Module Title	**Software Development**
Level	4
Reference No.	
Credits	20
Student Study Hours	Total: 200 Contact hours: 65 Student managed learning hours: 135
Pre-requisites	Fundamentals of Software Development
Co-requisites	None
Excluded combinations	None
Module co-ordinator	Paul Carden
Division	Computer Science & Informatics
Short Description	This module helps you to understand, develop a vocabulary and acquire some simple programming skills using a programming language. It is using the programming skills/knowledge gained during the first semester to extend the complexity of the coding. It will provide a familiarity with Objects and Classes and some experience in simple GUIs. Good development principles and the kind of support an IDE will provide will also be looked at including all important aspects of the application development process in whatever environment or programming language you may use in the future.
Aims	Building on the programming from the first semester, the aim of this module is to develop a vocabulary and set of skills which will support you in developing software in any application area.
Learning Outcomes	**LO1: Knowledge and Understanding** ● Describe and evaluate design notations, software development environments and programming languages **LO2: Intellectual Skills** ● Interpret and analyse requirements Search different sources for appropriate components (Maps to: BCS 2.2.1 a1-a5, a7-a9;) **LO3: Practical Skills** Specify, design, write, test, correct and document software to implement given requirements using built-in components (Maps to: BCS 2.2.1 b1-b4;) **LO4: Transferable Skills** You will develop problem solving skills (Maps to: BCS 2.2.1 c1-c2;)
Employability	This module delivers an increasingly sought after set of skills that are greatly valued in the job market and required by many companies. The skills developed in this module are applicable to many different languages and development platforms and are commonly sought after. All Information Technology professionals need to understand the process of software development even if they do not directly develop software.

Teaching and learning pattern	The lectures and the exercises deliver the basic concepts. The learning exercises are there to help you learn those concepts and the applications you are building put those ideas into context. The lab sessions allow your tutor the time to support your learning and for you to tackle the learning exercises. Private study time is essential for good progress.
Indicative content	● application development using Python ● functions ● classes ● objects ● scenarios from different application areas ● GUIs ● development tools
Assessment *Elements & weightings*	**COURSEWORK 100%** **Summative Assessment** Coursework: Expected to consist of individually assessed practical assessment linked to the development of an increasingly more complex piece of software. Participants provide evidence that they have been able to do the various parts of the assessment. (LO1, L04) **Formative Assessment** Formative assessment will be used throughout in a form of: ● observations ● quizzes ● Q&A ● Individual support and feedback on completed work will be given during most of the tutorials
Indicative Sources *(Reading lists)*	**Core**: ● http://sthurlow.com/python/ Online resource ● https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3 Online resource ● https://docs.python.org/3/tutorial/index.html - The official python site ● Dawson, M. (2010). *Python programming for the absolute beginner*. Boston, Mass.: Course Technology, a part of Cengage Learning. The CSI standard text ● Johnson, M. (2012). *A Concise Introduction to Programming in Python*. Boca Raton, Florida: CRC Press Taylor & Francis Group, *Second Edition*. Very good practical examples, short chapters **Optional:** ● Lutz, M. (2015). *Python pocket reference*. Sebastopol [etc.].[unknown location]: O'Reilly Media. This is small enough to keep with you. ● Severance, C. (2013). *Python for informatics*. 2nd ed. Ann Arbor, MI: Dr Chuck. **Advanced references:** ● Lutz, M. (2018). *Learning Python*. [unknown location]: O'Reilly. ● Chun, W. (2007). *Core Python programming,* 2nd Ed. Upper Saddle River, N.J.: Prentice Hall. ● Barry, P. (2010). *Head first Python*. Sebastopol, Calif.: O'Reilly.