Module Specification

Module Title	Introduction to Scientific Programming (15 credits)	Module Code	SBC5291

Credit Value:	15	Level:	5	Mode of Delivery:	On Campus	Semester:	1

Pre-requisite modules	Co-requisite modules	Overlapping modules

1) Content Description

Provide a description of the module, as it will appear in the Module Directory and on the Student Information System (approx. 70-80 words).

The module will provide basic computer programming skills as a tool for problem-solving and scientific data analysis, with emphasis on gaining hands-on programming experience. Topics to be covered will include basic concepts of algorithm design, use of an integrated development environment, data structures, control flow, functions and libraries. Applications will include visualisation, analysis and modelling of data relevant to Natural Sciences.

2) Module Aims

Specify the aims of the module, i.e. the broad educational purposes for offering this module.

To introduce basic concepts of computer programming, with emphasis on problem-solving and data analysis. In particular, the module will aim at providing students with:

- the ability to analyse a problem and decompose it into a sequence of instructions that can be implemented in a computer program
- hands-on experience with an integrated development environment to write, execute, test and share a program
- working knowledge of the basic syntax of a programming language
- the ability to write simple programs to analyse data from their own and other disciplines

3) Learning Outcomes

Identify the learning outcomes for this module, i.e. knowledge, skills and attributes to be developed through completion of this module. Outcomes should be referenced to the relevant QAA benchmark statements and the Framework for Higher Education Qualifications in England, Wales and Northern Ireland (2008). The SEEC Credit Level Descriptors for Further and Higher Education 2003 and Queen Mary Statement of Graduate Attributes should also be used as a guiding framework for curriculum design.

Academic Content:	
A1	Knowledge of fundamental programming concepts, including programming environments, data structures, control flow, input/output, functions and data visualisation and analysis.

Disciplinary skills - able to:	
B1	Design a basic algorithmic procedure to solve a problem.
B2	Write and test simple code to analyse scientific data.
B3	Use an integrated development environment for code writing, execution and testing.

Attributes:	
C1	Use information technologies to elaborate and interpret information effectively
C2	Use information technologies to visualise data effectively
C3	Apply analytical skills to investigate unfamiliar problems

QM Model Outcomes (available in QMPlus here):	
D1	(Multi/Inter-Disciplinarity) Demonstrate how discipline specific problem solving techniques or approaches may be generalised or applied in a broader context

4) Reading List

Provide an indicative reading list for the module. This should include key texts and/or journals but should not be an exhaustive list of materials.

C. Hill, "Learning Scientific Programming with Python", Cambridge University Press, 2016.
A. Sweigart, "Automate the boring stuff with Python", No Starch Press, 2015.
M. Dawson, "Python programming for the absolute beginner", 3rd edition, Course Technology PTR, 2010.
P. Barry, "Head First Python", O'Reilly, 2011.

5) Teaching and Learning Profile

Provide details of the method of delivery (lectures, seminars, fieldwork, practical classes, etc.) used to enable the achievement of learning outcomes and an indicative number of hours for each activity to give an overall picture of the workload a student taking the module would be expected to undertake. This information will form the Key Information Set for each undergraduate programme and will be used to populate the KIS widget found on the QMUL programme information pages. More information can be found online about KIS. You may also wish to refer to the QAA guidance on contact hours when completing this section.

Activity Type	KIS Category	Time Spent (in hours)
Lecture	Scheduled	9
Practical Classes and workshops	Scheduled	14
	Total	

Specify the total module notional study hours. This should be a total of the hours given for each activity. The notional study hours for each academic credit point is 10. A 15 credit point module therefore represents 150 notional study hours.

Activity Type	Total Time Spent (in hours)	Percentage of Time Spent
Scheduled learning and teaching	23	15
Placement	0	0
Independent Study	127	85
Total	150	100

Use the information provided in the box above to specify the total time spent and the percentage time spent in each category of teaching and learning activity.

6) Assessment Profile

Provide details of the assessment methods used to assess the achievement of learning outcomes.

Description of Assessment	Assessment Type	KIS Category	Duration/Length	Percentage Weighting	Final element of assessment	Qualifying Mark
Programming courseworks	Written assignment, inc Essay	Coursework		100	Yes	

Final element of assessment: The assessment that takes place last. There should normally be only one element of assessment marked as final unless two assessment or submission dates occur on the same day.

Qualifying mark: A specified minimum mark that must be obtained in one or more elements of assessment in order to pass a module. This is in addition to, and distinct from, the requirement to achieve a pass in the module mark to pass the module.

Reassessment

Provide details of the reassessment methods used, specifying whether reassessment is either standard reassessment or synoptic reassessment.

○ Standard Reassessment ◉ Synoptic Reassessment

Synoptic reassessment details (if you have indicated synoptic reassessment above, please give details)		
Brief Description of Assessment	Assessment Type	Duration/Length of Examination/ Coursework

Programming courseworks	Written assignment, inc Essay	