Title:	Non-linear Data Struct. & Alg. APPROVED
Long Title:	Non-linear Data Struct. & Alg.

Module Code:	COMP7038	**Duration:**	1 Semester

Credits:	5
NFQ Level:	Intermediate
Field of Study:	Computer Science
Valid From:	Semester 1 - 2017/18 (September 2017)
Module Delivered in	2 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 will be introduced to asymptotic cost analysis in order to assess the efficiency of algorithms and data structures when solving a computer science-related problem. The module will examine and assess dynamic programming and backtracking algorithms using nonlinear-based abstract data types.

Learning Outcomes	
On successful completion of this module the learner will be able to:	
LO1	Assess an algorithms computation complexity in terms of time and memory.
LO2	Compare and contrast the interfaces and internal representation of a number of nonlinear abstract data types.
LO3	Design and specify the operations of a nonlinear-based abstract data type in a declarative manner and implement them in a high-level programming language.
LO4	Assess the applicability of dynamic programming and backtracking algorithms to real-world problems.
LO5	Design and implement dynamic programming and backtracking algorithms and compare their formulations and solutions.

Pre-requisite learning		
Module Recommendations *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).*		
12788	COMP7035	Linear Data Struct. & Alg.
Incompatible Modules *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.*		
No incompatible modules listed		
Co-requisite Modules		
No Co-requisite modules listed		
Requirements *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.*		

No requirements listed

Module Content & Assessment

Indicative Content
Algorithm Computational Complexity. Evaluation of algorithm efficiency: Time and memory factors. Formal asymptotic cost analysis: Big O notation. Best, worse and average cases.
Nonlinear-based Abstract Dataypes (ADTs). Declarative semantics of nonlinear-based ADTs. ADT binary tree specification and implementation: Minimal set of operations. Extending the interface: Traversal, path properties and other supplementary operations. Comparable data type-based ADTs: Binary Search Trees (BST).
Graph-based ADTs. Declarative semantics as a generalisation of tree-based ADTs. Graph main concepts (sub-graph, path, cycle, connection) and categories (directed, weighted).
Dynamic Programming Algorithms. Tail recursion. Decreasing recursive design vs. increasing iterative design. Applications: Graphs, searching and resource allocation problems.
Backtracking Algorithms. Satisfaction vs. Optimisation problems. Decision making: Explicit and implicit constraints. Exhaustive search: Alive, expansion and dead nodes. Prunning function. Applications: Puzzles, graphs and resource allocation 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 nonlinear data structures. Produce a report to justify the data structures being chosen in terms of the time and memory complexities for the algorithms implementing the operations.	1,2,3	25.0	Week 6
Project	Design, implement and document a dynamic programming or backtracking 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

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.

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

Module Workload

Workload: Full Time				
Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Lecture deliverying 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 & Directed Learning (Non-contact)	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				
Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Lecture deliverying 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 & Directed Learning (Non-contact)	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

- **Thomas H. Cormen et. al. 2009, *Introduction to Algorithms*, 3rd Ed., MIT Press [ISBN: 9780262033848]**
- **Robert Sedgewick and Kevin Wayne 2011, *Algorithms*, 4th Ed., Addison-Wesley [ISBN: 9780321573513]**

Supplementary Book Resources

- **Narashimha Karumanchi 2016, *Data Structures And Algorithms Made Easy*, CareerMonk [ISBN: 9788193245279]**
- **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/visuali zation/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/j avase/documentation/index.html

Module Delivered in

Programme Code	Programme	Semester	Delivery
CR_KSDEV_8	Bachelor of Science (Honours) in Software Development	4	Mandatory
CR_KDNET_8	Bachelor of Science (Honours) in Computer Systems	8	Mandatory