PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
COLLEGE OF ENGINERING
DEPARTAMENT OF COMPUTER SCIENCE
ABET COURSE SYLLABI

IIC2133 DATA STRUCTURES AND ALGORITHMS

Credits and contact hours: 10 credits / 10 hours (3 h. Lectures; 1,5 h. Assistanship; 3 h. Independent learning experiences; 2,5 h. Group project development)

Instructor's name: Yadran Eterovic

Course coordinator's name Yadran Eterovic

Textbook:

- Goodrich, M; Tamassia, R.; Mount, D. (2011) Data Structures and Algorithms in C++ (2nd ed.). John Wiley & Sons, Inc.
- Cormen, T.; Leiserson, C.; Rivest, R.; Stein, C. (2009) Introduction to Algorithms (3rd ed.), The MIT Press.

Course Catalog Description: This course teaches the fundamental data structures and their algorithms, both in main memory and hard disk. It focuses on their typical usage, advantages and limitations. This course also teaches the main algorithmic techniques to solve discrete optimization problems, emphasizing quantitative analysis of algorithms.

Prerequisite Courses: IIC1253 Discrete mathematics and
IIC2233 Advanced computer programing

Co-requisite Courses: None

Status in the Curriculum: Minimum course

Course Learning Outcomes:

1. To explain the properties of basic data structures: arrays, linked lists, queues, stacks; heaps and priority queues, hash tables, search trees and dictionaries; graphs.
2. To demonstrate the correctness and calculate the performance of the major sorting algorithms, and algorithms for basic data structures.
3. To implement basic data structures and their algorithms.
4. To apply divide-and-conquer, backtracking, greedy algorithms and dynamic programming to solve specific problems.
5. To apply breadth and depth first search, minimum cost spanning trees, and minimum cost path on graphs.

Relation of Course to ABET Criteria:	a. Knowledge of mathematics, science and engineering e. Identify, formulate, and solve engineering problems k. Techniques, skills, and modern tools for engineering practice.

Topics covered:

1. Introduction. Role of algorithms in computing and the relevance of data structures in the design of efficient algorithms.
2. Basic structures. Arrays, linked lists, stacks, queues.
3. Trees. Binary trees; search trees; balanced search trees; splay trees; B-trees.
4. Other structures. Hash tables; priority queues; skip lists.
5. Sorting algorithms. Heapsort; Quicksort; performance analysis; linear time sorting; out of place sorting.
6. Algorithm techniques. Divide-and-conquer; back-tracking; dynamic programming; greedy algorithms.
7. Graphs. Matrix and adjacent list representations; elementary algorithms: breadth and depth search, and topological order; minimum spanning trees; shortest path.
8. Introduction to parallel algorithms. Am-dahl laws, scalability, perfectly parallel algorithms, patterns (divide-and-conquer, map-reduce), and examples (mergeSort).
9. Program representation. Interpreters, compilers, syntax trees, data structures to represent code, memory in execution time.