

Module Descriptor 2016/17  
School of Computer Science and Statistics.

<b>Module Code</b>	CS2010
<b>Module Name</b>	Algorithms and Data Structures
<b>Module Short Title</b>	CS2010
<b>ECTS weighting</b>	10
<b>Semester/term taught</b>	Michaelmas & Hillary Term
<b>Contact Hours</b>	Lecture & Tutorial hours: 33 Lab hours with lecturer and demonstrators per lab group: 11 Total hours: 44
<b>Module Personnel</b>	Dr Vasileios Koutavas (Michaelmas Term) Dr Ivana Dusparic (Hillary Term)
<b>Learning Outcomes</b>	Having successfully completed this module students will: <ol style="list-style-type: none"> <li>1. Have gained significant knowledge on algorithms and data structures, and the mathematical theory and techniques to evaluate their efficiency and effectiveness.</li> <li>2. Have the ability to evaluate algorithms in terms of their running time and memory space requirements and classify those algorithms in the major complexity classes using appropriate performance models.</li> <li>3. Be able to efficiently implement the operations of the main data structures used in most programming.</li> <li>4. Gain experience through experiments in implementing effective new and existing algorithms.</li> <li>5. Be able to identify the most suitable data structures and algorithms for each programming problem based on the parameters of the problem, the advantages and limitations of each data structure and algorithm, the resources available, the desired performance criteria etc.</li> <li>6. Be able to design and implement robust, effective and well-structured Java programs using industry standards such as 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.</li> </ol>
<b>Module Learning Aims</b>	The aim of the module is threefold: <ol style="list-style-type: none"> <li>1. To teach effective programming and problem solving, using a core toolset of classical algorithms and data structures.</li> <li>2. To introduce the methods for evaluating the performance and requirements of programs written by the students</li> <li>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.</li> </ol>
<b>Module Content</b>	<p>Theory:</p> <ul style="list-style-type: none"> <li>• Asymptotic growth functions and analysis of source code to derive running time and space requirements</li> <li>• Amortised running time analysis of algorithms</li> <li>• Permutations, Combinations and Sets</li> </ul> <p>Data structures:</p> <ul style="list-style-type: none"> <li>• Array and linked list implementations of stacks and queues.</li> <li>• Doubly linked lists</li> <li>• Union-find</li> <li>• Binary trees, binary search trees, balanced search trees, B-trees</li> <li>• Hash tables</li> <li>• Undirected, directed and weighted graph implementations using adjacency lists</li> <li>• Special topics</li> </ul> <p>Algorithms:</p>

Module Descriptor 2016/17  
School of Computer Science and Statistics.

	<ul style="list-style-type: none"> <li>• Recursion vs iteration; tree traversals</li> <li>• Greedy algorithms</li> <li>• Divide and conquer</li> <li>• Graph algorithms</li> <li>• Searching and Sorting algorithms</li> <li>• Special topics</li> </ul> <p>Programming:</p> <ul style="list-style-type: none"> <li>• Java generics</li> <li>• Iterators</li> <li>• JUnit testing</li> <li>• Design by Contract</li> </ul>
<b>Recommended Reading List</b>	<p><b>Main textbook:</b>  <i>Algorithms (4th Edition)</i>            Robert Sedgewick and Kevin Wayne  <a href="http://algs4.cs.princeton.edu/home/">http://algs4.cs.princeton.edu/home/</a>            Pearson Education 2011</p> <p><b>Other recommended textbooks:</b>  <i>Introduction to Programming Using Java (6th Edition)</i>            David J. Eck  <a href="http://math.hws.edu/javanotes/">http://math.hws.edu/javanotes/</a></p> <p><i>Introduction to Algorithms (3rd Edition)</i>            Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein  <a href="http://mitpress.mit.edu/books/introduction-algorithms">http://mitpress.mit.edu/books/introduction-algorithms</a>            MIT Press 2009</p> <p><i>Algorithms and Data Structures</i>            Nicolas Wirth            Prentice-Hall 1986</p> <p><i>Data Structures and Algorithms</i>            John E. Hopcroft and Jeffrey D. Ullman.            Addison-Wesley 1983</p>
<b>Module Pre Requisite</b>	An introductory course on programming with Java. CS1011 and CS1012
<b>Module Co Requisite</b>	N/A
<b>Assessment Details</b>	<p>35% Coursework, 65% Exam</p> <p><i>Assessment in the annual examinations is by written exam' (65%) and coursework (35%).            Assessment in the supplementals is by 100% written examination only.</i></p>
<b>Module approval date</b>	N/A
<b>Approved By</b>	N/A
<b>Academic Start Year</b>	N/A

Module Descriptor 2016/17  
School of Computer Science and Statistics.

Academic Year of Data	N/A
--------------------------	-----