H6PROG3: Programming III

Module Code:	H6PROG3
Long Title	Programming III APPROVED
Title	Programming III
Module Level:	LEVEL 6
EQF Level:	5
EHEA Level:	Short Cycle
Credits:	10
Module Coordinator:	FRANCES SHERIDAN
Module Author:	FRANCES SHERIDAN
Departments:	School of Computing
Specifications of the qualifications and experience required of staff	MSc Degree in Computing or cognate discipline or equivalent industry experience as a programmer.

Learning Outcomes

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

#	Learning Outcome Description
LO1	Explain the theory, concepts and principles of various elementary algorithms
LO2	Use iterative and recursive techniques to design, implement, and test sorting and searching algorithms
LO3	Appropriately apply a variety of algorithms to solve real-world problems
LO4	Conduct algorithm analysis in terms of performance and time complexity.

Dependencies

Module Recommendations

No recommendations listed

Co-requisite Modules

No Co-requisite modules listed

Entry requirements	Learners should have attained the knowledge, skills and competence gained from stage 1 of the BSc (Hons) in Data Science

H6PROG3: Programming III

Module Content & Assessment

Indicative Content
Recursion • Recursion vs iteration . • Properties of problems which can be solved by recursion
Functional Programming • Benefits of functional programming . • Eliminating side effects . • Lambdas
Algorithm Design • Assessing algorithm run-time complexity . • Assessing algorithm data storage requirements . • Determining the correctness of an algorithm
Trees • Introduction to Trees . • Tree structure and tree traversal .
Trees 2 • Searching a tree • Implementing a Binary Search Tree (BST) . • How to balance a tree . • Serializing a Data Structure
Graphs • What is a graph? . • How to represent a graph as a data structure . • Graph types (simple, directed, weighted)
Graphs 2 • Operations on Graphs • Implementing a graph using linear data structures
Search Algorithms • The importance of search algorithms . • Sequential search . • Binary Search . • Implementation of search for linear data structures
Sorting Algorithms • The importance of sorting . • Sorting and Searching and their interconnections . • Bubble Sort . • Insertion Sort
Sorting Algorithms 2 • Selection Sort . • Merge Sort .
Sorting Algorithms 3 • Quick Sort . • Specialized sorting algorithms for data
Algorithms on Graphs • Searching a Graph . • Dijkstra's Algorithm

Assessment Breakdown	%
Coursework	100.00%

Assessments

Full Time

Coursework

Assessment Type:	Continuous Assessment	**% of total:**	Non-Marked
Assessment Date:	n/a	**Outcome addressed:**	1,2,3,4
Non-Marked:	Yes		

Assessment Description:
Students will be given formative assessments to prepare them for the graded components, it is envisaged that the formative assessment will be largely of the same form as identified for the “lab work” segment discussed below.

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

Assessment Description:
Each week student will submit program code to the Moodle server for grading. Student will be supplied with an interface specification for the program(s) and the grading will be conducted via automated unit testing based on unknown inputs. Students will be examined on their ability to convey understanding of the programs which they have developed.

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

Assessment Description:
The students will have to develop solutions to programming problems relevant to all material covered in the module using a proctored computer in an examination environment. There will be a written component to assess the student ability to determine errors in a program.

No End of Module Assessment

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
The repeat strategy for this module is a practical programming examination. Students will be afforded an opportunity to repeat the examination at specified times throughout the year and all learning outcomes will be assessed in the repeat exam.

H6PROG3: Programming III

Module Workload

Module Target Workload Hours 0 Hours

Workload: Full Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Classroom & Demonstrations (hours)	24	Per Semester	2.00
Tutorial	Other hours (Practical/Tutorial)	36	Per Semester	3.00
Independent Learning	Independent learning (hours)	190	Per Semester	15.83
			Total Weekly Contact Hours	5.00

Module Resources

Recommended Book Resources

Goldwasser, M. T., Tamassia, R. &Goodrich, M. T.. (2013), Data Structures and Algorithms in Python, KG, Berlin: Springer-Verlag Berlin and Heidelberg GmbH & Co.

Kleinberg, J. & Tardos, E.. (2005), Algorithmic Design, USA.

Supplementary Book Resources

Cormen et al, T.. (2016), Introduction to Algorithms.

This module does not have any article/paper resources

This module does not have any other resources

Discussion Note:	