Data Structures
CS 260-002 Spring 2002

Course Description and Syllabus
Instructor: Krzysztof Nowak
Office: Korman 297
Office Hours: Monday, Wednesday, Friday 11:00 am - 11:50 am (or by an appointment)
e-mail: knowak@mcs.drexel.edu

Section 002: Monday, Wednesday, Friday 10:00 am - 10:50 am Matheson 309
Prerequisites: CS 171, CS 172

Data Structures is a Computer Science domain, which deals with the organization of data. Modern programming makes use of the following basic structures of data representation: linked lists, stacks, queues, trees and graphs. The course deals with implementations of these data structures and with algorithms, which allow one to operate efficiently on them. We cover theoretical concepts and methods, which are necessary in the process of design and evaluation of algorithms. Our presentation of the topic will follow the steps:

- Abstract understanding of data types
- Design and implementation
- Analysis of correctness, flexibility and efficiency
- Discussion of applications
One of the learning objectives of the course is to encourage the students to use pictorial representations of data structures and algorithms. The method of graphical display of the steps of an algorithm is particularly useful in the case of traversal algorithms based on recursion. Homework will be assigned on weekly basis. It will consist mostly of programming problems and tasks related to analysis of algorithms. Two midterms and the final exam are planned. The prerequisites for the course are computer-programming courses CS 171, CS 172.

Course Objectives

- Understanding theoretical concepts behind basic data types
- Getting familiar with specific implementations of data structures and learning about their advantages/disadvantages
- Gaining perspective on C++ language and its approach to object oriented programming
- Learning various applications and algorithms related to basic data structures

Required Text
Michael Main, Walter Savitch, *Data Structures and Other Objects Using C++*  
Addison-Wesley, 2000 (Second Edition)

Syllabus
Week 1: An overview of basic features of C++
Week 2: Pointers, dereferencing operator, address operator, pointers as value and reference parameters, dynamic arrays, operators new and delete
Week 3: Linked lists: the structure of nodes, the member selection operator, building and manipulating linked lists
Week 4: Review 1, Midterm 1, template functions and classes
Week 5: Standard Template Library of C++, algorithms based on recursion
Week 6: Trees: representations, implementations & algorithms
Week 7: Stacks, queues and related algorithms
Week 8: Review 2, Midterm 2, searching
Week 9: Sorting
Week 10: Hashing, Final Review

Course Grade and Exams
There will be two midterm exams and the final. The final exam will be comprehensive. The final grade will be computed as follows:
Midterm Exams 35%
Final Exam 30%
Homework/Programming Assignments 30%
Attendance 5%

Midterm 1: 04/17/02
Midterm 2: 05/15/02