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	COM477
Module Title	Applied Computational Methods
Level	4
Credit value	20
Faculty	FACE
HECoS Code	101029
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
BSc (Hons) Computer Science	Core
BSc (Hons) Computer Science (with Industrial Placement)	Core
BSc (Hons) Software Engineering	Core
BSc (Hons) Software Engineering (with Industrial Placement)	Core
BSc (Hons) Cyber Security	Core
BSc (Hons) Cyber Security (with Industrial Placement)	Core

Pre-requisites

None

Breakdown of module hours

Learning and teaching hours	36 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	**36** hrs
Placement / work based learning	0 hrs
Guided independent study	164 hrs
Module duration (total hours)	**200** hrs

For office use only	
Initial approval date	08/11/2023
With effect from date	Sept 2024
Date and details of revision	
Version number	1

Module aims

This module is designed to introduce computational and mathematical concepts required for the application of computer science, software engineering and cyber security. The emphasis of this module will be to demonstrate these concepts against their relevance to computing and the wider digital industries to further computing-related cognitive and practical abilities. Students will engage in ongoing coursework pieces that utilise problem solving skills to relate concepts, theories and methods to real-world and virtual scenarios. These will revolve around industry standard tools, hardware and software.

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

1	Identify mathematical problems in computing-related real-world scenarios.
2	Apply computational techniques to solve technical problems.
3	Relate the use of software packages to subject-specific mathematical solutions.

Assessment

Indicative Assessment Tasks:

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.

Coursework will take place throughout the module using scenarios based upon mathematical fundamentals and how they directly relate to computational solutions. Indicatively this may look like 4-6 standalone topic areas explored every 2-3 weeks of teaching. Students will be required to demonstrate their knowledge of applied mathematics and system hardware through questioning and demonstration surrounding these scenarios. These will be real-world scenarios and will represent applied theory according to the nature of computing, software and security.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1,2,3,	Coursework	100%

Derogations

None

Learning and Teaching Strategies

In line with the Active Learning Framework, this module will be blended digitally with both a VLE and online community. Content will be available for students to access synchronously and asynchronously and may indicatively include first and third-party tutorials and videos, supporting files, sections of code/diagrams or any additional content that supports their learning.

As this module progresses, a structured strategy will be used to support the students engaging with the key threshold concepts relating to the learning outcomes. The module will include a balanced mixture of engaging tutor-led lectures, demonstrations, and facilitation. As the module continues experiential and peer learning strategies will be encouraged as the students' progress with their coursework.

Indicative Syllabus Outline

Yearly content will be updated to represent the most appropriate content for current industry technologies, but a list of indicative topics could include:

- Algebra & Maths for Programming Languages
- Number Systems in Computer Science
- Linear Algebra
- Algorithms and Optimisation
- Artificial Intelligence Systems
- Statistics and Probability
- Set theory, Relations and Functions
- Introduction to Graphs, Trees and Networks
- Computational Complexity

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update. Please *ensure correct referencing format is being followed as per University Harvard Referencing Guidance.*

Essential Reads

R. T. White, A. T. Ray, *Practical Discrete Mathematics: Discover math principles that fuel algorithms for computer science and machine learning*, Birmingham: Packt Publishing, 2021.

Other indicative reading

P. Farrell, *Math Adventures with Python: An Illustrated Guide to Exploring Math with Code*, California: No Starch Press, 2019.
R. T. Kneusel, *Math for Deep Learning: What You Need to Know to Understand Neural Networks*, California: No Starch Press, 2021
T. Nield, *Essential Math for Data Science: Take Control of Your Data with Fundamental Linear Algebra, Probability, and Statistics*, 2022.
J. Vince, *Foundation Mathematics for Computer Science: A Visual Approach*, Hereford: Springer, 2020.