a.r.u.

Module Definition Form (MDF)

Module code: MOD004428	Version: 12 Date Amended: 16/Jul/2026

1. Module Title
Core Mathematics for Computing

2a. Module Leader
Ian van der Linde

2b. School
School of Computing and Information Sciences

2c. Faculty
Faculty of Science and Engineering

3a. Level
4

3b. Module Type
Standard (fine graded)

4a. Credits
15

4b. Study Hours
150

5. Restrictions

Type	Module Code	Module Name	Condition
Pre-requisites:	None		
Co-requisites:	None		
Exclusions:	None		
Courses to which this module is restricted:	None		

LEARNING, TEACHING AND ASSESSMENT INFORMATION

6a. Module Description

This module provides the essential mathematical and statistical foundations for computer science and related disciplines. Mathematical reasoning is a core skill for computing professionals and researchers, underpinning areas such as software development, artificial intelligence, cybersecurity, data science, and computer networks. The module supports the professional body accreditation requirements of your programme while developing analytical and problem solving skills that are highly valued by employers.

You'll study key topics in discrete mathematics, including set theory, logic, Boolean algebra, functions, matrices, sequences and series, and summation and product notation. These concepts are explored in the context of computing, demonstrating how mathematical structures and methods underpin algorithms, data structures, databases, programming languages, and formal reasoning about computational systems.

The module examines the relationship between Boolean algebra and the logical operators used in programming and digital systems, including the evaluation and simplification of logical expressions. Mathematical functions, including injective, surjective, and bijective mappings, are related to computational processes, while common numerical operations such as modular arithmetic, greatest common divisors, and least common multiples are introduced through practical computing applications. Sequences and series are used to develop an understanding of algorithmic growth and complexity, and combinatorial techniques such as permutations and combinations are applied to problems in algorithm design, cryptography, and the analysis of computational search spaces. Matrices are studied as a versatile mathematical tool with applications in image processing, graph and network representation, machine learning, and scientific computing.

The module also introduces the fundamental principles of probability and statistics, providing a basis for reasoning under uncertainty and analysing data. You'll study descriptive and inferential statistical methods, including data visualisation, common probability distributions, measures of central tendency and variability, confidence intervals, hypothesis testing, correlation, and simple linear regression. These techniques are widely used in computing applications, including data science, machine learning, experimental evaluation, and software and systems research.

By the end of the module, you'll have developed a strong mathematical foundation for later study and professional practice, together with the confidence to apply quantitative methods to a wide range of computational problems. Mathematics and statistics skills are regarded as a core competency in computing professionals, and graduates with these skills are highly valued by employers in all job roles in computing and more widely.

6b. Outline Content

Set Theory: sets; members; empty and universal sets; Venn and Euler diagrams; set operations (complement, union, intersection, difference and symmetric difference); relationship with and use of logic operations; roster definition and set builder notation; De Morgan's laws on sets; supersets and subsets; equality and equivalence; existential and universal quantifiers; number types; disjoint sets; disjoint union, cardinality; transfinite sets; ordered sets; number systems; cartesian product; powersets; partitions; set representation of lists, trees, and graphs.

Sequences and Series: identifying sequence type, arithmetic, geometric and harmonic progressions; rearrangement to standard form; sums and means of sequences; relationship to algorithms; periodic, convergent and divergent sequences; limiting behaviour; recurrent sequences (such as Fibonacci, Lucas and Padovan).

Summations and Product Notation: sigma and pi notation; evaluating expressions; finite and infinite sums and products; decomposition; simplification to closed form; re-arrangement; identifying infinite, annihilating and converging expressions; conversion between sums and products using logarithms; use in algorithmic complexity analysis.

Matrices: algebraic properties of matrices; matrix types such as row/column vector, square, symmetric, zeroes, ones, and diagonal; origin; matrix order; operations including transposition, Hadamard, scalar and regular multiplication, addition, inverse, trace, rank, negation, convolution; cellular automata; matrix representation of lists, trees, and graphs.

Functions: univariate, bivariate, and multivariate functions; plotting functions (such as absolute, square, and reciprocal); horizontal and vertical line tests; general, injective, surjective and bijective functions; strictly and non-strictly increasing/decreasing functions; graph-theoretic view of functions; gcd, lcm; floor/ceiling/rounding; piecewise functions (such as rect, Heaviside, signum); modulus; logarithms; computational functions (such as prime, perfect, divisor count/sum, Euler totient).

Boolean Algebra: Boole's postulates; basic logic operations (not, buffer, and, nand, or, nor, xor, xnor); De Morgan's law on logic expressions; tautology and contradiction; implication; truth tables; constant, monotonic, linear, balanced and evasive operations; simplification of Boolean expressions; comparison with regular algebra (such as by identify, commutative, associative, distributive, redundancy, inverse in/equality, and other laws).

Probability: terminology (experiment, trial, outcome, event); random numbers; distributions (uniform, Gaussian); events and non-events; addition and multiplication rules for independent and non-independent events; permutations and combinations; probability-based paradoxes; tabulation; partitioning solution spaces; probability trees; histograms.

Statistics: basic terminology (population, sample, sample size), distinction between descriptive and inferential statistics; extremum; measures of central tendency (mean, median, mode) and dispersion (range, mid-range, standard deviation); skewness and kurtosis; linear correlation; t-tests (one sample, paired and independent sample).

6c. Key Texts/Literature

The reading list to support this module is available at: https://readinglists.aru.ac.uk/

6d. Specialist Learning Resources

Laboratories equipped with the MATLAB numerical computing environment (The Mathworks Inc. Natick, MA), or a freeware alternative such as GNU Octave.

7. Learning Outcomes (threshold standards)		
No.	**Type**	**On successful completion of this module the student will be expected to be able to:**
1	Knowledge and Understanding	Use mathematical notations and apply a range of operations relevant to computer science, including bases, exponents and logarithms, sequences, summation/product notation, permutations and combinations.
2	Knowledge and Understanding	Apply Boolean algebra and other core concepts in discrete maths and set theory that
3	Intellectual, practical, affective and transferrable skills	Interpret, manipulate, simplify and solve elementary algebraic expressions, in the core mathematical domains covered by the module
4	Intellectual, practical, affective and transferrable skills	Present data graphically, and how to interrogate data to make statistical and probabilistic inferences.

8a. Module Occurrence to which this MDF Refers				
Year	**Occurrence**	**Period**	**Location**	**Mode of Delivery**
2026/7	ZZF	Template For Face To Face Learning Delivery		Face to Face

8b. Learning Activities for the above Module Occurrence			
Learning Activities	**Hours**	**Learning Outcomes**	**Details of Duration, frequency and other comments**
Lectures	12	1-4	Lectures = 1 hr/week for 12 weeks
Other teacher managed learning	24	1-4	Tutorials = 2 hr/week for 12 weeks
Student managed learning	114	1-4	Self-Study = 9½ hr/week for 12 weeks
TOTAL:	150		

9. Assessment for the above Module Occurrence

The assessment for some modules includes a presentation (see below for details on how this particular module is assessed). We recognise that presenting, especially to a large group, can cause significant anxiety or distress for some students. If this may affect you, please contact your Module Leader or the Disability and Neurodiversity Service as early as possible. We will work with you to consider appropriate reasonable adjustments. Depending on your individual needs, these might include, for example, presenting to a smaller group or presenting only to a member of academic staff.

Assessment No.	Assessment Method	Learning Outcomes	Weighting (%)	Fine Grade or Pass/Fail	Qualifying Mark (%)
010	Coursework	1-4	50 (%)	Fine Grade	30 (%)
In-Class Test 1 (1 1/2 Hours)					

Assessment No.	Assessment Method	Learning Outcomes	Weighting (%)	Fine Grade or Pass/Fail	Qualifying Mark (%)
011	Coursework	1-4	50 (%)	Fine Grade	30 (%)
In-Class Test 2 (1 1/2 Hours)					

In order to pass this module, students are required to achieve an overall mark of 40% (for modules at levels 3, 4, 5 and 6) or 50% (for modules at level 7*).

In addition, students are required to:
(a) achieve the qualifying mark for each element of fine graded assessment as specified above
(b) pass any pass/fail elements

[* the pass mark of 50% applies for all module occurrences from the academic year 2024/25 – see Section 3a of this MDF to check the level of the module and Section 8a of this MDF to check the academic year]