DEREE COLLEGE SYLLABUS FOR: ITC 3213 ALGORITHMS AND COMPLEXITY
3/0/3

(Updated Fall 2026)

UK LEVEL: 5
UK CREDITS: 15

PREREQUISITES:	ITC 2088 Introduction to Programming ITC 3006 Mathematics for Computing
CATALOG DESCRIPTION:	Study of algorithms and their complexity. Design, analysis and evaluation of performance. Complexity theory and classes of complexity. O, Big O and Theta notation. Computational classes. Union-Find, Divide and Conquer, Greedy, Dynamic programming, Linear Programming, Search in graphs, NP-completeness.
RATIONALE:	The course aims to acquaint students with the notion of algorithms and complexity, to present generalized techniques for the design and analysis of algorithms for problems that are useful in practice, to argue about the correctness of algorithms, and to become acquainted with the notion of computational classes.
LEARNING OUTCOMES:	As a result of taking this course, the student should be able to: 1. Analyze the characteristics of complexity classes and algorithms' performance in terms of time and space complexity. 2. Evaluate major algorithmic techniques and select the most appropriate for solving problems with consideration of time and space trade-offs. 3. Develop the appropriate algorithms using appropriate data structures to solve computational and graph problems.
METHOD OF TEACHING AND LEARNING:	In congruence with the teaching and learning strategy of the college, the following tools are used: • Lectures, class discussions, use of generative AI tools to inform course content and laboratory practical sessions and problem solving. • Office hours: Students are encouraged to make full use of the office hours of their instructor, where they can ask questions and go over lecture material. • Use of the Blackboard Learning platform, where instructors post lecture notes, assignment instructions, timely announcements, as well as additional resources.
ASSESSMENT:	**Summative:** 1st assessment: **Project** Problem solving — **30%** 2nd assessment: **Portfolio** Collection of in-class activities, quizzes, short tasks completed throughout the semester — **10%** Final assessment: **Final Examination** Problem solving — **60%** **Formative:** programming problems — **0%** The formative assessments aim to prepare students for the summative assessments and expose them to teamwork. The 1st summative assessment tests LO 2. The final summative assessment tests LOs 1-3.

	*The final grade for this module will be determined by averaging all summative assessment grades, based on predetermined weights for each assessment. If students pass the **final summative assessment,** which tests all Learning Outcomes for this module, and the average grade for the module is 40 or above, students are not required to resit any failed assessments.*
INDICATIVE READING:	**REQUIRED READING:** 1. Levitin, A. (2012). *Introduction to the design & analysis of algorithms*. Boston: Pearson. 2. Instructor notes and examples. **RECOMMENDED READING:** 1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*.(4th edition) MIT press. 2. Cormen, T. H. (2009). *Introduction to algorithms*. Cambridge, MA: MIT Press. 3. Dasgupta, S., & Papadimitriou, C. (2008). *Algorithms*. Boston: McGraw-Hill Higher Education. 4. Edmonds, J. (2008). *How to think about algorithms*. Cambridge: Cambridge University Press. 5. Kleinberg, J., & Tardos, E. (2006). *Algorithm design*. Boston: Pearson/Addison-Wesley. 6. Lewis, H., & Papadimitriou, C. *Elements of the theory of computation* (207). Upper Saddle River, N.J.: Prentice-Hall.
INDICATIVE MATERIAL: *(e.g. audiovisual, digital material, etc.)*	**REQUIRED MATERIAL:** N/A **RECOMMENDED MATERIAL:** 1. Coursera, Algorithms Part-1 https://www.coursera.org/course/algs4partI 2. Udacity, Introduction to algorithms https://www.udacity.com/wiki/cs215
COMMUNICATION REQUIREMENTS:	Daily access to the course's site on the College's Blackboard CMS. Effective presentation skills using proper written and oral English. Communicate and coordinate during team activities.
SOFTWARE REQUIREMENTS:	MS-Office, Blackboard CMS, Java, C or Python programming languages
WWW RESOURCES:	• ACM wiki on algorithms and complexity: http://wiki.acm.org/cs2001/index.php?title=Algorithms_and_complexity • Algorithms wiki: http://en.wikibooks.org/wiki/Algorithms • Complexity theory from Wolfram: http://mathworld.wolfram.com/ComplexityTheory.html • Information and Computation journal: http://projects.csail.mit.edu/iandc/ • Journal of Graph Algorithms and Applications: http://www.cs.brown.edu/sites/jgaa/ • ACM journal on experimental Algorithms: http://www.jea.acm.org/about.html

	• ACM Transactions on Algorithms http://talg.acm.org/ • Journal of the ACM http://jacm.acm.org/ • The NP versus P source http://www.win.tue.nl/~gwoegi/P-versus-NP.htm • Algorithms and Complexity resources: http://www.dcs.gla.ac.uk/research/algorithms/links.html
INDICATIVE CONTENT:	Algorithm Analysis and Computational Complexity • The role of algorithms in computing and problem solving • Growth of functions, asymptotic notation, and complexity analysis • Analysis of recursive and non-recursive algorithms, computational tractability, and NP-completeness Algorithm Design Techniques • Brute-force, divide-and-conquer, and greedy approaches • Dynamic programming and optimisation techniques • Space-time trade-offs and efficient algorithm design Data Structures and Advanced Algorithmic Applications • Hash tables and efficient data organisation • Graph algorithms and network-based problem solving • Linear programming and advanced computational problems