Module code: MOD004428	Version: 11 Date Amended: 11/Jul/2025

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

As a student embarking on a degree in computer science or a closely related discipline, this module will equip you with the core mathematical skills needed to succeed. This module also contributes to the professional body accreditation of your programme of study, reflecting the criticality of mathematics/statistics skills in professional computing roles and in computer science research.

During much of the module, you'll be studying topics in discrete mathematics, such as set theory and logic, Boolean algebra, functions, matrices, sequences/series and product/summation notations. Further topics include probability and statistics, which are useful in understanding the behaviour of non-deterministic algorithms, data visualisation, and in the design and implementation of computer science research projects. The topics you''ll study will be directly related to computing principles. For example the use of set theory for: the representation of computational structures such as lists, trees and graphs; computations on discrete collections of data (such as in databases); the relationship between number classes and data types and in evaluating computability; and parallels between set theoretic operations and programming logic. The relationship between Boolean algebra and logic operators used in computer programming will be discussed, along with topics like the evaluation and simplification of Boolean expressions. Topics such as sequences and series will be related to elementary algorithm complexity (e.g., linear, logarithmic, and exponential functions), and mathematical functions (injective, surjective, bijective) will be related to program functions, with common functions found in nearly all non-trivial computer programs (such as modulus, floor/ceiling, and numerical operations such as gcd and lcm) being introduced and demonstrated in context. Permutations and combinations will be related to computer security, and the notion of intractable computational problems. Matrices will be in discussed in terms of their ability to represent computational structures such as images, graphs, computer networks and cellular automata. Core descriptive and inferential statistics used for data visualisation and hypothesis testing (including histograms, distribution types, measures of central tendency and dispersion, and basic inferential statistics such as t-tests and linear correlation) will be examined.

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**
2025/6	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					
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]