

H6PROG1: Programming I

Module Code:	H6PROG1
Long Title	Programming I APPROVED
Title	Programming I
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 equivalent industry experience.
Learning Outcomes	
<i>On successful completion of this module the learner will be able to:</i>	
#	Learning Outcome Description
LO1	Explain the core concepts of programming
LO2	Implement the core syntax and semantics of an object oriented programming language
LO3	Identify and implement principles of good algorithm design
LO4	Locate, and address, logical and syntactic errors in computer programs
Dependencies	
Module Recommendations	
No recommendations listed	
Co-requisite Modules	
No Co-requisite modules listed	
Entry requirements	See section 4.2 Entry procedures and criteria for the programme including procedures recognition of prior learning

H6PROG1: Programming I

Module Content & Assessment			
Indicative Content			
Programming Languages Introduction to Programming. . • Programming Abstractions (High-level to Low-Level) . • Programming Environments (Text Editors, IDEs)			
Programming Languages 2 Programming paradigms (Functional, Declarative, Imperative, Object-Oriented) . • Beginning Object-Oriented Programming . • Compiled vs Interpreted. • Approaches to Problem Solving . • Emerging Programming Paradigms and Languages			
Working with Numerical Data The use of variables . • Appropriate numerical data types . • Arithmetic expression evaluation . • Logical operations and conditional logic . • The role of constants			
Conditionals • Controlling what happens in the program with an 'if' statement . • Boolean logic expression to control the condition of the if statement			
Conditionals 2 • Nesting 'if' statements . • Other conditional constructs such as the 'switch' statement			
Repetition • How to make a program repeat some portion of itself . • while loops, for loops and other forms of repetition			
Repetition 2 • the role of Boolean logic in looping . • Identifying the appropriate loop type			
Characters & Strings • Textual data as Characters & Strings . • Processing text input (parsing and modifying) . • Producing text output			
Arrays • The basics of what an Array is . • Using arrays to store data. • Arrays and loops . • Arrays to store Objects and primitives . • Arrays as method parameters . • N-dimensional Arrays			
Classes • Creating a class to encapsulate an entity . • Instantiating a class . • Constructor arguments and mutable state . • Variable scoping (local, class level etc.) . • Access modifiers . • Method return values . • Rules for parameter passing (pass by reference, pass by value)			
Object Oriented Programming Basics • Objects as State and Functionality united . • Object Declaration & Instantiation . • Principles of Message Passing . • Program Structure (Imports, Comments, route of execution, methods, classes)			
User Involvement • Accepting input from the user or another program . • Producing output to standard output or to a file.			
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
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. Topics which will be addressed include control flow, use of variables, intelligent usage of objects.			
Assessment Type:	Practical	% of total:	50
Assessment Date:	n/a	Outcome addressed:	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 <i>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.</i>			
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.			

H6PROG1: Programming I

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