Discrete Mathematics syllabus

Course Description

Discrete mathematics covers mathematical topics relating to discrete structures and processes that students of computer science will encounter throughout their study. This module helps you hone your skills in thinking abstractly. This module covers widely applicable mathematical tools for computer science, including topics from logic, set theory, combinatorics, and graph theory. It includes practice in reasoning formally and proving theorems. It helps learning to reason rigorously, and to distinguish correct from faulty mathematical arguments. Through this module, you will develop the fundamental discrete mathematical tools that will support you throughout the BSc Computer Science programme. You will learn a wide range of the discrete mathematical concepts and techniques that underpin Computer Science.

Course Goals and Objectives

Upon successful completion of this course, you will be able to:

1. Manipulate Boolean expressions and interpret them as set descriptors
2. Interpret logical statements, produce truth tables, and prove logical properties
3. Explain the relationship between recursion and induction and do simple induction proofs
4. Describe the correspondence between relations and graphs, and prove properties of special relations
5. Define properties of trees and graphs
6. Able to find the number of possible combination or permutations in a given scenario.

Textbook and Readings

The essentials readings for this course will come from the following text book, which is available in the University of London digital library:

- Kenneth, H, Rosen. Discrete Mathematics and its Applications. (2012) 7th Edition
- David Mackinson, Sets, Logic and Maths for Computing, Springer Verlag. 2012

This course does not require you to read the whole book, you will be given specific readings for each topic from these texts are listed with direct links on the Readings page for each topic.

You will also be asked to do some independent research from online sources or using the University of London digital library.

Course Outline

The course consists of 10 topics, each of which spans 2 weeks.

Topic 1: Sets	**Learning Objectives:** • Understand sets and powersets of a manipulate • Learn the listing methods and rules of inclusion methods • Learn to manipulate set operations • Represent a set using Venn diagram • Understand and apply De Morgan's law • Understand, and apply commutative, associative and distributive laws
Topic 2: Functions	**Learning Objectives:** ● Understand what is a function and define its domain, and co-domain, ● Define the range of a function and the pre-image of an element ● Recognise linear, quadratic and exponential functions ● Define and learn examples of injective and surjective functions. ● Build functions compositions and a function's inverse. ● Recognise and define some particular functions such as Absolute value, Floor, Ceiling

Topic 3: Propositional Logic	**Learning Objectives:** ● Define propositional logic and learn some of its properties ● Give some examples of domains where propositional logic is used ● Defining a proposition in mathematical context and distinguish examples of sentences considered as propositions and other that are not propositions ● Learn how propositional variables help to simplify notations ● Learn how we can build a truth table and give an example of how it works ● Practice how to build compound statements using logical operators and take into consideration the order of precedence of the operators ● Define truth sets and learn some examples of truth sets for some compound propositions ● Define what is an implication and equivalence, what are their properties and build their truth table ● Learn some important laws of propositional logic including De Morgan's laws, and practice their use in building a reasoning, and proving equivalence
Topic 4: Predicate logic	**Learning Objectives:**

	● Learn how predicate logic has some limitations in describing some mathematical statements ● Learn and practice how to use logical quantifiers, nested quantifiers, binding variables and logical operators. ● Learn and take into consideration the precedence of quantifiers ● Learn the intuition behind De Morgan's laws, how to formalise its expression and how it can be used in negating nested quantifiers. ● Learn in detail some of the widely used rules of inference, with and without quantifiers, and how they are used to build valid arguments. ● Discover some examples of formal fallacies.
Topic 5: Boolean algebra	**Learning Objectives:** ● Discover the history of Boolean algebra as well as its applications. ● Introduce a form of Boolean Algebra called 'Two-valued Boolean Algebra and its basic rules and operations. ● Introduce the six axioms of Huntington's, and discover some basic theorems that can be deduced from these six axioms such as De Morgan's theorem and principle of duality.. ● Learn, the main mains ways of proving theorems. ● Introduce Boolean functions, and present their algebraic and standardised forms. ● Introduce the basic logic gates as well as their corresponding truth tables ● Learn how to represent De Morgan's laws using logic gates. ● Introduce and define combinational circuits or logic networks, and learn how they can be

	useful in designing systems for solving a particular problem. ● Learn how to build a logic network from a Boolean function and how to find a Boolean function from its corresponding circuit. ● Introduce and discuss the Karnaugh map, or K-Map.
Topic 6: induction & Recursion	**Learning Objectives:** ● Learn how to use direct proof, proof by contraposition and proof by contradiction to prove mathematical statements ● Introduce Mathematical induction and learn the intuition behind it. ● Learn the structure of Induction and discover how it can be used to prove formulas, Inequalities and Divisibility. ● Assess the validity of an induction and explain why it can be incorrect. ● Discover another form of mathematical induction, called strong induction and show how it can be used in proofs ● Learn how induction, strong induction and well ordering can be used equivalently in building proofs. ● Introduce recursion and present recursively defined functions, recursively defined sets and recursive algorithms. ● Introduce recurrence relations and linear sequences and discover an example of the Hanoi tower problem. ● Discover some examples on how to solve a linear recurrence. ● Discover the special case of solving Fibonacci recurrence. ● Learn how strong induction can be used to solve recurrence relations.

Topic 7: Graphs	**Learning Objectives:** ● Get some insight on the origins of graph theory and some of its applications ● Learn what is a graph, a vertex and edge ● Discover what is a loop, a parallel edge and direct graph ● Identify the difference between a walk, trail, circuit, path and cycle in a graph. ● Learn the meaning of an Euler path, a Hamiltonian path and a Hamiltonian cycle. ● Learn what is a connected graph, a strongly connected digraph the concept of the transitive closure of a graph ● Define the degree of a vertex, represent a graph using its degree sequence, learn some of its properties and apply them to some examples. ● Define what a simple graph and investigate the concept of regular graphs. ● Define what is a complete graph and show some of its properties. ● Identify isomorphic graphs and examine some examples. ● Discover a special type graph called a bipartite graph and examine some examples. ● Learn the concept of maximum matching and how to apply the Hopcroft-Karp algorithm to find the maximum matching in a bipartite graph. ● Introduce the concept of weighted graphs. ● Discover Dijkstra's algorithm, and see how it works in practice ● Learn how to represent a graph using an adjacency list or adjacency matrix and discover some of their properties.
Topic 8: Trees	**Learning Objectives:** ● Discover the mathematical concept of trees ● Define what is an acyclic graph, a tree, a forest and rooted trees. ● Learn what is a spanning tree of a graph and how to construct a spanning tree

	● Define isomorphic spanning trees and discover some examples ● Define the weight of spanning tree and what is the minimum-cost spanning tree ● Discover Kruskal's algorithm for finding minimum-cost spanning trees and how it works through an example. ● Discover Prim's algorithm and how it works through an example. ● Define the depth and height in a tree and lean some properties of rooted trees. ● Define isomorphic rooted trees and learn some of their properties. ● Learn what is a binary search tree and how it's used in practice through examples.
Topic 9: Relations	**Learning Objectives:** ● Discover and define the concept of mathematical relation ● Learn how to evaluate the cartesian product ● Define a relation between elements of the same set ● Learn how to represent relations via matrices or using directed graphs and examine some examples ● Understand the property of transitivity and the concept of transitive closure of a relation ● Define reflexive, symmetric and anti-symmetric relations learn how to assess if a digraph or an adjacency matrix is satisfying each of these three relations ● Define equivalence relations and build equivalence classes and examine examples demonstrating these concepts ● Understand partial and total order and examine some examples.

	Learning Objectives:
Topic 10: Combinatorics	● Introduce some applications of combinatorics in computer science. ● Learn the basic counting rules such as the product rule, the addition rule, the subtraction rule and the division rule. ● Learn the pigeonhole principle, and discover its uses with an example. ● Introduce a more generalised version of the pigeonhole principle and work through some examples that put it into practice. ● Learn how counting problems can be solved by finding the number of ways to arrange a specified number of distinct elements, where the order matters or, in some cases, doesn't. ● Learn and practice counting problems using the two concepts of permutations and combinations as well as generalised permutations and combinations. ● Introduce the binomial theorem, Pascal's identity and Pascal's triangle and learn some of its applications. ● Learn that counting problems can be reduced to simply finding the number of ways in which objects can be placed into boxes. ● Count the number of ways in which objects can be placed into boxes, where objects can either be distinguishable or indistinguishable.

Learning Activities of This Course

The course is comprised of the following elements:

● Lecture videos. In each module the key concepts will be presented through a collection of short video lectures. You may stream these videos for playback within the browser by clicking on their titles or download the videos.

- Readings. Each topic may include several suggested readings. These are a core part of your learning, and, together with the videos, will cover all of the concepts you need for this course.
- Practice Quizzes. Each topic will include practice quizzes, intended for you to assess your understanding of the key concepts. End of topic quizzes in the second half of the course will not contribute to your final grade, but will provide important practice for your exam. You will be allowed unlimited attempts at each practice quiz. Each attempt may present a different selection of questions to you. There is no time limit on how long you take to complete each attempt at the quiz. All practice quizzes do not contribute toward your final score in the course.
- Graded Quizzes. Almost every week in topics 1-5 will include one summative quiz in the end of week. These quizzes will contribute to your coursework grade. You will be allowed two attempts per quiz. Your average score will be used when calculating your final score in the summative quiz. The time between these two attempts is at least 8 hours.
- Peer Reviewed Assignments. Topics 1-5 include peer reviewed assignments. These exercises will test your understanding of the concepts. You will read a short prompt, then perform a task in response. You will then be required to review three of your peers' submissions. Peer reviewed assignments will not contribute to your final grade, but will be valuable practice for your coursework and exam.
- Discussion Prompt. Each topic include discussion prompts. You will see the discussion prompt alongside other items in the lesson. Each prompt provides a space for you to respond. After responding, you can see and comment on your peers' responses. All prompts and responses are also accessible from your private group discussion forum.

How to Pass This Course

The course has two major assessments each worth 50% of your grade:

- Coursework: this consists of several activities that you do on the Coursera platform and which will be assessed half way through course (after week 12)
- Written examination: you will take this at an examination centre in your country.

There are also several activities that are graded but have 0 weight. That means that they will not count towards your final grade, but they are a key part of your learning and you need to do them.

The mark shown on the Coursera platform is your coursework mark and you should remember that the exam counts for another 50%.

The coursework consists of several activities. This is a detailed breakdown of all of the marks.

Activity	Required?	Deadline week	Estimated time per module	% of final grade
End of week quizzes for topics 1-5	Yes	1-12	1-2 hours	20%
Written, staff graded coursework	Yes	12	20 hours	30%
Written examination	Yes	22	2 hours 15 minutes	50%