MODULE DESCRIPTOR

TITLE	Mathematics for Software Engineering			
SI MODULE CODE	12-4501			
CREDITS	20			
LEVEL	4			
JACS CODE	*G120*			
SUBJECT GROUP	Mathematics			
DEPARTMENT	Engineering and Mathematics			
MODULE LEADER	David Cooper			
NOTIONAL STUDY HOURS BY TYPE	Tutor-led	Tutor-directed	Self-directed	Total Hours
	48	96	56	200

MODULE AIM(S)

The module aims to consolidate the students' knowledge of basic numerical skills, provide a foundation of mathematical and statistical knowledge, skills and concepts for application in software engineering and emphasise the fundamental rule of mathematics in computing.

MODULE LEARNING OUTCOMES

By engaging successfully with this module a student will be able to

1. demonstrate accurate use of basic numerical, algebraic and geometrical skills,

2. use and apply the properties of discrete mathematical structures in a computing setting,

3. understand and apply appropriate mathematical structures in software engineering.

INDICATIVE CONTENT

Elementary numerical and algebraic processes and two-dimensional geometry.
Standard functions: polynomials, trigonometric functions.
Computer arithmetic.
Introduction to set theory and logic.
Basic probability and statistics.
Functions and recursion.
Introduction to formal grammars.
Data organisation: data structures, e.g., arrays, lists and trees.
Matrices - basic ideas and concepts.

All topics will be applied to relevant problem areas.

LEARNING AND TEACHING METHODS

Students will be supported by lectures and tutorials with notes and worksheets provided. Further information will be published via Blackboard. Additional drop-in Maths Help is available. Students will learn by doing mathematics.

Summative continuous assessment on this module is also formative, with feedback provided. Further feedback will be provided to students through their work in tutorials.

ASSESSMENT STRATEGY AND METHODS

Task No.	TASK DESCRIPTION	SI Code	Task Weighting	Word Count / Duration	In-module retrieval available
1	Coursework	CW	60%		Y
2	Examination	EX	40%	60 min + 90 min	N

ASSESSMENT CRITERIA

In order to pass the module you will:

1. demonstrate the use of basic numerical, algebraic and geometrical skills,

2. use and apply the properties of discrete mathematical structures in a computing setting,

3. understand and apply appropriate mathematical structures in software engineering.

In order to achieve the equivalent of a first class mark in the module you will:

1. clearly demonstrate accurate use of basic numerical, algebraic and geometrical skills,

2. use and accurately apply the properties of discrete mathematical structures in a computing setting,

3. show an excellent understanding of appropriate mathematical structures and their application in software engineering.

FEEDBACK

Students will receive feedback on their performance in the following ways:

- Questions will be set which relate to each of the topics covered in the lectures. These will not be formally assessed but will enable students to evaluate their learning on each topic. Tutorials can be used to discuss the solutions to these questions and for other self-directed learning to be identified if required.
- The coursework will consist of several stages designed to allow the tutor to provide feedback on each part. Feedback will be via a mixture of written comments and marking and verbal feedback, enabling students to further explore how their performance can be improved in subsequent parts of the assessment.
- Blackboard will be used to provide students with performance grades and assessment feedback, as well as additional information that may be requested as a result of student feedback throughout the progress of the module.

LEARNING RESOURCES (INCLUDING READING LISTS)

Course lecture notes and tutorial sheets.

D Booth (1998) Foundation Mathematics, Addison Wesley.

N Dean (1997) The Essence of Discrete Mathematics, Prentice Hall.

JT Callender and R Jackson (1995) Exploring Probability and Statistics by Spreadsheet, Prentice-Hall.

L Goldschlager and A Lister (1988) Computer science: a modern introduction, 2nd ed., Prentice Hall.

D Harel (1992) Algorithmics : the spirit of computing, 2nd ed., Addison Wesley.

REVISIONS

Date	Reason
July 2012	Assessment Framework review

SECTION 2 'MODEL A' MODULE (INFORMATION FOR STAFF ONLY)

MODULE DELIVERY AND ASSESSMENT MANAGEMENT INFORMATION

MODULE STATUS - INDICATE IF ANY CHANGES BEING MADE

NEW MODULE	N
EXISTING MODULE - NO CHANGE	Y
Title Change	
Level Change	
Credit Change	
Assessment Pattern Change	
Change to Delivery Pattern	
Date the changes (or new module) will be implemented	**09/2012**

MODULE DELIVERY PATTERN - *Give details of the module delivery pattern. If the course has more than one intake, for example, September and January, please give details of the module start and end dates for each intake.*

	Module Begins	**Module Ends**
Course Intake 1		
Course Intake 2		
Course Intake 3		

Is timetabled contact time required for this module?	Y

Are any staff teaching on this module non-SHU employees?	N
If yes, please give details of the employer institution(s) below	
What proportion of the module is taught by these non-SHU staff, expressed as a percentage?	

MODULE ASSESSMENT INFORMATION

Does the Module (using Model A Assessment Pattern) Require Either*	
Overall Percentage Mark of 40%	Y
Overall Pass / Fail Grade	N

NB: Choose one of the above – Model A module cannot include both percentage mark and pass/fail graded tasks

FINAL TASK

According to the Assessment Strategy shown in the Module Descriptor, which task will be the LAST TASK to be taken or handed-in? (Give task number as shown in the Assessment Strategy)	Task No. 2

MODULE REFERRAL STRATEGY

Task for Task (as shown for initial assessment strategy)	Y
Single Referral Package for All Referred Students	N