Module Title	**Fundamentals of Software Development**
Level	4
Reference No.	CSI_4_FSD
Credits	20
Student Study Hours	Total: 200 Contact hours: 52 Student managed learning hours: 148
Pre-Requisites	None
Co-requisites	None
Excluded combinations	None
Module coordinator	TBC
Division	Division of Computer Science and Informatics
Short Description	This module teaches the fundamentals of computer programming covering variables, datatypes, arrays, algorithms, conditional and iterative code and the use of functions. Students will learn to write simple programs making use of a contemporary programming language and development environment.
Aims	To provide students with fundamental programming skills. To give students understanding of the fundamental concepts of software development such as source code and the compilation and execution of programs. To enable students to become familiar with development tools and environments.
Learning Outcomes	**LO1: Knowledge and Understanding** ● Describe design notations, software development environments and programming languages and their purpose and interaction. (Maps to: BCS 2.2.1 a1-a6, a9) **LO2: Intellectual Skills** ● Interpret and analyse specifications. () ● Make effective use of technical reference materials. (Maps to: BCS 2.2.1 a1-a6, a9) **LO3: Practical Skills** ● Design, write, test, correct and document simple software to implement given specifications. (Maps to: BCS 2.2.1 a7-a8) **LO4: Transferable Skills** ● Keep a coherent, evaluative and reflective log of work produced. (Maps to: BCS 2.2.1 c1-c2)
Employability	This module will give you the necessary vocabulary to discuss software development with colleagues. All Information Technology professionals need to understand the process of software development even if they do not directly develop software.
Teaching and Learning Pattern	This module emphasises learning through practical exercises and the development of actual software artefacts. Short lectures will be used to inform the laboratory activities and provide a forum for discussion of issues students have encountered in the practical work. The lab sessions will occupy the majority of the contact time and will involve much independent working. Students are required to keep a clear record of the work they have done and are encouraged to experiment and investigate beyond the basic material being taught.

Indicative Content	● The common elements of a programming language. ● Using the IDE to assist with software development. ● Syntax of a programming language. ● Variables, datatypes, arrays and list structures. ● Conditional and iterative programming constructs. ● The use of functions and variable scoping. ● Basic Input/Output techniques. ● Internal documentation.
Assessment Elements & weightings	**COURSEWORK 100%** **Summative Assessment** Coursework: expected to be individually assessed by an in-class test and a lab-based development exercise. (LO1-LO4) **Formative Assessment** Skills for the summative assessment will be embedded throughout formative opportunities in Lectures and Workshops. Formative assessment will take different forms, such as: ● logbook used for formative assessment by discussion with the tutor throughout the module ● verbal feedback on tutorial activities ● observation and questioning to provide instant feedback as the student takes part in learning activities
Indicative Sources (Reading lists)	**Core**: ● Barry, P. (2016) *Head first Python*. Cambridge: O'Reilly Media ● Chun, W.J. (2006) *Core Python programming* (2nd Ed.) Prentice-Hall ● Sedgewick, R., Wayne, K. (2013) *Introduction to Programming in Java: An Interdisciplinary Approach*. Pearson; ISBN 1292025565 ● Lliang, Y. (2014) *Intro to Java Programming, Comprehensive Version*. Pearson; ISBN 1292070013 **Supplementary**: ● Dawson, M. (2010) *Python programming for the absolute beginner.* (3rd ed) Cengage Learning ● Lutz, M. (2009) *Python pocket reference* (4th Ed.) Cambridge: O'Reilly Media ● Lutz, M. (2009) *Programming Python* (4th Ed.) Cambridge: O'Reilly Media