ITC 3387 Data Structures and Analysis of Algorithms

DEREE COLLEGE SYLLABUS FOR:

ITC 3387 DATA STRUCTURES AND ANALYSIS OF ALGORITHMS – LEVEL 5
(Updated Winter 2011)  UK Credits: 15

PREREQUISITES:  CS 1070 Introduction to Information Systems
                 CS 2188 Introduction to Programming
                 MA 1001 Finite Mathematics

CATALOGUE DESCRIPTION:
Algorithmic design; interaction between algorithm and data structure in creating efficient code. Common types of algorithms and data structures; data structures usage and implementation. Lists, stacks, queues, hash tables and trees. Algorithmic mechanisms and problem solving techniques.

RATIONALE:
This module introduces the theory and practice of problem solving in computing through the development of algorithms for common computer science problems, and to use this in the wider context of problem solving and software development. It presents a variety of algorithmic techniques and aims to provide an understanding of the use and importance of data structures. It introduces the idea of classifying data according to its abstract behaviour, as distinct from its representation. A range of well-established data types are examined and their properties are described so that it becomes clear which representation are appropriate under which circumstances. An understanding of the basic skills needed in algorithmic design and the interaction between algorithm and data structure in creating efficient code is emphasised.

LEARNING OUTCOMES: On completion of the module students should be able to:
1. Recognise different types of data structures and know their advantages and disadvantages.
2. Differentiate between various algorithmic techniques such as iteration and recursion.
3. Appreciate the resource requirements of various algorithms and data structures.
4. Identify the most appropriate data structure and algorithm for a range of applications.
5. Implement efficiently various common data structures and algorithms.

METHOD OF TEACHING AND LEARNING: In congruence with the learning and teaching strategy of the College, the following tools/activities are used:
- Lectures, class discussions, and programming practice in problem solving using data structures & algorithms.
- Office hours held by the instructor to provide further assistance to students.
- Use of the Blackboard site to further support communication, by posting lecture notes, assignment instruction, timely announcements, and online submission of assignments, but also as a live example of e-learning applications.
ITC 3387 Data Structures and Analysis of Algorithms

ASSESSMENT:

<table>
<thead>
<tr>
<th>Summative:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Coursework: programming problems</td>
<td>40</td>
</tr>
<tr>
<td>Final Examination (2-hour, comprehensive): programming problems</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Formative:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Take-home “diagnostic” test:</td>
<td>0</td>
</tr>
<tr>
<td>programming problems</td>
<td></td>
</tr>
</tbody>
</table>

The diagnostic test aims to shape teaching along the semester and prepare students for the final examination.
The coursework tests Learning Outcomes 1-3.
The final examination tests Learning Outcomes 1-5.
(Guidelines and assessment rubrics are distributed on the first day of classes along with the Course Outline.)

INDICATIVE READING: REQUIRED READING:


RECOMMENDED READING:

- Laboratory notes, exercises and on-line tutorials (College Blackboard)

COMMUNICATION REQUIREMENTS:

Daily access to the course’s site on the College’s Blackboard CMS.

SOFTWARE REQUIREMENTS:

Java, C++

WWW RESOURCES:

www.computer.org/proceedings/words/0101/01010239abs.htm
www.nist.gov/dads
cips.ee.uwa.edu.au/~morris/Year2/PLDS210/ds_ToC.html

INDICATIVE CONTENT:

1. Motivation for implementing data structures.
2. Solving problems through the construction of algorithms.
3. Fundamental data structures.
4. Algorithmic mechanisms such as sequence, iteration, choice and recursion.
5. Tree structures and algorithms analysis.
8. Text processing.

10. Strategies for problem solving such as divide and conquer, branch and bound greedy algorithms.