



Title:	Linear Data Struct. & Alg. APPROVED
Long Title:	Linear Data Struct. & Alg.
Module Code:	COMP7035
Duration:	1 Semester
Credits:	5
NFQ Level:	Intermediate
Field of Study:	Computer Science
Valid From:	Semester 1 - 2017/18 (September 2017)
Module Delivered in	4 programme(s)
Module Coordinator:	Sean McSweeney
Module Author:	Ignacio Castineiras
Module Description:	Data structures and algorithms are fundamental elements in many computing applications. In computer programs data structures offer different techniques for storing data while algorithms provide the methods for manipulating this data. In this module the learner will be introduced to the application of algorithms and data structures to effectively tackle information representation and manipulation when solving a computer science-related problem. The module will examine and assess divide & conquer and greedy algorithms using linear-based abstract data types.
Learning Outcomes	
<i>On successful completion of this module the learner will be able to:</i>	
LO1	Assess the role of an abstract data type in isolating the data usage from its internal representation.
LO2	Compare and contrast the interfaces and internal representation of a number of linear abstract data types.
LO3	Design and specify the operations of a linear-based abstract data type in a declarative manner and implement them in a high-level programming language.
LO4	Assess the applicability of divide and conquer algorithms and greedy algorithms to real-world problems.
LO5	Design and implement divide and conquer and greedy algorithms and compare their formulations and solutions.
Pre-requisite learning	
Module Recommendations	
<i>This is prior learning (or a practical skill) that is strongly recommended before enrolment in this module. You may enrol in this module if you have not acquired the recommended learning but you will have considerable difficulty in passing (i.e. achieving the learning outcomes of) the module. While the prior learning is expressed as named CIT module(s) it also allows for learning (in another module or modules) which is equivalent to the learning specified in the named module(s).</i>	
Incompatible Modules	
<i>These are modules which have learning outcomes that are too similar to the learning outcomes of this module. You may not earn additional credit for the same learning and therefore you may not enrol in this module if you have successfully completed any modules in the incompatible list.</i>	
No incompatible modules listed	
Co-requisite Modules	
No Co-requisite modules listed	
Requirements	
<i>This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed. You may not enrol on this module if you have not acquired the learning specified in this section.</i>	
No requirements listed	

Module Content & Assessment

Indicative Content

Data Structures and Algorithms.

Computational model for solving a real-world problems. Data types and data structures. Representing information from a problem domain. Algorithm specification and implementation. Manipulation of the information to solve problems.

Abstract Data type (ADT).

Datatype: Values, operations and internal representation. Data type: Values, operations and internal representation. ADT: Separation of data usage from its internal representation. Creator, observer and mutator operations. Partial and total properties. Generics: Separation of data construction from its underlying elements.

Linear-based ADTs.

Brief declarative semantics of linear-based ADTs. ADT list. Specification: Minimal set of operations. Representation: Static (array-based) vs. dynamic (node-based) implementations. Extending the interface: Supplementary operations. Addition linear-based ADTs: Stacks and queues.

Divide and Conquer Algorithms.

Direct solution vs. decomposition solutions. Recursion as a problem solving technique. Applications: Sorting and searching problems.

Greedy Algorithms.

Selected and discarded candidates. Selection, satisfiability and solution functions. Applications: Resource allocation and scheduling problems.

Assessment Breakdown	%
Course Work	50.00%
End of Module Formal Examination	50.00%

Course Work				
Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Project	Define, specify and document the set of operations for a novel ADT. Implement the set of operations of the ADT using an internal representation based on static and dynamic linear data structures. Produce a report to justify the set of operations and the concrete data structures chosen for the internal representation.	1,2,3	25.0	Week 6
Project	Design, implement and document a divide and conquer or a greedy algorithm to tackle some real-life problems. Produce a report to justify the algorithm family being selected in terms of how effective it is to model the problem domain.	1,4,5	25.0	Week 11

End of Module Formal Examination				
Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Formal Exam	End of Semester Formal Examination.	1,2,3,4,5	50.0	End-of-Semester

Reassessment Requirement
<p>Coursework Only This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.</p>

The institute reserves the right to alter the nature and timings of assessment

Module Workload

Workload: Full Time				
<i>Workload Type</i>	<i>Workload Description</i>	<i>Hours</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecture	Lecture delivering theory underpinning learning outcomes.	2.0	Every Week	2.00
Lab	Practical computer-based lab supporting learning outcomes.	2.0	Every Week	2.00
Independent Learning	Independent Study.	3.0	Every Week	3.00
Total Hours				7.00
Total Weekly Learner Workload				7.00
Total Weekly Contact Hours				4.00

Workload: Part Time				
<i>Workload Type</i>	<i>Workload Description</i>	<i>Hours</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecture	Lecture delivering theory underpinning learning outcomes.	2.0	Every Week	2.00
Lab	Practical computer-based lab supporting learning outcomes.	2.0	Every Week	2.00
Independent Learning	Independent Study.	3.0	Every Week	3.00
Total Hours				7.00
Total Weekly Learner Workload				7.00
Total Weekly Contact Hours				4.00

Module Resources

Recommended Book Resources

- Narashimha Karumanchi 2016, *Data Structures And Algorithms Made Easy*, CareerMonk [ISBN: 9788193245279]
- Richard Neapolitan 2014, *Foundations of Algorithms*, 5th Ed., Jones and Bartlett Publishers [ISBN: 9781284049190]

Supplementary Book Resources

- Thomas H. Cormen et. al. 2009, *Introduction to Algorithms*, 3rd Ed., MIT Press [ISBN: 9780262033848]
- Christopher Steiner 2012, *Automate This: How Algorithms Came to Rule Our World*, Penguin [ISBN: 9781591844921]
- John V. Guttag 2013, *Introduction to Computation and Programming Using Python*, MIT Press [ISBN: 9780262525008]
- Michael T. Goodrich et. al. 2014, *Data Structures and Algorithms in Java*, Wiley Publishing [ISBN: 9781118771334]
- Michael T. Goodrich et. al. 2013, *Data Structures and Algorithms in Python*, Wiley Publishing [ISBN: 9781118290279]

This module does not have any article/paper resources

Other Resources

- Website: *Learn to think as a Computer Scientist*
<https://www.coursera.org/specializations/algorithms>
- Website: *Data Structure Visualizations*
<https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
- Website: *CodinGame - Practice coding with fun programming challenges*
<https://www.codingame.com/>
- Website: *Python documentation*
<http://www.python.org/doc/>
- Website: *Java documentation*
<http://www.oracle.com/technetwork/java/javase/documentation/index.html>

Module Delivered in

Programme Code	Programme	Semester	Delivery
CR_KSDEV_8	<u>Bachelor of Science (Honours) in Software Development</u>	3	Mandatory
CR_KDNET_8	<u>Bachelor of Science (Honours) in Computer Systems</u>	3	Mandatory
CR_KCOMP_7	<u>Bachelor of Science in Software Development</u>	3	Mandatory
CR_KCOME_6	<u>Higher Certificate in Science in Software Development</u>	3	Mandatory