COURSE SPECIFICATION FORM

for new course proposals and course amendments

Department/School:	**Mathematics**	**Academic Session:**	**2017-18**
Course Title:	Complexity Theory	**Course Value:** (UG courses = unit value, PG courses = notional learning hours)	0.5 unit
Course Code:	MT3130	**Course JACS Code:** (Please contact Data Management for advice)	G100
Availability: (Please state which teaching terms)	Term 1	**Status:**	Optional Condonable
Pre-requisites:	MT2630	**Co-requisites:**	-
Co-ordinator:	-		
Course Staff:	-		
Aims:	To introduce the technical skills to enable the student to understand the different classes of computational complexity, recognise when different problems have different computational hardness, and to be able to deduce cryptographic properties of related algorithms and protocols.		
Learning Outcomes:	1. understand the formal definition of algorithms and Turing machines 2. understand that not all languages are computable and prove simple examples 3. organise the low-level complexity classes (P, NP, coNP, NP-complete, RP, ZPP, BPP, PSPACE) into a hierarchy and prove simple languages exist in each class 4. give examples of one-way functions and hardcore functions, and demonstrate that every NP function has a hardcore predicate 5. use complexity theoretic techniques as a method of analysing communication services		
Course Content:	Algorithms: Motivation for complexity; languages; deterministic Turing machines; Church-Turing thesis; randomised algorithms. Computability: Goedel numbers; incomputable languages. Low-level complexity classes: Class P; 2-SAT; class NP; Cook’s theorem; 3-SAT; coNP; class RP; class BPP; probability amplification; relation between classes; class PSPACE. One-way functions: One-way functions; one-way permutations; trapdoors; hardcore functions; Goldreich-Levin theorem Applications of complexity theory to communication: Applications of complexity theory to analysing the efficiency of communications’ services.		
Teaching & Learning Methods:	The total number of notional learning hours associated with this course are 150. 3 hours of lectures a week over 11 weeks. 33 hours total. 117 hours of private study, including work on problem sheets and examination preparation. This may include discussions with the course leader if the student wishes.		
Key Bibliography:	Complexity and cryptography by Talbot and Welsh (001.5436 TAL) Introduction to the theory of complexity by Bouvet and Crescenzi (519.22 BOV) Foundations of cryptography by Goldreich (001.5436 GOL)		
Formative Assessment & Feedback:	Formative assignments in the form of 8 problem sheets. The students will receive feedback as written comments on their attempts.		
Summative Assessment:	**Exam:** 100% Written exam. A two hour paper. **Coursework:** None		

Updated September 2017

The information contained in this course outline is correct at the time of publication, but may be subject to change as part of the Department’s policy of continuous improvement and development. Every effort will be made to notify you of any such changes.