

§A Module Description is a concise description of the module. It should list the learning outcomes and describe the means by which these are achieved and demonstrated. You should bear in mind that the audience of the module description will be primarily students. Please complete the fields below, ensuring that the grey shaded text is deleted once the form is complete.



Computing for Mathematics

In the modern world it is imperative for a mathematician to know how to program. This module will give students an introduction to general concepts of programming that should empower them through their degree and beyond.

This module will introduce Students to programming through Python. The module will also teach particularities of programming applied to mathematics through Sage; an open source mathematics package built on Python.

School of Mathematics
Module Code
External Subject Code
Number of Credits 20
Level
Module Leader

Module offered on a free-standing basis?

Please enter yes if this module is offered on a free-standing basis. For details of free-standing modules, click [here](#).

Any restrictions to free-standing basis?

Please identify any restrictions here – these include levels of competency etc.

Maximum Number on Module

Language of module delivery **English**

On completion of the module a student will be able to:

- Understand and be able to write in Python the following programming ideas: Conditional Statements; Flow Control; Data Structures; Recurrence, Basic ideas of Object Orientated Programming.
- Use the above and a Mathematics package (Sage) to tackle mathematical problems.
- Have a basic knowledge of LaTeX.
- Work in groups to tackle problems and convey solutions to those problems through presentation.

How the module will be delivered

The module will be delivered over two terms. The first term will be used to deliver most of the syllabus through contact time. In particular through lectures and labs sessions. Importantly, an emphasis will be given to labs where the students will be able to learn through doing. The second term will

also have some contact time but the majority of delivery will be through self study and group work.

In the first semester:

- Lectures: approximately 11 hours
- Lab Sessions: approximately 22 hours

In the second semester:

- Lectures: approximately 7 hours

Skills that will be practised and developed

The following skills will be practised and developed:

- Programming;
- Group Work;
- Research;
- Presentation

How the module will be assessed

The module will be assessed in two parts. The first semester will make use of individual assessments (class exercise and course work) to assess the content delivered. The second part of the assessment will be through a group project done over the entire second semester. Students will be asked to build a program to solve/illustrate a particular mathematical problem (of the groups' choosing). The assessment will be based on difficulty, output (program and written report) and on a final presentation. The students will be expected to work the equivalent of 5 credits for this part of the assessment. The first two learning outcomes will be assessed with the first part of the assessment whilst all of them will be assessed with the second part.

Type of assessment	% Contribution	Title	Duration (if applicable)	Approx. date of Assessment
Class Test	40%	"Programming with Python class test"	1hr	Half way through 1 st semester
Individual Course Work	30%	"Using Sage to solve a mathematical problem course work"		End of 1 st semester
Group Project	30%	"Building a program to solve a mathematical problem course work with write up in LaTeX"		End of 2 nd semester

The potential for reassessment in this module

This module will be reassessed through an individual course work assignment and a class test to be carried out over the summer.

TBC – By the School

Syllabus content

- Basic programming
- Conditional statements
- Flow control
- Data structures
- Sorting algorithms
- Searching algorithms
- Recurrence
- Basic Objects Orientated Programming
- Solving Algebraic equations using Sage
- Handling Series using Sage
- Drawing plots using Sage
- Symbolic calculus using Sage
- LaTeX
- Library Skills
- PDP
- Careers
- Enterprise

Indicative Reading and Resource List:

The course is self contained but the following are recommended:

- A byte of Python
- Beginning Python: from Novice to Expert
- Learning Python
- Sage Beginner's Guide
- Sage Documentation
- More Math Into LaTeX: A Guide for Documentation and Presentation