

CS126 Design of Information Structures 2003-04

Module Organisers

Steve Russ	CS 317	sbr@dcs
Andrei Krokhin	CS 230	ak@dcs
Tim Heron	CS 316	theron@dcs
Zabin Visram	CS 115	zabin@dcs

Overall Aims

- (i) to gain familiarity with the *specification*, *implementation* and *use* of some abstract data types (ADTs);
- (ii) to introduce some useful and standard ADTs such as linked lists, stacks, queues, binary trees, graphs, sets etc;
- (iii) to learn some standard algorithms for common tasks (such as searching and sorting) and some elementary methods of measuring the complexity, and of showing the correctness, of algorithms.

General Approach

- The main emphasis of the course is on developing a broader view of programming: the challenges and the technical resources for developing useful software.
- The language being used is Java, which offers good support for ADTs through its class libraries and principles. A basic competence in Java is assumed. A critical appreciation of OOP is encouraged.
- The course is fairly equally balanced between theoretical aspects (e.g. principles behind the representation and design of ADTs) and practical aspects such as the construction, documentation and testing of programs using Java tools.

Specific Objectives

- You should become familiar with a range of standard ADTs and algorithms and how they can be used to accomplish common programming tasks.
- You should be able to assess the complexity and correctness of simple algorithms, and choose appropriate algorithms for given tasks.
- You should have practical experience of designing some user-defined ADTs, and associated algorithms, for a non-standard application.

Course Content

Concept of the abstract data type (ADT)

Specification and implementation of ADTs

Standard examples of ADTs and algorithms

Elementary complexity and correctness of algorithms

Applications of ADTs to program development

Assessment

50% by coursework (10% of this in Lab sessions)

50% by 1.5 hour examination

Books

David A Watt & Deryck F Brown, *Java Collections*, Wiley, 2001 (www.wiley.co.uk/wattbrown)

**Duane Bailey, *Java Structures*, McGraw-Hill, 2nd ed. 2003
(www.cs.williams.edu/JavaStructures)**

William J Collins, *Data Structures and the Java Collections Framework*, McGraw-Hill, 2002

Michael Goodrich & Roberto Tamassia, *Data Structures and Algorithms in Java*, John Wiley, 2nd ed. 2001

Mark Allen Weiss, *Data Structures and Problem Solving using Java*. Addison-Wesley, 2nd ed. 2002

Course web page (for Laboratory sessions):

<http://www.dcs.warwick.ac.uk/~zabin/dis.html>

News group:

uwarwick.dcs.course.cs126