IMPERIAL

Module Specification

Basic details

UID		Cohorts covered (Earliest cohort)	
Long title	Discrete Mathematics		
New code	ELEC50006	New short title	Discrete Maths
Brief description of module *(approx. 600 chars.)*	Computer engineers need to bridge the analogue/digital divide, and must be able to apply mathematical techniques from both discrete and continuous maths. This module will introduce you to aspects of discrete mathematics and abstract algorithms. It is not intended to cover all of computer science theory, but will provide you with the necessary theoretical background to allow you to work closely with computer scientists and apply applied and theoretical methods from computer science.		
Available as a standalone module/ short course?	N		

Statutory details

	ECTS	CATS	Non-credit	
Credit value	5	10	N	HECOS codes
FHEQ level	Level 5			

Allocation of study hours

	Hours	
Lectures	15	
Group teaching	7	*Incl. seminars, tutorials, problem classes.*
Lab/ practical		
Other scheduled		*Incl. project supervision, fieldwork, external visits.*
Independent study	103	*Incl. wider reading/ practice, follow-up work, completion of assess*
Placement		*Incl. work-based learning and study that occurs overseas.*
Total hours	125	
ECTS ratio	25.00	

Project/placement activity

Is placement activity allowed?	No

Module delivery

Delivery mode	Taught/ Campus	Other	
Delivery term	Term 1	Other	

Ownership

Primary department	Electrical and Electronic Engineering
Additional teaching departments	

Delivery campus	South Kensington

Collaborative delivery

Collaborative delivery?	N
External institution	N/A
External department	N/A
External campus	N/A

Associated staff

Role	CID	Given name	Surname
Module Leader		Dan	Goodman

Learning and teaching

Module description

Learning outcomes	Upon completion of this module, you will be able to: 1. Apply graph algorithms to efficiently explore data-structures 2. Prove algorithms are correct using inductive logic 3. Describe computability in terms of turing machines 4. Use complexity notation to analyse running time and storage requirements of a computer program. 5. Use the divide and conquer strategy for designing algorithms, and analyse their efficiency in terms of complexity notation using the Master Theorem.
Module content	- Big-O and complexity classes - Graph algorithms - Inductive proofs - Dynamic programming - Sets and numbers - Computability
Learning and Teaching Approach	You will be taught through lectures in a large lecture theatre to the whole year group using a classical blackboard and chalk approach. These lectures will be recorded. Exercise classes consist of thirty-sixty students will focus on applying the techniques covered in lectures to a variety of problems ranging from the easy to the difficult, with the chance to ask questions and get one-to-one help. These classes will involve both individual as well as group work. In tutorial groups of 3-4 students you will have the opportunity to solve a selection of problems with your personal tutor. Laboratory exercises will link the topics taught directly to their applications. Simulations will be part of the learning in applied settings. Workshops will be used to go over examples and maths-in-action workbooks.
Assessment Strategy	There will be one on-line tests one during reading week. The final exam will cover all the topics in the module.
Feedback	Feedback will be given in the exercise classes. The on-line tests will generate automatic feedback. Feedback on the exams will be giving in after re-assessments have been given. This feedback will come in the form of annotated example answers. The annotations will highlight where the student cohort did well and where general problems in understanding were observed.

Reading list	
Required equipment/ software	NA

Quality assurance

Date of first approval	
Date of last revision	
Date of this approval	July 2025
Module leader	Dan Goodman
Notes/ comments	

Office use only

QA Lead	
Department staff	
Date of collection	
Date exported	
Date imported	

Template version

Assessment details

Grading method	Numeric	Pass mark	40%

Assessments

Assessment type	Assessment description	Weighting	Pass mark	Must pass?
Examination	autumn midterm computer based test	20%	40%	N
Examination	exam (summer)	80%	40%	N
		100%		