H8ADVPROG: Algorithms and Advanced Programming

Module Code:	H8ADVPROG
Long Title	Algorithms and Advanced Programming APPROVED
Title	Algorithms and Advanced Programming
Module Level:	LEVEL 8
EQF Level:	6
EHEA Level:	First Cycle
Credits:	5
Module Coordinator:	Adriana Chis
Module Author:	Adriana Chis
Departments:	School of Computing
Specifications of the qualifications and experience required of staff	MSc and/or PhD degree in computer science or cognate discipline. May also have industry experience.

Learning Outcomes

On successful completion of this module the learner will be able to:

#	Learning Outcome Description
LO1	Use iterative and recursive techniques to design and implement sorting and searching algorithms.
LO2	Conduct in depth algorithm analysis in terms of time complexity and present the results of analysis.
LO3	Evaluate algorithms, identify an algorithm from a range of possible options, and implement the algorithm to solve computational problems in particular contexts.
LO4	Identify and apply best practices including exception handling and design patterns in the implementation of software solutions to solve real-world problems.

Dependencies

Module Recommendations

No recommendations listed

Co-requisite Modules

No Co-requisite modules listed

Entry requirements	

H8ADVPROG: Algorithms and Advanced Programming

Module Content & Assessment

Indicative Content
Algorithms • Algorithm design and development • Properties of algorithms (e.g. correctness, generality) • Empirical measurements of algorithm's performance • Time complexity, asymptotic notations (e.g. Big O Notation, Big Omega Notation)
Sorting Algorithms • Bubble sort • Insertion sort • Quicksort • Mergesort • Perfomance comparison of sorting algorithms
Searching Algorithms • Sequential search • Binary search (iterative and recursive implementations)
Defensive programming • Input validation and data sanitization • Exception handling (general exception handling, declaring new exception types)
Multithreaded Programming • Concurrent execution of threads (multitasking) • User created threads • Thread priorities • Thread states and lifecycle
File I/O • Low-level file I/O • High-level file I/O

Assessment Breakdown	%
Coursework	50.00%
End of Module Assessment	50.00%

Assessments

Full Time

Coursework

Assessment Type:	Continuous Assessment	**% of total:**	50
Assessment Date:	n/a	**Outcome addressed:**	1,2,3,4
Non-Marked:	No		

Assessment Description:
The continuous assessment will consist of in-class practical tests. The practical assessments aim to evaluate students' knowledge and ability to identify, analyse, implement and use different algorithms, and best practices to solve computational problems. Students will be assessed both on their development skills and their ability to convey understanding of the programs they have developed

End of Module Assessment

Assessment Type:	Terminal Exam	**% of total:**	50
Assessment Date:	End-of-Semester	**Outcome addressed:**	1,2,3,4
Non-Marked:	No		

Assessment Description:
End-of-Semester Final Examination

No Workplace Assessment

Reassessment Requirement

Repeat examination
Reassessment of this module will consist of a repeat examination. It is possible that there will also be a requirement to be reassessed in a coursework element.

Reassessment Description
Reassessment of this module will be via repeat examination which evaluates all learning outcomes.

H8ADVPROG: Algorithms and Advanced Programming

Module Workload

Module Target Workload Hours 0 Hours

Workload: Full Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	No Description	24	Per Semester	2.00
Laboratories	No Description	24	Per Semester	2.00
Independent Learning Time	No Description	77	Once per semester	6.42
			Total Weekly Contact Hours	4.00

Module Resources

Recommended Book Resources

Michael T. Goodrich,Roberto Tamassia,Michael H. Goldwasser. (2014), Data Structures and Algorithms in Java, 6th Edition. John Wiley & Sons, p.736, [ISBN: 1118771338].

Supplementary Book Resources

Donald Ervin Knuth. (1997), The Art of Computer Programming: Fundamental algorithms, 3rd Edition. Addison-Wesley Professional, p.650, [ISBN: 0201896834].

Donald Ervin Knuth. (1998), The Art of Computer Programming: Sorting and searching, 2nd Edition. Addison-Wesley Professional, p.780, [ISBN: 0201896850].

Paul Deitel,Harvey Deitel. (2017), Java How to Program, Early Objects, Pearson, 11th Edition, p.1296, [ISBN: 9780134743356].

This module does not have any article/paper resources

This module does not have any other resources

Discussion Note:	