Title:	Programming Fundamentals APPROVED
Long Title:	Programming Fundamentals

Module Code:	SOFT6018	**Duration:**	1 Semester

Credits:	5
NFQ Level:	Fundamental
Field of Study:	Computer Software
Valid From:	Semester 1 - 2017/18 (September 2017)
Module Delivered in	6 programme(s)
Next Review Date:	October 2021
Module Coordinator:	Sean McSweeney
Module Author:	CLIONA MC GUANE

Module Description:	Computer programming involves instructing the computer to perform a certain task. It achieves this by writing software (computer programs) that defines instructions presented as source code. These instructions manipulate different types of objects and involves describing processes, procedures and algorithms. In this module students will be introduced for the first time to computer programming. As part of this module the fundamental programming concepts and constructs in a high-level programming language will be presented.

Learning Outcomes	
On successful completion of this module the learner will be able to:	
LO1	Determine the outcome of programs, written in a high-level programming language, that utilise basic programming concepts and constructs.
LO2	Implement solutions to programming tasks by identifying necessary variables and constants and choosing appropriate data types for these variables and constants.
LO3	Choose appropriate sequential, conditional and iterative constructs for a given task.
LO4	Implement solutions to programming tasks that require files.
LO5	Test and debug programs developed.
LO6	Document code using best practices and conventions.

Pre-requisite learning
Incompatible Modules *These are modules which have learning outcomes that are too similar to the learning outcomes of this module. You may not earn additional credit for the same learning and therefore you may not enrol in this module if you have successfully completed any modules in the incompatible list.*
No incompatible modules listed
Co-requisite Modules
No Co-requisite modules listed
Requirements *This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed. You may not enrol on this module if you have not acquired the learning specified in this section.*
No requirements listed
Co-requisites
No Co Requisites listed

Module Content & Assessment

Indicative Content
Fundamental Programming Concepts Students will learn the fundamental programming concepts that apply to modern programming languages: data types, variables, constants, operators, expressions, input and output using keyboard and screen. Control structures - conditional control (if, else, if then), iterative control (loop and exit statements), sequential control and selection statements.
Documentation of Code Students will be encouraged to document their code at all stages and to employ appropriate layout and use standard naming conventions to aid the readability of their code.
Testing and Debugging Test plans will be designed during the design stage to ensure that the defined problem is understood and that the student's subsequent code solves the defined problem. Students will learn how to use an appropriate development environment to identify and eliminate syntax and logical errors.
Files This section shall include, but is not be limited to, writing to and reading from text files, reading up to the end of the file, and storing data in appropriate constructs for processing.

Assessment Breakdown	%
Course Work	100.00%

Course Work

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Multiple Choice Questions	This quiz will examine a student's knowledge of basic programming concepts, his/her ability to understand code and his/her ability to fix code.	1,2,3	10.0	Week 4
Practical/Skills Evaluation	An open book in-lab exam in which the student can use his/her knowledge, programs and notes to code a solution to a defined problem. An example exam would be getting data from the user through the keyboard, processing that data, and displaying a result on the screen. A test plan may be provided to the student.	2,3,5,6	15.0	Week 7
Project	The project aims to evaluate the student's programming ability by solving a defined problem. An example of a project would be a Christmas wish-list system, in which users would create an account, provide details of items and their prices, and the list would be stored for him/her. The user would be able to retrieve and add to these lists at any time. The student may be required to develop a test plan.	2,3,4,5,6	25.0	Week 10
Practical/Skills Evaluation	An open book in-lab exam in which the student can use his/her knowledge, programs and notes to code a solution to a defined problem. An example of an exam might be a programme to determine the winner of a race. Users would provide the details of the an entrant's name and his/her race time. The winner of the race would be displayed on the screen along with some statistics e.g. average time, slowest time. All data would be stored in a file. A test plan may be provided to the student.	2,3,4,5,6	50.0	Week 13

No End of Module Formal Examination

Reassessment Requirement
Coursework Only *This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.*

The institute reserves the right to alter the nature and timings of assessment

Module Workload

Workload: Full Time				
Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Introduce programming concepts.	2.0	Every Week	2.00
Lab	Design, code, test and debug solutions to defined problems.	3.0	Every Week	3.00
Independent Learning	Practise coding.	2.0	Every Week	2.00
			Total Hours	7.00
			Total Weekly Learner Workload	7.00
			Total Weekly Contact Hours	5.00

Workload: Part Time				
Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Introduce programming concepts.	1.0	Every Week	1.00
Lab	Design, code, test and debug solutions to defined problems.	3.0	Every Week	3.00
Independent Learning	Practise coding.	3.0	Every Week	3.00
			Total Hours	7.00
			Total Weekly Learner Workload	7.00
			Total Weekly Contact Hours	4.00

Module Resources

Recommended Book Resources

- **Tony Gaddis 2014, *Starting out with Python, Global Edition*, 3 Ed., Pearson [ISBN: 9781292065502]**
- **Tony Gaddis 2015, *Starting out with Java: From Control Structures through Objects*, 6 Ed., Pearson [ISBN: 9781292110653]**

This module does not have any article/paper resources

Other Resources

- **Website: *Style Guide for Python Code***
  https://www.python.org/dev/peps/pep-0008 /
- **Website: *Python IDE***
  https://www.jetbrains.com/pycharm/
- **Website: *Eclipse IDE***
  https://eclipse.org/
- **Website: *Java IDE***
  http://www.bluej.org/
- **Website: *YouTube video tutorials for a variety of programming languages***
  http://www.thenewboston.com
- **Website: *Python Language Reference***
  https://docs.python.org/3/
- **Website: *Java JDK***
  http://www.oracle.com/technetwork/java/j avase/downloads/index.html

Module Delivered in

Programme Code	Programme	Semester	Delivery
CR_KSDEV_8	Bachelor of Science (Honours) in Software Development	1	Mandatory
CR_KDNET_8	Bachelor of Science (Honours) in Computer Systems	1	Mandatory
CR_KITMN_8	Bachelor of Science (Honours) in IT Management	1	Mandatory
CR_KITSP_7	Bachelor of Science in Information Technology	1	Mandatory
CR_KCOMP_7	Bachelor of Science in Software Development	1	Mandatory
CR_KCOME_6	Higher Certificate in Science in Software Development	1	Mandatory