ongoing independent and group programming practical tasks. In-class discussions and oral feedback will be provided throughout these activities.

Assessment Type:	Continuous Assessment	**% of total:**	50
Assessment Date:	n/a	**Outcome addressed:**	1,2,3
Non-Marked:	No		

Assessment Description:
The continuous assessment will consist of in-class practical tests. The practical assessments aim to evaluate the students' knowledge and ability to identify, evaluate, implement and use data structures and algorithms to solve computational problems. Students will be assessed both on their development skills and their ability to convey understanding of the data structures and algorithms they have developed.

End of Module Assessment

Assessment Type:	Terminal Exam	**% of total:**	50
Assessment Date:	End-of-Semester	**Outcome addressed:**	1,2,3
Non-Marked:	No		

Assessment Description:
End-of-Semester Final Examination

No Workplace Assessment

Reassessment Requirement

Repeat examination
Reassessment of this module will consist of a repeat examination. It is possible that there will also be a requirement to be reassessed in a coursework element.

Reassessment Description
Repeat examinationReassessment of this module will be via repeat examination which evaluates all learning outcomes.

H8DS: Data Structures

Module Workload

Module Target Workload Hours 0 Hours

Workload: Full Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Classroom & Demonstrations (hours)	24	Per Semester	2.00
Tutorial	Other hours (Practical/Tutorial)	36	Per Semester	3.00
Independent Learning	Independent learning (hours)	190	Per Semester	15.83
			Total Weekly Contact Hours	5.00

Module Resources

Recommended Book Resources

Michael T. Goodrich,Roberto Tamassia,Michael H. Goldwasser. (2014), Data Structures and Algorithms in Java, John Wiley & Sons, p.736, [ISBN: 1118771338].

Mark Allen Weiss. (2010), Data Structures & Problem Solving Using Java, Addison-Wesley Longman, p.985, [ISBN: 0321541405].

Supplementary Book Resources

Donald Ervin Knuth. (1997), The Art of Computer Programming: Fundamental algorithms, Addison-Wesley Professional, p.650, [ISBN: 0201896834].

Paul Deitel,Harvey Deitel. (2017), Java How to Program, Early Objects, Pearson, p.1296, [ISBN: 9780134743356].

This module does not have any article/paper resources

This module does not have any other resources

Discussion Note:	