<table>
<thead>
<tr>
<th><strong>Module Code</strong></th>
<th>CS2010</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module Name</strong></td>
<td>Algorithms and Data Structures</td>
</tr>
<tr>
<td><strong>Module Short Title</strong></td>
<td>CS2010</td>
</tr>
<tr>
<td><strong>ECTS weighting</strong></td>
<td>10</td>
</tr>
<tr>
<td><strong>Semester/term taught</strong></td>
<td>Michaelmas &amp; Hillary Term</td>
</tr>
</tbody>
</table>
| **Contact Hours** | Lecture & Tutorial hours: 33  
Lab hours with lecturer and demonstrators per lab group: 11  
Total hours: 44 |
| **Module Personnel** | Dr Hugh Gibbons (Hillary Term)  
Dr Vasileios Koutavas (Michelmas Term) |
| **Learning Outcomes** | Having successfully completed this module students will be able to:  
- Evaluate algorithms in terms of their running time and memory space requirements and classify those algorithms in the major complexity classes.  
- Efficiently implement the operations of core data structures.  
- Implement effective algorithms.  
- Identify the most suitable data structures and algorithms for each programming problem based on the parameters of the problem, the resources available, the desired performance criteria etc.  
- Design and implement more robust, effective and well-structured Java programs using Abstract Data Types and the approaches of unit testing, test coverage, Design by Contract, and pre-/post-conditions. Students will be able to use the last two approaches to avoid defensive programming. |
| **Module Learning Aims** | The aim of the module is threefold:  
1. To teach effective programming and problem solving, using a core toolset of classical algorithms and data structures.  
2. To introduce the methods for evaluating the performance and requirements of programs written by the students  
3. To promote effective software engineering by using well-established techniques for code modularity, structuring, debugging and readability, such as Design by Contract, and unit testing. |
| **Module Content** | **Theory:**  
- Asymptotic growth functions and analysis of source code to derive running time and space requirements  
- Amortised running time analysis of algorithms  
- Permutations, Combinations and Sets  
**Data structures:**  
- Array and linked list implementations of stacks and queues.  
- Doubly linked lists  
- Union-find  
- Binary trees, binary search trees, balanced search trees, B-trees  
- Hash tables  
- Undirected, directed and weighted graph implementations using adjacency lists  
- Special topics |
 Algorithms:
• Recursion vs iteration; tree traversals
• Greedy algorithms
• Divide and conquer
• Graph algorithms
• Searching and Sorting algorithms
• Special topics
 Programming:
• Java generics
• Iterators
• JUnit testing
• Design by Contract

Recommended Reading List
Main textbook:
Robert Sedgewick and Kevin Wayne
http://algs4.cs.princeton.edu/home/
Pearson Education 2011

Other recommended textbooks:
Introduction to Programming Using Java (6th Edition)
David J. Eck
http://math.hws.edu/javanotes/

Introduction to Algorithms (3rd Edition)
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein
http://mitpress.mit.edu/books/introduction-algorithms
MIT Press 2009

Algorithms and Data Structures
Nicolas Wirth
Prentice-Hall 1986

Data Structures and Algorithms
John E. Hopcroft and Jeffrey D. Ullman.
Addison-Wesley 1983

Module Pre Requisite
An introductory course on programming with Java. CS1011 and CS1012

Module Co Requisite

Assessment Details
35% Coursework, 65% Exam

Assessment in the annual examinations is by written exam’ (65%) and coursework (35%).
Assessment in the supplementals is by 100% written examination only.

Module approval date

Approved By

Academic Start Year

Page 2 of 3
Module Descriptor 2014/15
School of Computer Science and Statistics.

<table>
<thead>
<tr>
<th>Academic Year of Data</th>
</tr>
</thead>
</table>
