SCHOOL OF INNOVATIVE TECHNOLOGIES AND ENGINEERING

Department of Industrial Systems Engineering

Module Information Pack

BSc (Hons) Computer Science with Network Security
(BCNS/24A/FT2)

Module Name: Maths for Computer Science

Module Code: BCNS1101C

Academic Year 2024 – Semester 1 (FT)

Programme Director: Dr (Mrs) Sandhya Armoogum

Programme Coordinator: Prof (Dr) Vinaye Armoogum

Module Coordinator: Dr K. Dookhitram

Module Convenor: Mr JHEENGUT Ramakant

Office:

Phone No. 58241905

E-mail: rajheengut@umail.utm.ac.mu

Lecture Day and Time: Monday 14 00 to 17 00

Credits & Level: 6 credits, Level 1

Pre-requisites (If applicable): None

Co-requisites (If applicable): None

Method of Delivery & frequency of Class: 15 weeks; 15 x 3 hrs sessions of lectures and tutorial sessions. The students are recommended to conduct self-development activities.

Method and Criteria of Assessment:

Coursework (30%)	
Class Test 1	15%
Class Test 2	15%
Written Examinations (70%)	
Exam	70%
Total:	**<u>100%</u>**

Module Aims:

The aim of this module is to provide students with a solid understanding of foundational mathematical concepts relevant to computer science. Through theoretical study and practical applications, students will develop the mathematical skills necessary to analyse algorithms, design efficient data structures, and comprehend advanced computer science topics.

Module outlines:

The module begins with an exploration of logical operations and proofs, introducing students to fundamental concepts in propositional and predicate logic, along with proof techniques such as direct proofs, proof by contradiction, and mathematical induction. Subsequently, students delve into set theory, functions, and relations, gaining insight into set operations, function properties, and relation types. Following this, student's transition to Boolean algebra, where they learn about Boolean operators, laws, and applications in digital logic circuits. The module then progresses to cover combinatorial topics such as counting techniques, permutations, and combinations, essential for analysing algorithm efficiency and data structure design. Students are then introduced to finite state machines, graph theory, and number theory, exploring their applications in modelling computational problems and designing algorithms. Finally, the module concludes with a study of matrices, focusing on matrix operations, solution techniques for linear systems, and their significance in computer graphics, optimization, and algorithm analysis.

Learning Objectives:

1. **Demonstrate proficiency in logical operations and proofs** by applying propositional and predicate logic to analyze and construct logical arguments, and employing proof techniques such as direct proofs, proof by contradiction, and mathematical induction.
2. **Apply set theory concepts** to model and solve problems including set operations, functions, and relations.
3. **Utilize Boolean algebra** to design and analyze digital logic circuits, including the simplification of Boolean expressions and the implementation of logic gates.
4. **Apply counting techniques**, permutations, and combinations.
5. **Design and analyze finite state machines** to model and solve problems in control systems, parsing, and other computational processes.
6. **Apply graph theory** concepts and algorithms to model and solve problems including graph traversal.
7. **Apply number theory concepts**, including prime numbers and modular arithmetic.
8. **Apply matrix operations** and techniques to solve systems of linear equations, and analyze algorithms.
9. **Integrate mathematical concepts** to analyze algorithms, design efficient data structures, and solve computational problems effectively.
10. **Communicate mathematical ideas** effectively through written explanations and presentations, demonstrating understanding and application of mathematical concepts.

Module Learning Outcomes:

By the end of this module, students should be able to:

1. Understand and apply logical operations and proofs to analyze and construct arguments.
2. Demonstrate proficiency in set theory, functions, and relations, and their applications in modeling computational problems.
3. Apply Boolean algebra concepts to design and analyze digital logic circuits and Boolean functions.
4. Analyze and design combinatorial structures using counting techniques, permutations, and combinations.
5. Design and analyze finite state machines for modeling and solving computational problems.
6. Apply graph theory concepts and algorithms to model and solve problems.
7. Demonstrate understanding of fundamental concepts in number theory.
8. Apply matrix operations and techniques for solving systems of linear equations.

Lecture Schedule:

Week No.	Topics / Tutorial
1	Logical operations and proofs
2	Set theory, functions and relations
3	Boolean algebra
4	Digital logic and operations
5	Digital logic and operations
6	Counting techniques
7	**Written Test 1** (Based on Week 1 to 6)
8	Permutations and combinations
9	Finite state machines
10	Graph theory
11	Number theory
12	Matrices and solution of linear systems
13	Matrices and solution of linear systems
14	**Written Test 2 (**Based on Week 7 to 13)
15	Revision

Reading List:

Recommended Text:

- E. Kreyszig, Advanced Engineering Mathematics
- G. James, Modern Engineering Mathematics

Past Exam Papers

Past Exam Papers are downloadable at http://www.utm.ac.mu/resource/.