Course Code	Course Title
COMP 3711H	**Honors Design and Analysis of Algorithms**

Course Description

Techniques for designing algorithms, proving their correctness, and analyzing their running times. Topics covered include: sorting, selection, heaps, balanced search trees, divide-and-conquer, greedy algorithms, dynamic programming, and graph algorithms. The class will also provide an introduction to advanced techniques such as amortized analysis and the design of randomized and approximation algorithms, as well as providing exposure to more advanced algorithmic solutions to optimization problems, e.g. linear programming and network flow.

List of Topics

• Sorting Algorithms
• Balanced Binary Search Trees
• Algorithmic Design Techniques
– Divide-and-Conquer
• The Master Theorem
– Greedy Algorithms
– Dynamic Programming
– Randomization
• Graph Algorithms
• Network Flow
• Possible Other topics
– Hashing
– Pattern Matching
– Approximation Algorithms

Textbooks

Algorithms – by Dasgupta, Papadimitriou, and Vazirani – Prepublication version available online

Reference books

- Introduction to Algorithms (3rd ed)
  - Cormen, Leiserson, Rivest and Stein. MIT Press
  - E-version available from the university library
- Programming Pearls (2nd ed)
  - Bentley. Addison Wesley
- Algorithm Design
  - Kleinberg and Tardos. Addison Wesley
- *Problems on Algorithms (*2nd ed)

    - Ian Parberry and William Gasarch (free book)
- Algorithms
    - Jeff Erikson (free book)

Grading Scheme

4 assignments	40%
Midterm exam	30%
Final exam	30%
Total	100%

Course Intended Learning Outcomes

1. Describe fundamental concepts and techniques for determining the asymptotic behavior of real-valued functions defined in natural numbers.
2. Explain recurrence equations and solve common recurrences using a variety of techniques.
3. Analyze an algorithm described in plain language or some form of pseudocode in terms of its time (or space) efficiency as a function of the size of a problem instance.
4. Explain how various data structures, including trees, heaps and disjoint set structures, influence the time efficiency of algorithms.
5. Apply general algorithmic design and analysis techniques to solving problems, including greedy, divide-and-conquer and dynamic programming.
6. Identify randomization in algorithms and analyze basic randomized algorithms such as randomized quicksort and selection.