Module code	SM-1301			
Module Title	Discrete Mathematics			
Degree/Diploma	Bachelor of Science (Mathematics)			
Type of Module	Major Option			
Modular Credits	4	**Total student Workload**	10	hours/week
		Contact hours	4	hours/week
Prerequisite	None			
Anti-requisite	None			

Aims

To introduce concepts and techniques of mathematics (mostly discrete) needed for various areas of Mathematics and Computer Science. On completing this module, the student should be able to use the concepts and techniques of discrete mathematics to prove programme correctness, investigate algorithm complexity, set up economical computer networks and model computation.

Learning Outcomes

On successful completion of this module, a student will be expected to be able to:

Lower order :	30%	- describe the arithmetic techniques developed in ancient Babylon and Egypt - describe the Greek contributions to geometry and number theory - describe the development of algebra in medieval Islam - describe the contributions of pre-modern China and India to mathematical theory - summarise the advances made in all branches of mathematics from the 16th to the 19th centuries
Middle order :	60%	- formulate and solve the same types of problems that early mathematicians themselves faced and solved (e.g. taking square roots, proving the irrationality of surds, enumerating the Platonic solids, constructing figures with straight edge and compass, solving algebraic equations, factorising cubic polynomials, manipulating logarithms, summing infinite series) - describe and use the axiomatic deductive method in geometry
Higher order:	10%	- extend the techniques used by early mathematicians to new classes of mathematical problems - work independently

Module Contents

- Logic statements, connectives, canonical forms (CNF, DNF), inference, formal proof, predicates.
- Sets, multi-sets; set of natural numbers, basic properties; countable sets.
- Counting techniques for counting permutations, combinations, bit strings and subsets.
- Relations: representations and manipulation through Boolean matrices, digraphs; Posets, Hasse diagrams.
- Functions and their growth.
- Algorithms: proof of correctness; computational complexity.
- Graphs: vertex, edge; incidence, adjacency matrices, vertex degree; complete bipartite graphs; subgraphs, graph isomorphism; paths, cycles, connected graphs; Euler circuits, Hamilton cycles.
- Trees: minimum-weight spanning trees, minimum-distance trees.
- Modelling computation: languages, grammars; finite-state machines; Turing machines.

Assessment	Formative assessment	Tutorial and feedback.
	Summative assessment	Examination: 60%
		Coursework: 40% - 2 tests (40%)