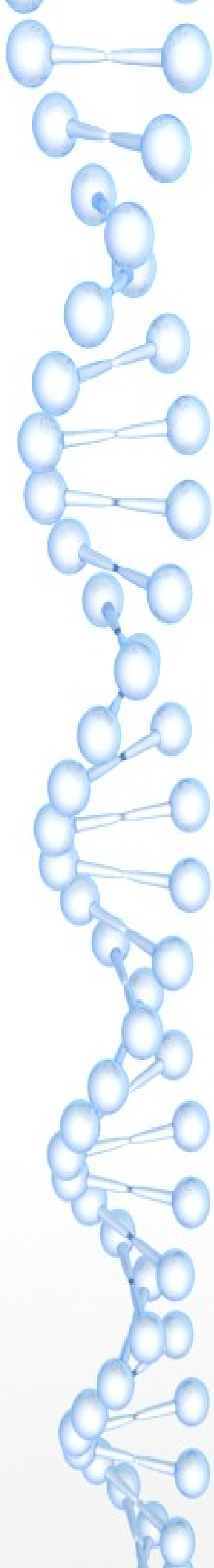

CSC 301: Data Structures and Algorithms

Course Overview

Denis L. Nkweteyim

CSC 301: Data Structures and Algorithms

- Schedule
  - Mondays: 07:00 – 09:00 @ Amphi 150 D
  - Wednesdays: 13:00 – 15:00 @ CRB II, 150A
- Instructors
  - Dr. Denis L. Nkweteyim

Recommended Readings

- Algorithms in C: Fundamentals, data structures, sorting, searching by Sedgewick (1998)
- Algorithms in C: Graph algorithms by Sedgewick (2002)

Expected Pedagogic Activities

- Teaching
  - Lectures: 35 hours
  - Tutorials: 5 hours
  - Practicals: 20 hours
- Largely practical course
  - Student should spend significantly more time outside class hours than indicated above, doing practical work (writing and understanding code)
  - Tutorials are designed to reinforce the lectures
    - Will be given at appropriate times in the course of the semester
- No substitute for attending class
  - Understanding concepts presented will be essential
  - Simply trying to cram the material will not help

Evaluation

- CA - 30 %
- Semester Exam: 70 %
- Note
  - Both CA and semester exam will be highly biased towards programming

Some advice

- Don'ts
  - Skip lectures & tutorials
  - Cram close to exam time
  - Expect the exam to be a memory test
- Dos
  - Attend lectures & tutorials
  - Practice the skills you learn immediately
  - Expect a problem-solving exam

Course overview

- Algorithm
  - A problem-solving method
  - Implemented in a computer as a program
  - Generally
    - Independent of the machines on which they are implemented
    - Independent on a particular compiler or programming language

Course overview

- Data structure
  - Most algorithms involve methods of organizing the data involved in the computation
  - Data objects created in this way are called data structures

Course overview

- Problem-solving
  - Programmer chooses from a number of different algorithms
  - For small problems
    - Does not matter much which approach is used
      - Gain in time if the most efficient algorithm is chosen might be negligible
  - For large complex problems
    - Need to choose algorithms that manage space (memory) and time well
      - Carefully designed algorithm may reduce computation time by a factor of thousands, if not millions of times over a poorly designed one
  - Well-designed algorithms pay off far better than more sophisticated hardware (faster processors, larger memories)
    - Improvement factor of thousand or even millions vs improvement factor of about 10 – 100

Course overview

- Choice of which algorithm to use can be complicated, involving sophisticated mathematical analysis
  - Course examines algorithms and data structures
    - Algorithms
      - Basic methods to analyze algorithms to determine their efficiencies
      - Consideration of a number of standard algorithms
    - Data structures
      - Study of a range of data structures
        - Arrays, linked lists, strings, recursive data structures, and abstract data types like stacks and queues, and
      - Algorithms that make use of these data structures
        - Sorting, searching, tree and graph algorithms
    - Concrete implementations of data structures and algorithms in the C programming language