

Module Guide

Undergraduate Programme Academic Year 2009/2010

Module:	Data Structures and Algorithms UG2
Web-site:	http://bcu.copsewood.net/
School:	Computing, Telecommunications and Networks
Module Co-ordinator:	Richard Kay
Module Tutors:	Richard Kay
Contact Information:	Richard.Kay@bcu.ac.uk
Brief Descriptions of the Items of Assessment: You will be expected to complete ALL Assessments.	<p>1. Assessment 1 comprises 2 separate programming design and testing exercises with separate deliverables.</p> <p>2. Assessment 2 comprises a closed book written exam.</p> <p>Information is for guidance only. See ECMS My Course on the intranet for details.</p>
Assessment Weighting:	See ECMS My Course on the intranet for details
<p>Individual assignments. The work you submit shall be your own and not the product of collaboration with anyone else. Plagiarism will be penalised.</p> <p>In-course assessments shall be submitted through the Coursework Collection System, to the module co-ordinator.</p>	
Contents of Guide:	
Syllabus and supporting information	Teaching schedule

Syllabus and supporting information

Recommended reading

Berry, Meekings & Soren, (1998) A Book on C, MacMillan

Kelley & Pohl, (1998) A Book on C, 4th ed., Addison-Wesley

Kernighan & Ritchie, (1998) The C Programming Language, Prentice-Hall

Standish, M (1995) Data Structures, Algorithms and Software Principles in C, Addison-Wesley

Weiss, M (1997) Data Structures and Algorithm Analysis in C, 2nd ed., Addison-Wesley

Aims and Objectives of Module

To further develop understanding and skills of writing software programs.

To introduce students to the concepts of data structures and algorithms.

To impart an understanding of data design and manipulation at various levels.

To develop in-depth understanding and skills in writing software programs.

To extend previous experience to implement advanced data structures and algorithms

Brief Module Description

Advanced language features: Arrays, structures, pointers, type casting, file I/O, dynamic memory allocation.

Bit level programming: Bit operators, masking, logical operators

Data Structures: An appropriate comparison and analysis of data structures including arrays, linked lists, queues, stacks, hash tables, trees and graphs.

Algorithms: An appropriate selection of algorithms including recursion, iteration, sorting, searching, record insertion/deletion.

Complex Algorithms: An appropriate selection of algorithms such hashing, searching, balancing, and traversing trees, minimum spanning trees.

Algorithm Analysis: Comparison of various sort algorithms, searching, insertion, deletion and balancing of records stored within various data structures.

Required Language and Recommended Tools

All programs submitted for module coursework must be written in ANSI Standard 'C' (e.g. as described in Kernighan and Ritchie 2e). Use of proprietary libraries not included in this standard (e.g. conio.h) should be avoided.

A suitable and free (price and freedom) programming development tool for this purpose is Dev C++ version 4. See <http://www.bloodshed.net/devcpp.html> .

Teaching Schedule for: **Data Structures and Algorithms UG2**

Wk No	Date (Mon)	Lecturer	Lecture Topic	Tutorial / Lab Topic	Assignment *	
					Set	Due In
1	28-Sep-09	RK	Structures with functions pointers and arrays	Functionality of database binary	A1.1	
2	05-Oct-09		Bitwise operators	Design documentation		
3	12-Oct-09		Dynamic memory allocation	Test planning and documentation		
4	19-Oct-09		File organisation strategies	Multi-source file projects organisation		
5	26-Oct-09		Parsing and validating text input	File input and output exercises		
6	02-Nov-09		Keys, insertion, modification and deletion	Memory sizing, allocation and copying		
7	09-Nov-09		Linear and binary searching	Work on assignment 1.1		
8	16-Nov-09		Slow but simple iterative sorting algorithms	Work on assignment 1.1		
9	23-Nov-09		Assignment Completion			
10	30-Nov-09		Stacks and queues	Work on assignment 1.1		
11	07-Dec-09		Linked lists	Work on assignment 1.1		
12	14-Dec-09		In class formative test	Introducing assignment 1.2	A1.2	A1.1
Christmas Vacation (3 weeks)						
13	11-Jan-10		Pointers to pointers	Linklist functions and multilevel pointers		
14	18-Jan-10		Recursion	Work computing fibonnacci numbers		
15	25-Jan-10		Fast recursive sorting	Library qsort and counting comparisons		
16	01-Feb-10		Binary trees	Running knuth binary tree program		
17	08-Feb-10		AVL balanced trees	Work on assignment 1.2		
18	15-Feb-10		Assignment Completion			
19	22-Feb-10		Heaps and the heapsort	Work on assignment 1.2		
20	01-Mar-10		Hash functions and hash tables. Bucket sort.	Work on assignment 1.2		
21	08-Mar-10		Pointers to functions and code reuse	Work on assignment 1.2		
22	15-Mar-10		Random numbers and Monte Carlo methods	Work on assignment 1.2		
23	22-Mar-10		Introduction to graph (networks) theory	Work on assignment 1.2		A1.2
Easter Vacation (3 weeks)						

Wk No	Date (Mon)	Lecturer	Lecture Topic	Tutorial / Lab Topic	Assignment	
					Set	Due In
24	19-Apr-10	RK	A graph-based route-planning program	Programming exercises for exam revision		
25	26-Apr-10		Geometrical algorithms and numerical stability			
26	03-May-10		Final week of teaching (revision classes) (Monday – Bank Holiday)			
27	10-May-10		Exams		A2	
28	17-May-10		Exams			
29	24-May-10					

* Assignment Set and Due week indication above is for guidance only. See ECMS My Course on the intranet for details.