H6PROG2: Programming II

Module Code:	H6PROG2
Long Title	Programming II APPROVED
Title	Programming II
Module Level:	LEVEL 6
EQF Level:	5
EHEA Level:	Short Cycle
Credits:	5
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 the equivalent experience in industry as programmer.

Learning Outcomes

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

#	Learning Outcome Description
LO1	Apply theoretical concepts to a range of contexts and problem domains
LO2	Formulate computer program solutions to well defined abstract problems
LO3	Use object-oriented techniques such as interfaces, inheritance, and generics to package ADTs appropriately
LO4	Analytically incorporate ADTs and associated implementations into systems that use complex data structures.

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

H6PROG2: Programming II

Module Content & Assessment

Indicative Content
Data Connectivity • Low Level and High Level File I/O . • Database Programming - CRUD . • Parsing Data Exchange Formats e.g. JSON, XML • File Manipulation . • With with UNIX pipes (accepting input producing output)
Exception Handling • Dealing with errors via exception handling mechanisms • Syntactic and semantic errors (run-time and before) • Error mitigation
Inheritance and Polymorphism • The role of reuse and inheritance . • How to utilize polymorphic constructs in programming . • Use of support libraries from external sources
Regular Expressions • Introduction to Regular Expressions
Regular Expressions 2 • Developing programs for data processing activities (e.g., data extraction, cleaning, merging, aggregation, analysis, reporting) using regular expressions
Design Patterns • What are Design Patterns? / Gang of Four patterns . • Template Pattern . • Strategy Pattern . • Observer Pattern
Design Patterns 2 • Composite Pattern . • Design Patterns for event handling . • Stream Processing
Software Testing • The importance of testing • Methods of testing . • Writing a Unit Test . • Preconditions and post conditions . • Black Box and White Box
Linear Data Structures • Refresher on Data Structures . • Lists (Singly linked and doubly linked)
Linear Data Structures • Stacks . • Queue
Linear Data Structures • Operations performed on Linear Data Structures
Associative Data Structures • Key-value pairs . • Maps (Hash-Maps) . • JSON
Associative Data Structures • Extending in-built classes with new functionality (e.g. new hashing algorithms for Maps)

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:
Ongoing independent and group programming activities and feedback.

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:	Practical	**% 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.

H6PROG2: Programming II

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)	24	Per Semester	2.00
Independent Learning	Independent learning (hours)	77	Per Semester	6.42
			Total Weekly Contact Hours	4.00

Module Resources

Recommended Book Resources
Lutz, M.. (2013), Learning Python (5th ed), O'Reilly Media.
Supplementary Book Resources
Beazley, D. & Jones, B. K.. (2013), Python Cookbook (3rd ed), O'Reilly Media.
This module does not have any article/paper resources
This module does not have any other resources

Discussion Note:	