MODULE SPECIFICATION

Module Code:	CONL704

Module Title:	Data Structures and Algorithms

Level:	7	**Credit Value:**	15

Cost Centre(s):	GACP	JACS3 **code**: HECoS code:	I320 100956

Faculty:	FAST	**Module Leader:**	Jessica Muirhead

Scheduled learning and teaching hours	15 hrs
Guided independent study	135 hrs
Placement	0 hrs
Module duration (total hours)	150 hrs

Programme(s) in which to be offered (not including exit awards)	Core	Option
MSc Computer Science (online)	✓	☐
MSc Computer Science with Big Data Analytics	✓	☐
MSc Computer Science with Cyber Security	✓	☐
MSc Computer Science with Networking	✓	☐
MSc Computer Science with Software Engineering	✓	☐

Pre-requisites
Studied CONL701 Critical Research for Postgraduate Study

Office use only
Initial approval: 04/09/2019 Version no:1
With effect from: 01/01/2020
Date and details of revision: Version no:

MODULE SPECIFICATION

Module Aims
This module aims to give students a thorough grounding in the theories and application of computer algorithms, abstract data types, underlying data structures and their integration to produce efficient programs. This allows students to develop the knowledge and skills to be able to analyse problems and then design, implement, and analyse, effective algorithmic solutions. Students will become familiar with the implications of algorithmic solutions in terms of their computational complexity (space, time and logical) and develop a working knowledge of optimal and approximate (including heuristic) solutions to problems.

Intended Learning Outcomes

Key skills for employability

KS1	Written, oral and media communication skills
KS2	Leadership, team working and networking skills
KS3	Opportunity, creativity and problem solving skills
KS4	Information technology skills and digital literacy
KS5	Information management skills
KS6	Research skills
KS7	Intercultural and sustainability skills
KS8	Career management skills
KS9	Learning to learn (managing personal and professional development, self-management)
KS10	Numeracy

At the end of this module, students will be able to		Key Skills	
1	Analyse and interpret a range of problems and produce designs and models for algorithmic solutions	KS1	KS2
		KS3	KS4
		KS5	KS10
2	Implement computational solutions that demonstrate proficiency in a range of algorithmic techniques and data structures	KS1	KS2
		KS3	KS4
		KS5	KS10
3	Identify and evaluate problems and solutions in terms of their computational complexity	KS1	KS2
		KS3	KS4
		KS5	KS10
4	Develop solutions to problems using appropriate programming techniques	KS1	KS2
		KS3	KS4
		KS5	KS10
5	Explain and justify the structure of algorithms using computational thinking terminology	KS1	KS2
		KS3	KS4
		KS5	KS10

Transferable skills and other attributes
Personal motivation, organisation and time management Written and verbal communication skills Research and analytical skills

Derogations
None

Assessment:

Indicative Assessment Tasks:

This module will be assessed through a series of weekly Portfolio programming tasks designed to test students' understanding of the module content. At the end of the module, a final larger activity will synthesise all of the students' knowledge of data structures and algorithms.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)	Duration or Word count (or equivalent if appropriate)
1	1,2,3	Portfolio	70%	2,000 (equiv)
2	4,5	Practical	30%	1,000 (equiv)

Learning and Teaching Strategies:

The overall learning and teaching strategy is one of guided independent study requiring ongoing student engagement. Online material will provide the foundation of the learning resources, requiring the students to login and engage on a regular basis throughout the eight-week period of the module. There will be a mix of suggested readings, discussions and interactive content containing embedded digital media and self-checks for students to complete as they work through the material and undertake the assessment tasks. The use of a range digital tools via the virtual learning environment together with additional sources of reading will also be utilised to accommodate learning styles. There is access to a helpline for additional support and chat facilities through Canvas for messaging and responding.

Syllabus outline:

1. Introduction to Python and Object Orientated Programming
2. Algorithms and complexity
3. Stacks, queues and lists
4. Recursion
5. Sorting and searching
6. Trees and tree algorithms
7. Graphs and graph algorithms

Indicative Bibliography:

Essential reading

Miller, B., & Ranum, D. (2013) *Problem Solving with Algorithms and Data Structures.* Franklin, Beedle & Associates. Available online: https://runestone.academy/runestone/books/published/pythonds/index.html

Other indicative reading

Aho, A.V., Hopcroft, J.E., & Ullman, J. (1983) *Data Structures and Algorithms.* Addison-Wesley.

Barry, P. (2016) *Head First Python: A Brain-Friendly Guide.* O'Reilly Media, Inc.
Cormen, T.H. (2009) *Introduction to Algorithms.* 3rd ed. Cambridge, Mass: MIT Press.

Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2013) *Data structures and algorithms in Python.* John Wiley & Sons Ltd.

Knuth, D.E. (1997) *The Art of Computer Programming, Volume 1: Fundamental Algorithms.* 3ed. Addison-Wesley.

Matthes, E. (2016) *Python crash course: a hands-on, project-based introduction to programming.* No Starch Press.

Neapolitan, R.E. and Naimipour, K. (2014), *Foundations of Algorithms.* 5th ed. Jones & Bartlett Learning.

Runestone (n.d.) *Foundations of Python Programming.* Available online: https://runestone.academy/runestone/books/published/fopp/index.html

Sedgewick, R. (2011) *Algorithms.* 4th ed. Addison-Wesley.

Wentworth, P., Elkner, J., Downey, A. B., & Meyers, C. (2019) *How to Think Like a Computer Scientist.* 3rd ed. Available online: https://buildmedia.readthedocs.org/media/pdf/howtothink/latest/howtothink.pdf