THE UNIVERSITY OF BUCKINGHAM

Sarajevo School of Science and Technology

MODULE SPECIFICATION

Name of Module		Data Structures and Algorithms					
Parent School/Dept		**Computer Science**					

Programme(s) where module is offered		BSc Computer Science with Electrical Engineering;
		BSc Computer Science with Economics;
		BSc Computer Science with Business;
		BSc Computer Science with Political Science;					

Status (core, option, free choice)		Core		**Pre-Requisite Modules or Qualifications**		MATH180, CSIS120	
FHEQ Level	4	**Unit Value**	6 ECTS	**Module Code**	CSIS 250	**Module coordinator**	Dr. Jasminka Hasic Telalovic
Semester taught		Spring		**Applicable From**		2019	

Educational Aims of the Module

This module introduces students to fundamental data structures and basic performance measures for these data structures, together with analysis and design of computer algorithms. The module teaches both the theoretical and programming underpinnings of the data structures as well as main applications and usage scenarios. The module emphasizes the connections between the mathematical analysis and programming aspects of data structures and how they together affect the performance. For each data structure, students learn to distinguish their performance with respect to the search/insert/delete performance as well as the space requirements and the implementation complexity. Later on the students are introduced to algorithms, their asymptotic performance, apply important algorithmic paradigms to problems at hand and use algorithms in real-life engineering situations. The module introduces students to basic problem-solving paradigms and teaches about flavors of problems solved by a particular paradigm. The module emphasizes design over implementation of algorithms and focuses on asymptotically efficient solutions to given problems. Throughout the module, important real-life examples are given of problems solved using particular paradigms. The main expected outcome of the class is when presented with a computational problem, students are able to suggest the right data structure to use to solve it. Throughout their assignments, students learn to use data structures and algorithms in different scenarios.

Module Outline/Syllabus

- Introduction to data structures, Asymptotic notation
- Arrays, Resizing arrays, Matrices, Stressen,
- Sorting algorithms and their complexity: Bubble sort, Selection sort, Insertion sort
- Stacks, Queues
- Linked Lists, Skip Lists
- Recursion, Master Method, Fibonacci
- Trees: Introduction, Binary Search trees
- Binary search trees
- Heaps, Priority queues
- Hashing
- Graphs
- String data structures
- Divide and conquer algorithms
- Linear time-sorting algorithms
- Greedy Algorithms
- Minimum Spanning Trees
- Shorthest Paths
- Dynamic Programming
- NP-Completeness

Student Engagement Hours

Type	Number per Term	Duration	Total Time
Lectures	30	2 hours	60 hours
Laboratory sessions	15	2 hours	30 hours
Total Guided/Independent Learning Hours			**60**
Total Contact Hours			**90**
Total Engagement Hours			**150**

Assessment Method Summary

Type	Number Required	Duration / Length	Weighting	Timing/Submission Deadline
Assignment (Programming challenge/homework)	5	800 words	30%	Weeks 2, 4, 6, 8, 12
Mid-term exam	1	90 minutes	20%	Week 9
Final Exam	1	180 minutes	50%	End of semester

Module Outcomes

Intended Learning Outcomes: 1. Understanding a number of the basic and advanced data structures and algorithms related issues such as efficiency, usability, complexity, etc 2. Understanding the mathematical and programming concerns when implementing and using algorithms and appropriate data structure 3. Effective problem-solving using presented algorithms 4. Understanding relevant data structure issues and current research trends	→	**Teaching and Learning Strategy:** 1. Laboratory sessions (ILO:1-4) 2. Lectures delivered containing the material from the module outline (ILO:1-4) 3. Regular presentation of solutions with peer feedback and discussion are encouraged both during lecture time and especially during lab time (ILO:1-4) 4. Programming challenge (ILO:1-4)
	→	**Assessment Strategy** 1. Programming Challenge (ILO:1-4) 2. Mid-term exam (ILO:1, 3) 3. Final exam (ILO:1-4) 4. Assignment (ILO:1-4)
Practical Skills 1. Enhanced programming skills 2. Ability to analyse a given practical problem and suggest the right data structure based on the required insert/search performance 3. Ability to analyse the time and space constraints for a given application 4. Present a solution to a technical problem in a natural language	→	**Teaching and Learning Strategy:** 1. Assignment (PS: 1-3) 2. Programming challenge (PS: 1-4) 3. Reading and in class practice (PS: 1-3)
	→	**Assessment Strategy** 1. Mid-term exam (PS:1-2) 2. Final exam (PS:1-3) 3. Assignment (PS:1-3) 4. Programming challenge (PS:1-4)
Transferable Skills 1. Communication skills 2. Presentation skills 3. Computer literacy	→	**Teaching and Learning Strategy:** 1. In-class communication (TS: 1-3) 2. Reading and exercises during tutorial sessions (TS: 1-3) 3. Reading and in class practice (TS: 1-3)
	→	**Assessment Strategy** 1. Programming challenge (TS: 1-3) 2. Assignment (TS:1-3)

Key Texts and/or other learning materials

Set Text

Karumanchi, N., 2011, Data Structures and Algorithms Made Easy: Data Structure and Algorithmic Puzzles, 2nd Edition and Cormen, T.,Leiserson,C., Rivest, R., Stein,C., (2009), Introduction to Algorithms, 3rd Edition. MIT Press

Supplementary Materials

- Lafore, R., 2002, *Data Structures and Algorithms in Java,* 2nd Edition, SAMS
- Goodrich & Tamassia, 2014, *Data Structures and Algorithms in Java* , 6th Edition, John Wiley & sons
- Mark Allen Weiss, 1996, *Data Structures and Algorithm Analysis in C*, 2nd Edition, Addison -Wesley
- Mark Allen Weiss, 2012, *Data Structures and Algorithm Analysis in Java* 3rd Edition, Edition, Addison-Wesley.
- David Flanagan, 2014, *Java in a Nutshell,* 6th Edition, O'Reilly
- Bruce Eckel, 2006,*Thinking in Java,* Prentice-Hall PTR.
- Kathy Sierra and Bert Bates, 2005, *Head First Java*, 2nd Edition, O'Reilly
- Bruno Preiss, 1999, *Data structures and Algorithms with Object Oriented Design Patterns in Java*, John Wiley & Sons.
- Clifford Shaffer*,* 2011, *Data Structures and Algorithm Analysis*,3rd Edition, Dover Publications
- Sartaj Sahni, 2004, *Data Structures, Algorithms, and Applications in Java*, 2nd Edition, Silicon Press Alfred Aho, 1983, John Hopcroft, and Jeffrey Ullman, *Data Structures and Algorithms*, Addison-Wesley
- MIT Algorithms and Data Structures Podcasts [online], https://itunes.apple.com/us/itunes-u/introduction-to-algorithms/id341597754?mt=10 (Accessed 6th June 2016).

Skiena, S., (2010). Algorithm Design Manual, 2nd edition, Springer

Dasqupta, S., Papadimitriou, C., Vazirani, U., (2006), *Algorithms,* McGraw-Hill

Kleinberg, J., Tardos, E., (2013), *Algorithm Design,* Pearson

Algorithms, (2015) Open Access Journal [online], http://www.mdpi.com/journal/algorithms (Accessed 6th June 2016)

- Elsevier, (2016), Journal of Discrete Algorithms [online], http://www.journals.elsevier.com/journal-of-discrete- algorithms/open-archive (Accessed 6th June 2016)

Please note: This specification provides a concise summary of the main features of the module and the learning outcomes that a typical student might reasonably be expected to achieve and demonstrate if he/she takes full advantage of the learning opportunities that are provided. More detailed information on the learning outcomes, content and teaching, learning and assessment methods of each module and programme can be found in the departmental or programme handbook. The accuracy of the information contained in this document is reviewed annually by the University of Buckingham and may be checked by the Quality Assurance Agency.

Date of Production	Spring 2019
Date approved by School Learning and Teaching Committee	
Date approved by School Board of Study	
Date approved by University Learning and Teaching Committee	
Date of Annual Review	