Title (Units): **COMP 2230 Design and Analysis of Algorithms (3,3,0)**

Course Aims: The primary objective of this course is to introduce the topic of algorithms as a precise mathematical concept, and study how to design algorithms, establish their correctness, study their efficiency and memory needs. The course consists of a strong mathematical component in addition to the design of various algorithms. By the end of the course, the successful student will be able to:

1. Understand, explain, model, and analyze a given software problem as an algorithm.
2. Investigate whether the algorithm found is the most efficient.
3. Formulate the time order analysis for an algorithm.
4. Formulate the space needs for the implementation of an algorithm.
5. Prove the correctness of an algorithm.

Prerequisite: COMP1210 Data Structures and Algorithms,
MATH1130 Discrete Structures

Learning Outcomes (LOs):
Upon successful completion of this course, students should be able to:

No.	Learning Outcomes (LOs)
	Knowledge
1	Explain the basic concepts of time and space complexity, divide-and-conquer strategy, dynamic programming, greedy and approximate algorithms, amortized analysis and computational geometry
2	Describe the methodologies of how to analyze an algorithm
3	Describe the data structures of red-black tree, B-tree, binomial heap and disjoint sets
4	Identify the complexity of problems
	Professional Skill
5	Solve a problem using an algorithm and evaluate its correctness
6	Formulate the time-complexity analysis for an algorithm
7	Design a better algorithm to solve the problems
	Attitude
8	Build up analyzing, designing and programming skills

Calendar Description: This course builds on the study of the analysis and implementation of algorithms and data structures from COMP1210. The goal is to introduce a number of important algorithms that are interesting both from a practical and theoretical point of view. Algorithm design paradigms such as divide-and-conquer and dynamic programming will be discussed, and algorithms for e.g. sorting, searching, and graph problems will be developed.

Assessment:

No.	Assessment Methods	Weighting	Remarks
1	Continuous Assessment	40%	Continuous assessments are designed to measure how well students have learned the basic concepts and skills of analyzing, designing and implementing an algorithm.
2	Examination	60%	Final Examination questions are designed to evaluate how far students have achieved their intended learning outcomes. Questions will primarily be analysis and skills based to assess the student's ability in the analysis, design and realization of algorithms.

Rubrics:

Excellent (A)	• Achieve the first seven LOs, with strong evidence of having achieved the last LO, demonstrating a good mastery of both the theoretical and practical aspects of the knowledge and skills associated with algorithm design and analysis • Able to develop correct and efficient solutions to problems with the algorithm complexity

	analysis • Demonstrate a thorough understanding and solid knowledge of algorithm design and analysis concepts, and algorithm techniques • Able to apply a variety of algorithm techniques and relevant knowledge for solving new problems
Good (B)	• Achieve most of the first seven LOs, with evidence of having achieved the last LO, demonstrating a good understanding of the knowledge and skills associated with algorithm design and analysis • Able to develop correct solutions to problems with the basic algorithm analysis • Demonstrate a competent level of knowledge of algorithm design and analysis • Ability to make use of appropriate techniques and knowledge and apply them to familiar problems
Satisfactory (C)	• Achieve some of the first seven LOs, demonstrating a basic level of understanding of the knowledge and skills associated with algorithm design and analysis • Able to provide acceptable solutions to problems with the basic knowledge of algorithm complexity analysis • Demonstrate an adequate level of knowledge of algorithm design and analysis • Ability to make use of some techniques and knowledge and apply them to familiar situations
Marginal Pass (D)	• Achieve few of the first seven LOs, with minimal understanding of the associated concepts and underlying methodologies • Able to provide solutions to simple problems • Demonstrate a basic level of knowledge of algorithm design and analysis • Ability to apply some techniques and knowledge to a limited number of typical situations
Fail (F)	• Achieve none of the first seven LOs, with little understanding of the associated concepts and underlying methodologies • Unable to provide solutions to simple problems • Knowledge of algorithm design and analysis falling below the basic minimum level • Unable to apply techniques and knowledge to situations or problems

Learning Outcomes and Weighting:

Content	**LO No.**
I. Introduction and Mathematical Foundations	1, 2
II. Sorting Algorithms	1, 2, 5-6, 8
III. Advanced Data Structures	3, 5, 8
IV. Algorithm Design and Analysis Techniques	1, 2, 5-8
V. NP-Completeness	1, 4, 8
VI. Computational Geometry	1, 5, 8
VII. Approximation Algorithms	1, 5, 8

References:

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 2nd Edition, The MIT Press, 2001.

Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 3rd Edition, Addison Wesley, 2006.

Anany Levitin, Introduction to the Design and Analysis of Algorithms, 2nd Edition, Addison Wesley, 2006.

R.C.T. Lee, S.S. Tseng, R.C. Chang and Y.T. Tsai, Introduction to the Design and Analysis of Algorithms --- A Strategic Approach, McGraw-Hill, 2005.

Jon Kleinberg and Eva Tardos, Algorithm Design, Addison Wesley, 2005.

Course Content in Outline:

Topic

I. Introduction and Mathematical Foundations
- A. Algorithms, Analyzing and Designing
- B. Growth of Functions
- C. Recurrences
- D. Probabilistic Analysis and Randomized Algorithms

II. Sorting Algorithms
- A. Quicksort
- B. Sorting in Linear Time
- C. Medians and Order Statistics

III. Advanced Data Structures
- A. Red-black Trees
- B. B-trees
- C. Binomial Heaps
- D. Disjoint Sets

IV. Algorithm Design and Analysis Techniques
- A. Dynamic Programming
- B. Greedy Algorithms
- C. Amortized Analysis
- D. Divide-and-Conquer

V. NP-Completeness
- A. Polynomial Time
- B. Polynomial-time Verification
- C. NP-completeness and Reducibility
- D. NP-completeness Proofs
- E. NP-complete Problems

VI Computational Geometry
- A. Line-segment Properties
- B. Determining Segments Intersection
- C. Finding Convex Hull
- D. Finding Closest Pair of Points

VII Approximation Algorithms
- A. The Vertex-cover Problem
- B. The Traveling-salesman Problem
- C. The Set-covering Problem