Module Title	**Python programming for AI and Visualization**
Course Title	MSc Applied AI
School	☐ ASC ☐ ACI ☐ BEA ☐ BUS ☒ ENG ☐ HSC ☐ LSS
Division	Computer Science and Informatics
Parent Course (if applicable)	
Level	7
Semester	The module will be offered in both semester
Module Code *(showing level)*	CSI_7_PPA
JACS Code (completed by the AQE)	
Credit Value	20 credit points
Student Study Hours	Total: 200 Contact hours: 52 Student managed learning hours: 148 Requirements for Self-Managed Learning Hours: Undertake research work, complete and write up lab exercises and assessments.
Pre-requisite Learning	None
Co-requisites	None
Excluded combinations	None
Module co-ordinator	Name: TBC
Short Description (max. 100 words)	This module provides an intensive introduction to programming, especially for students without a computer science background. The module will cover the basics of program development in Python, with a specific attention of its use in visualizing and exploring data, and in translating mathematical problems and models into a computational solution
Aims	The module aims at providing the students with the fundamentals programming skills to understand and develop computational solutions in Python, and to translate formalized algorithm in computational solutions.
Learning Outcomes (4 to 6 outcomes)	**A. Knowledge and Understanding:** • Knowledge and understanding of fundamental programming concepts (covers BCS requirements: 6.1.1, 6.1.2, 6.1.3, 7.1.2, 7.1.6, 7.1.7, 7.1.8; 8.1.1, 8.1.2, 8.2.1, 8.2.2, 9.2.1, 9.2.2) • Knowledge and understanding of Python programming language (covers BCS requirements: 6.1.1, 6.1.2, 6.1.3, 7.1.2, 7.1.6, 7.1.7, 7.1.8; 8.1.1, 8.1.2, 8.2.1, 8.2.2, 9.2.1, 9.2.2) **B. Intellectual Skills:**

	• Conduct a critically evaluative analysis of a case-based domain using appropriate mathematical and computational modelling; also developing the in-depth knowledge necessary to identify and apply suitable techniques in order to synthesize advanced theory/practical concepts. (covers BCS requirements: 8.1.1 - 8.1.3; 9.1.1 - 9.1.3; 10.1.1 - 10.1.3) **C. Practical Skills:** • Ability to develop a basic program following the principles of software design and development, including debugging and documentation (covers BCS requirements: 8.1.1 - 8.1.3; 9.1.1 - 9.1.3) • Ability to frame a problem in a controlled simulation, to gather insight in the complexity of the data (covers BCS requirements: 8.1.1 - 8.1.3; 9.1.1 - 9.1.3; 10.1.1 - 10.1.3) **D. Transferable Skills:** • Be able to visually present the results of computational model and of data analysis (BCS requirements: 8.2.1, 8.2.1; 9.1.1 - 9.2.3; 10.2.1 - 10.2.2) • Be able to make concise, engaging and well-structured oral presentations, arguments, and explanations; Communication /presentation of computational solution and coding prototype to a wide range of audiences. (BCS requirements: 8.2.1, 8.2.1; 9.1.1 - 9.2.3; 10.2.1 - 10.2.2)
Employability	There is current and rapidly increasing commercial need/demand for graduates/postgraduates with skills in the areas of computing and computational modelling. The module delivers skills in these areas that are directly relevant in both commercial and research environments.
Teaching and learning pattern	Contact hours includes the following: (please click on the checkboxes as appropriate) ☒ Lectures ☐ Group Work: ☐ Seminars ☐ Tutorial: ☒ Laboratory ☐ Workshops ☐ Practical ☐ VLE Activities
Indicative content	The following list of topics is indicative (not exhaustive) of typical module content: • Fundamentals of programming: data, data representation, variables. Conditional structures and iterations. Functions (input parameters and output values, variable scope), classes and modules. • Using programming libraries. • Best practices of Python coding software engineering of solutions, testing, debugging, documenting • Translating mathematical formulations and pseudo-coded algorithms into Python. Basic optimization methods: gradient descent, simulated annealing. • Data loading and organization • Data visualization • Data simulations
Assessment method (Please give details – of components, weightings, sequence of components, final component)	Formative assessment: The students will usually be given a range of weekly tutorial-based tasks (both individual/group work) comprised of formative exercises imparting the knowledge and skills required to satisfy the learning outcomes Summative assessment: Coursework 50% Final report 50%

Mode of resit assessment (if applicable)	Formative assessment: Summative assessment: Coursework 50% Examination 50%
Indicative Sources ***(Reading lists)***	**Core Materials:** • Downey, Allen, Think Python, Second edition. (O'Reilly Media 2016) • Phuong Vothihong, Martin Czygan, Ivan Idris, Magnus Vilhelm Persson & Luiz Felipe Martins. Python: End to End Data Analysis. (Packt Publishing 2017) **Optional Materials:** • Zed A. Shaw, : A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code, (Addison-Wesley Professional 2017) • Anthony Scopatz, Kathryn D. Huff , Effective Computation in Physics: Field Guide to Research with Python (O'Reilly, 2015) • Sarah Guido, Andreas C. Mueller , Introduction to Machine Learning with Python: A Guide for Data Scientists (O'Reilly, 2016) • Mario Döbler, Tim Großmann, The Data Visualization Workshop: An Interactive Approach to Learning Data Visualization, 2nd Edition Paperback (2020) • DE Knuth. The Art of Computer Programming - Volume 1 (Addison-Wesley; 3rd edition 1997). • Benjamin Baka. Python Data Structures and Algorithms. (Packt Publishing 2017)
Other Learning Resources	TBD Supplementary materials for all of the software used in the module will be available on the module VLE site.