BOOK: Kenneth, H, Rosen. Discrete Mathematics and its Applications. (2012) Global Edition

Week	Topic	Secions	Essential reading	Topics Covered	Learning Objectives	Exercises
1	Sets	2.1	117–126	The definition of a set; elements of a set; cardinality of a set; listing method; set builder method; subsets of a set; and powersets.	Define a set, the elements of a set and the cardinality of a set. Define the concepts of the universal set and the complement of a set, and the difference between a set and a powerset of a set. Define the concepts of the union, intersection, set difference and symmetric difference, and the concept of a membership table.	pp.127–128, exercises 1–8 and 12–19.
2	Sets	2.2	128–137	set operations; membership tables; Venn diagrams; the complement of a set; De Morgan's laws; and set properties.	Understand the concept of Venn diagrams and how they are used to represent and compare different set expressions. Understand and prove De Morgan's law using membership tables.	pp.138–139, exercises 1–9, 14, 15, 17, 18, 22–26, 32 and 34–36.
3	Functions	2.3	140–146	the definition of a function; the domain and co-domain; the range; injection (one-to-one); surjection (onto); and plotting functions.	Define a function. Describe the properties of functions. Explain how to plot a function.	pp.153–154, exercises 1–4 and 6–17.
4	Functions	2.3	146–153	function composition; bijections; and the floor and ceiling functions.	Discuss special functions. Describe inverse functions.	pp.153–155 exercises 5, 14–25, 30–36, 44–46 and 49.
5	Propositional logic	1.1	1–12	the definition of a proposition; truth tables and truth sets; and compound propositions.	Explain and apply basic concepts of propositional logic. Construct truth tables of propositions and use them to demonstrate the equivalence of logical statements.	p.12, exercises 1–3.
6	Propositional logic	1.1	1–12.	logical implication; converse, contrapositive and inverse; equivalence; the laws of propositional logic (including De Morgan's laws); and the precedence of logical operators.	How to formalise a logical implication Apply the laws of propositional to analyse propositions and arguments.	p.13, exercises 6–12 p.14, exercises 19–21 p.33, exercises 6–11.
7	Predicate logic	1.4, 1.5	34–49 and 53–60	the definition of predicates; quantifiers; logical operators; and nested quantifiers.	Describe the basic concepts of predicate logic. Describe existential and universal quantifiers. Assign truth values to quantified statements.	pp.50–51, exercises 1–8, 10–12 and 15–20 p.60, exercises 1–5.
8	Predicate logic	1.3, 1.6	26–28 and 62–74	De Morgan's laws; rules of inference; and rules of inference with quantifiers.	Identify logical equivalence involving quantifiers and apply De Morgan's laws. Apply predicate logic to programming.	p.32, exercises 1–5 pp.74–75, exercises 1–5 and 8–12.
9	Boolean Algebra	12.4, 12.5, 12.6	811–821	the definition of Boolean algebra; Huntington's postulates; De Morgan's theorems; the principle of duality; and algebraic forms.	Write Boolean expressions. Use the laws of Boolean algebra to simplify Boolean expressions. Represent a Boolean function. Simplify logic circuits/expressions. Convert truth tables into Boolean expressions and vice versa.	pp.818–819, exercises 1–5, 20–25 and 35–40.
10	Boolean Algebra	1.2	15–21	the definition of a gate and a circuit; writing Boolean expressions; and circuit simplification (including algebraic and Karnaugh maps).	Write Boolean expressions. Use the laws of Boolean algebra to simplify Boolean expressions. Represent a Boolean function. Simplify logic circuits/expressions. Convert truth tables into Boolean expressions and vice versa.	p.22, exercises 23 and 24.
10					**MID TERM EXAM**	
11	Mathematical Induction	5.1, 5.2	307–325 and 328–336	the definition of mathematical induction; the structure of induction; examples of correct and mistaken proofs by induction; the definition and examples of strong induction; and the definition and examples of the well-ordering property.	State the principle of mathematical induction. Discuss the ideas of the base step and inductive step in a proof by mathematical induction. Apply the ideas of mathematical induction to recursion and recursively defined structures.	pp.325–326, exercises 1, 2, 6, 7, 15 and 16 p.338, exercises 19–21.

Week	Topic	Secions	Essential reading	Topics Covered	Learning Objectives	Exercises
12	Recursion	5.3, 5.4	339–351	the definition of recursion; recursively defined functions; recursively defined sets; and recursively defined algorithms and examples.	Describe the concept of recursion and give examples of its application. Identify the base case and the general case of a recursively defined problem.	p.351, exercises 1–9 and 15–17.
13	Graphs	10.1, 10.2, 10.5	617–625, 627–631, 637–640, 666–676	the definition and application of a graph; degree sequence; simple, regular and complete graphs; and paths.	Define a graph, edges, vertices, parallel edges, loops, cycles and walk, path and connected graphs. Describe the degree sequence of a graph and the relation that links the sum of the degree sequence. Describe special graphs: simple graphs, complete graphs and r-regular graphs.	pp.625–626, exercises 1–3, 18, 19 and 20 pp.641–642, exercises 1–10 and 28–32 pp.676–677, exercises 1–4 and 12–23.
14	Graphs	10.2, 10.3, 10.4, 10.5	632–636, 643–650, 652–663 and 681–688	the adjacency matrix of a graph, isomorphic graphs, bipartite graphs and Dijkstra's algorithm.	Define the adjacency matrix of a graph.	pp.641–642, exercises 17–23 pp.650–652, exercises 1–4, 6–18, 22–24 and 40–44 pp.663–664, exercises 1–3, 7, 16 and 17 pp.688–689, exercises 2–5, 13 and 14.
15	Trees	11.1, 11.4, 11.5	715–717, 753–762 and 764–769	the definition of a tree; spanning trees; weighed graphs; minimum spanning trees; and Prim's and Kruskal's algorithms.	Define a tree. Define spanning trees. Define non-isomorphic spanning trees.	p.725, exercises 1 and 11p.725, exercises 1 and 11 p.763, exercises 1–3, 5 and 6 p.769, exercises 1–5.
16	Trees	11.1, 11.2	717–725 and 726–729	ooted trees and binary search trees.	Define minimum spanning trees. Define rooted trees and binary trees, and find the height of binary trees.	pp.725–726, exercises 2–8, 12, 13, 24 and 26 p.738, exercises 1–4.
17	Relations	9.1	553–561	the definition of a relation; the graph and matrix representation of a relation; and the following relation properties: reflexivity, symmetry, anti-symmetry and transitivity.	Define a relation. Define a relation digraph. Describe the properties of a relation.	p.561, exercises 1–12 p.562, exercises 20–22.
18	Relations	9.3, 9.5, 9.6	570–575, 587–594 and 597–601	equivalence classes; and partial order and total orders.	Define an equivalence relation. Define partial and total order.	pp.575–576, exercises 1–4 and 14–22 pp.594–595, exercises 1–6, 9, 17–23, 25 and 32 p.609, exercises 1, 5–7 and 12–15.
19	Combinatorics	6.1, 6.3	375–385 and 395–401	the basic counting principles; complex counting problems; the pigeonhole principle; the generalised pigeonhole principle; permutations and combinations; examples of strong induction; and the definition and examples of the well-ordering property.	Apply the addition principle and the multiplication principle to count objects when they are sampled with or without replacement.	pp.386–387, exercises 1–5, 7, 8, 14–17, 30, 32 and 40 pp.394–395, exercises 1–6, 14, 15, 19 and 28 pp.402–403, exercises 1–10, 16, 17, 20, 21, 32 and 34.
20	Combinatorics	6.4, 6.5, 6.6	403–409, 410–419 and 422–425	binomial coefficients and identities; Pascal's identity and triangle; and generalised permutations and combinations.	Calculate the number of permutations of length r chosen from a set of n objects. Demonstrate how to use combination formulae to count the number of unordered subsets of r objects taken from a set of n distinct objects. Distinguish between combinations and permutations. Apply the techniques learnt in new counting problems.	pp.409–410, exercises 1–7 and 13–15 pp.419–421, exercises 1–8, 27, 28, 30–32, 44 and 45. p.425, exercises 1–3.
21	Revisions					
22					**F I N A L E X A M**	