Prifysgol Wrecsam
Wrexham University

Module specification

When printed this becomes an uncontrolled document. Please access the Module Directory for the most up to date version by clicking on the following link: Module directory

Module Code	CONL704
Module Title	Data Structures and Algorithms
Level	7
Credit value	15
Faculty	FACE
HECoS Code	100956
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
MSc Computer Science (online)	Core
MSc Computer Science with Artificial Intelligence	Core
MSc Computer Science with Big Data Analytics	Core
MSc Computer Science with Cyber Security	Core
MSc Computer Science with Software Engineering	Core
MSc Computer Science with UX	Core

Pre-requisites

None

Breakdown of module hours

Learning and teaching hours	15 hrs
Placement tutor support	0 hrs
Supervised learning e.g. practical classes, workshops	0 hrs
Project supervision (level 6 projects and dissertation modules only)	0 hrs
Total active learning and teaching hours	15 hrs
Placement / work based learning	0 hrs
Guided independent study	135 hrs
Module duration (total hours)	150 hrs

For office use only	
Initial approval date	04/09/2019

For office use only	
With effect from date	01/01/2020
Date and details of revision	27/06/2024 Programme revalidation
Version number	2

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 programmes. Students will learn about the important properties of some data structures and algorithms that are of foundational importance to modern computer science. These will be implemented using a general-purpose programming language. 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.

Module Learning Outcomes - at the end of this module, students will be able to:

1	Analyse and interpret a broad range of complex problems, demonstrating originality and critical thinking in designing sophisticated models and algorithmic solutions.
2	Implement computational solutions that demonstrate a high level of proficiency across a diverse array of algorithmic techniques and data structures.
3	Identify and critically evaluate problems and their solutions in terms of computational complexity.
4	Demonstrate an in-depth understanding and the ability to use and design programs incorporating appropriate and complex data structures.
5	Provide a critical explanation and justification of the structure of algorithms using advanced computational thinking terminology.

Assessment

This section outlines the type of assessment task the student will be expected to complete as part of the module. More details will be made available in the relevant academic year module handbook.

Indicative Assessment Tasks:

This module will be assessed through two programming tasks designed to test students' understanding of the module content. The first coursework is designed to assess students' capacity to implement ideas in software. At the end of the module, a final larger activity will synthesise all of the students' knowledge of data structures and algorithms and will require them to both write programmes in an object-oriented language and provide a critical reflection of the approach, coding practices and code complexity.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1, 2, 3	Coursework	50%
2	2, 3, 4, 5	Coursework	50%

Derogations

None

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 log in and engage regularly throughout the eight weeks of the module. There will be a mix of suggested readings, discussions and video content containing embedded digital content and self-checks for students to complete as they work through the material and undertake the assessment tasks. The use of a range of 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.

Indicative Syllabus Outline

- Introduction to Python and Object Orientated Programming
- Algorithms and complexity
- Sets and dictionaries
- Stacks, queues and lists
- Recursion, Sorting and searching
- Trees and tree algorithms
- Graphs and graph algorithms

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update.

Essential Reads

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

Other indicative reading

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

P. Barry, *Head First Python: A Brain-Friendly Guide.* Sebastopol, CA: O'Reilly Media, Inc., 2009.

T.H. Cormen, C. E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*. 3rd ed. Cambridge, MA: MIT Press, 2009.