

Module Title	Software Development
Level	4
Reference No.	CSI_4_SOD
Credits	20
Student Study Hours	Total: 200 Contact hours: 52 Student managed learning hours: 148
Pre-requisites	Fundamentals of Software Development
Co-requisites	None
Excluded combinations	None
Module co-ordinator	TBC
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	<p>LO1: Knowledge and Understanding</p> <ul style="list-style-type: none"> Describe and evaluate design notations, software development environments and programming languages <p>LO2: Intellectual Skills</p> <ul style="list-style-type: none"> Interpret and analyse requirements ∓ Search different sources for appropriate components (Maps to: BCS 2.2.1 a1-a5, a7-a9) <p>LO3: Practical Skills</p> <ul style="list-style-type: none"> ∓ Specify, design, write, test, correct and document software to implement given requirements using built-in components (Maps to: BCS 2.2.1 b1-b4) <p>LO4: Transferable Skills</p> <ul style="list-style-type: none"> ∓ 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	<ul style="list-style-type: none"> application development using Python functions classes objects scenarios from different application areas GUIs development tools

<p>Assessment <i>Elements & weightings</i></p>	<p>COURSEWORK 100%</p> <p>Summative Assessment</p> <p>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-LO4)</p> <p>Formative Assessment</p> <p>Formative assessment will be used throughout in a form of:</p> <ul style="list-style-type: none"> ● observations ● quizzes ● Q&A ● Individual support and feedback on completed work will be given during most of the tutorials
<p>Indicative Sources <i>(Reading lists)</i></p>	<p>Core:</p> <ul style="list-style-type: none"> ● http://sthurlow.com/python/ ● https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3 ● https://docs.python.org/3/tutorial/index.html - The official python site ● Steven F. Lott, Functional Python programming, Packt Publishing, 2018 ● Lutz, Mark. Python Pocket Reference : Python in Your Pocket, O'Reilly Media, Incorporated, (2014). ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/lbsuuk/detail.action?docID=1619476. <p>Optional:</p> <ul style="list-style-type: none"> ● “Python Pocket Reference” Mark Lutz is small enough to keep with you ● “Learning Python” Mark Lutz 4th edition O'Reilly. This is quite intense. Not for you if you cannot understand the basics. ● The following are very good books but not for the absolute beginner ● “Core Python programming”, Wesley. J. Chun 2nd Ed, Prentice Hall, 2007 ● “Head First Python” Paul Barry ● “Programming Python, Second Edition By Mark Lutz ● BIF-4-SD2 6Bell Douglas Software Engineering for Students: A Programming Approach Addison-Wesley, 4th Edition, 2005, ISBN 0321261275 If you want to understand where we are going with this. <p>(There are many other internet resources, and these will be documented on Moodle)</p>