H06IP: Introduction to Programming

Module Code:	H06IP
Long Title	Introduction to Programming APPROVED
Title	Introduction to Programming
Module Level:	LEVEL 6
EQF Level:	5
EHEA Level:	Short Cycle
Credits:	5
Module Coordinator:	FRANCES SHERIDAN
Module Author:	Sam Cogan
Departments:	School of Computing
Specifications of the qualifications and experience required of staff	Master's degree in computing or cognate discipline.

Learning Outcomes	
On successful completion of this module the learner will be able to:	
#	**Learning Outcome Description**
LO1	Explain the core concepts of programming
LO2	Implement the core syntax and semantics of a modern 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		
67614	H06IP	Introduction to Programming
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

H06IP: Introduction to Programming

Module Content & Assessment

Indicative Content
Programming Languages Introduction to Programming. •Programming Abstractions (High-level to Low-Level) •Programming Environments (Text Editors, IDEs) •Programming paradigms (Functional, Declarative, Imperative, Object-Oriented) •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
Beginning Object Oriented Programming •Object Declaration. •Object Creation. •Program Components. •Import Statement. •Class Declaration •Method Declaration •Input/output
Defining Instantiable Classes •Defining Instantiable Classes. •Instantiable Classes and Constructors. •Local variables. •Return Values. •Parameter Passing. •Visibility Modifiers: public and private •Multiple Constructors.
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 and 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 2 •Arrays as method parameters •2-dimensional Arrays
Additional Language Students are introduced to a second language and are shown device setup and introductory paradigms

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

Assessments

Full Time

Coursework

Assessment Type:	Continuous Assessment	**% of total:**	40
Assessment Date:	Every Week	**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 either via automated unit testing based on unknown inputs, or via in class marking of work. 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:	Project	**% of total:**	10
Assessment Date:	Sem 2 End	**Outcome addressed:**	2,3
Non-Marked:	No		

Assessment Description:
Students are given a small project to produce in a secondary language. The project will cover the main basic programming paradigms eg -If statements -Loops -Arrays

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
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.

H06IP: Introduction to 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
Practical	No Description	24	Per Semester	2.00
Independent Learning	No Description	77	Once per semester	6.42
			Total Weekly Contact Hours	4.00

Workload: Part Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	No Description	24	Per Semester	2.00
Independent Learning	No Description	72	Once per semester	6.00
Lab	No Description	24	Per Semester	2.00
			Total Weekly Contact Hours	4.00

Module Resources

Recommended Book Resources

Sedgewick, Robert & Wayne, Kevin,. (2017), ntroduction to Programming in Java: An Interdisciplinary Approach,, 2nd Edition. Addison-Wesley.

Supplementary Book Resources

Deitel & Deitel. (2014), How to Program in Java, 10th. Prentice Hall.

Schildt Herbert. (2014), Java: A Beginner's Guide, 6. McGraw-Hill Osborne, [ISBN: 978-007180925].

Walter Savich. (2014), Java: An Introduction to Problem Solving and Programming, 7th. Addison-Wesley.

Mark Lutz. (2013), Learning Python, 5th Edition, 5. O'Reilly.

This module does not have any article/paper resources

Other Resources

[Website], CodeAcademy. (2019), Learn Java, CodeAcademy, https://www.codecademy.com/learn/learn-j ava

Discussion Note:	