Imperial College London

DEPARTMENTS OF MATHEMATICS AND COMPUTING

MATHEMATICS AND COMPUTER SCIENCE
(BSc GG14, MSci GG41)

GUIDE TO COURSES

THIRD/FINAL YEAR (BSc), THIRD YEAR (MSci)
2012—2013

Notes and syllabus details on second and Third Year/level courses and Fourth Year course intentions for students in their Third Year

Prof. *Jonathan Mestel*
Director of Undergraduate Studies
Mathematics

Dr. Jeremy Bradley
Course Director <jmc@doc.ic.ac.uk>
Computing

October 2012 (v3)

The information given is current at this date and may be subject to alteration

This guide gives course information about all courses which have been validated at this level. A selection is made at each session to give students a suitable set of course from which to make their choice.

All Departmental course documents may be viewed via the Departmental undergraduate webpage at:

http://www3.imperial.ac.uk/mathematics/students/undergraduate/courseguides

NOTES ON THIRD AND FOURTH YEAR COURSES

Each Third Year/Fourth Year Mathematics and each Third Year/Fourth Year Computing course carries a half unit of credit. Students may normally choose *new courses up to a value of four course units* from the available courses in their Third Year of study. MSci students choose a further four course units in their Fourth Year of study.

Please note that the choice of options of all students is subject to the approval of the Director of Studies (Mathematics) and Course Director (Computing).

FOR STUDENTS ENTERING IMPERIAL COLLEGE FROM OCTOBER 2008, IT IS NORMALLY REQUIRED FOR PASSES IN ALL COURSE ELEMENTS (HERE HALF UNITS) TO BE ACHIEVED FOR GRADUATION.

Students at graduation may be awarded Honours degrees classified as follows: First, Second (upper and lower divisions) and Third.

Year weightings for Honours: BSc (GG14) 1 : 3 : 4 and MSci (GG41) 1 : 2 : 3 : 4

THIRD YEAR (BSc)

In the Third Year, a total of 7 half-unit courses are taken. In addition, there is also a COMPULSORY INDIVIDUAL PROJECT C376 which carries a half unit of credit (with enhanced weighting) and a COMPULSORY COMPUTING GROUP PROJECT. The computing group project will also incorporate attendance at lectures and assessment of the material included in the course C302 Software Engineering Methods (C302 does not count as one of the Computing half-unit courses).

(M) - At least 2 and at most 5 Mathematics courses are to be chosen from the overall list of courses which are made available. This includes the Mathematics courses listed below as available in Second Year.

The list of suggested course options for Mathematics is given below. If you wish to take a third year Maths option that is not listed and the timetabling allows it, then you may do so, but you have to inform the Director of Studies in Maths and the JMC Course Director.

SECOND LEVEL COURSES	
M2PM5	Metric Spaces and Topology
M2AA1	Differential Equations

THIRD LEVEL COURSES

NUMERICAL/APPLIED ANALYSIS	
M3N7*	Numerical Solution of Ordinary Differential Equations
M3N10*	Computational Partial Differential Equations I

PURE MATHEMATICS	
Geometry	
M3P5*	Geometry of Curves and Surfaces
M3P20*	Geometry I: Algebraic Curves
M3P21*	Geometry II: Algebraic Topology
M3P23*	Computational Algebra and Geometry
Algebra and Discrete Mathematics	
M3P6*	Probability Theory
M3P7*	Functional Analysis
M3P8*	Algebra III
M3P10*	Group Theory
M3P11*	Galois Theory
M3P12*	Group Representation Theory
M3P17*	Algebraic Combinatorics
M3P18*	Fourier Analysis and Theory of Distributions
M3P19*	Measure and Integration
Number Theory	
M3P14*	Number Theory
M3P15*	Algebraic Number Theory
M3P16*	Analytic Number Theory

STATISTICS	
M2S1	Probability and Statistics II
M2S2	Statistical Modelling I
M3S1*	Statistical Theory I
M3S2*	Statistical Modelling II
M3S4*	Applied Probability
M3S7*	Statistical Pattern Recognition
M3S8*	Time Series
M3S9*	Stochastic Simulation
M3S11*	Games, Risks and Decisions
M3S14*	Survival Models and Actuarial Applications
M3S16*	Credit Scoring I

M3S17*	Credit Scoring II

OTHER MATHEMATICAL COURSE	
M3T	Communicating mathematics

* indicates that a course is available in enhanced form as part of the Fourth Year programme for MSci students [see page 8].

(C) - At least 2 and at most 5 Computing courses are to be chosen from the following:

C303	Software Engineering - Systems Verification	Term2
C317	Graphics	Term 2
C318	Custom Computing	Term 2
C320	Complex Systems	Term 1
C322	Communicating Computer Science in Schools	Terms 1,2,3
C330	Network Security	Term 1
C332	Advanced Computer Architecture	Term 2
C333	Robotics	Term 2
C335	Distributed Systems	Term 1
C337	Simulation and Modelling	Term 1
C341	Introduction to Bioinformatics	Term 2
C343	Operations Research	Term 1
C382	Type Systems for Programming Languages	Term 1
C395	Machine Learning	Term 1
C526	Databases	Term 2
C527	Computer networks and Distributed systems	Term 2
C528	Concurrent Programming	Term 2

NOTE:

(1) Recommended prerequisites in Computing are given as a guide to the background material required. Without these, in some cases you may need to do some extra background reading in order to fully understand the material. However you may still take a course without sitting the recommended prerequisite.

(2) C221, C526 and C527 may not be taken in both Year 2 and Year 3
C528 may not be taken in Year 3 if C223 was taken in Year 2
C335 and C527 are mutually exclusive

(3) It is recommended that each student normally take no fewer than 3 and no more than 5 optional courses in Term 1 (excluding the project)

OTHER COURSES

H01	Philosophy I	1 + seminars	Terms 1 & 2
H04	Controversies and Ethical Dilemmas in Science and Technology	1 + seminars	Terms 1 & 2
H05	European History : 1870 – 1989	1 + seminars	Terms 1 & 2
H06	Politics	1 + seminars	Terms 1 & 2
H07	History of Science	1 + seminars	Terms 1 & 2
H08	Global History of Twentieth Century Things	1 + seminars	Terms 1 & 2
H09	History of Medicine	1 + seminars	Terms 1 & 2
H12	Music and Western Civilisation	1 + seminars	Terms 1 & 2
H13	Communicating Science: The Public and the Media	1 + seminars	Terms 1 & 2
H15	Humanities Essay	Tutorials + private study	Terms 1 & 2
H19	Creative Writing	1 + tutorials	Terms 1 & 2
H21	Music Technology	2 + private study	Terms 1 & 2
H22	Philosophies of Science: Theory, Society and Communications	1 + seminars	Terms 1 & 2
BS0806	Entrepreneurship	1 + seminars/tutorial	Term 1
BS0808	Finance and Financial Management	1 + seminars/tutorials	Term 2
BS0820	Innovation Management	1 + seminars/tutorials	Term 1
BS0815	Managerial Economics	1 + seminars/tutorials	Term 1
BS0821	Project Management	1 + seminars/tutorials	Term 2

NOTE:

- **Humanities courses extend throughout Terms 1 <u>and</u> 2**
- **BS0806 (Entrepreneurship) is permitted for THIRD YEAR BSc (GG14) STUDENTS ONLY**
- **H15 Humanities Essay normally requires permission from the Humanities Programme and approval by the Department of Mathematics**

STUDENTS ARE ENCOURAGED TO CONSIDER BROADENING THEIR STUDY PROGRAMME BY TAKING ADVANTAGE OF THIS HUMANITIES/MANAGEMENT PROVISION.
IF THEY SO WISH, THIRD YEAR STUDENTS MAY OFFER A HALF-UNIT OPTION FROM THIS LIST.

It is important to note that places in Humanities/Management courses are normally limited – individual bookings for them should be done separately for these courses on the forms at the Humanities and Business School websites <u>before the end of the previous session</u>.

The Department always endeavours to avoid timetabling Mathematics courses during the allocated 12-2 College slots reserved for the above courses. However, timetabling by the Humanities or Business School outside these periods cannot be so protected by the Departments.

For syllabus and timetabling constraints, see the Humanities and Imperial College Business School entries in the College calendar.

THIRD YEAR (MSci)

Students should be considering their programme for Third and Fourth Year as a whole with the help of these notes and advice from the two Departments.

In the Third Year a total of 8 half-unit courses are taken. Students are required to offer M2S1 Probability and Statistics II as part of their Third Year programme if they are intending to take further M3 or M4 statistics courses. There is also a compulsory Computing group project in Term 1, and a compulsory industrial placement from the beginning of June until the end of September. The Computing Group Project also incorporates attendance at lectures and assessment of the material included in the course C302 Software Engineering Methods (C302 does not count as one of the Computing half-unit courses). The industrial placement is organised and managed by the Computing Department and forms part of the Fourth Year half unit project.

(M) - At least 2 and at most 6 additional Mathematics courses are to be chosen from the overall list of courses made available. The full listing is as given above for THIRD YEAR (BSc).

(C) - At least 2 and at most 6 additional Computing courses are to be chosen from the overall list of courses made available. The full listing is as given above for THIRD YEAR (BSc).

NOTE:

(1) Recommended prerequisites in Computing are given as a guide to the background material required. Without these, in some cases you may need to do some extra background reading in order to fully understand the material. However you may still take a course without sitting the recommended prerequisite.

(2) C221, C526 and C527 may not be taken in both Year 2 and Year 3
C320, C330, C332, C382 cannot be taken in both years 3 and 4
C528 may not be taken in Year 3 if C223 was taken in Year 2
C335 and C527 are mutually exclusive

(3) It is recommended that each student normally take no fewer than 1 and no more than 4 optional courses in Term 1

(H) – A half unit Humanities option is permitted to be taken as part of MSci third year. The full listing is as given for the third year with the exception of BS0806 Entrepreneurship.

FOURTH YEAR (MSci)

Students will take 7 half unit courses, including at least 2 from Mathematics and at least 2 from Computing. Students also submit a compulsory half-unit individual project, either in Mathematics or Computing with enhanced weighting.

(M)

The full programme of courses to be available to Third and Fourth Year students in future sessions is under continuous review. Many courses are intended to run every session, but others do tend to run in alternate sessions.

Some courses are available in Fourth Year in M4 "...with Advanced Study" form. The enhancement is in the form of an additional masters level exam question for each Advanced Study option taken on top of the base M3 exam. This enhancement is not part of the M3 assessment, only part of that for M4. Naturally students may not offer such an M4 course if they have previously taken the M3 version.

Other courses are not available before the Fourth Year stage – these M4 courses have a strong overlap with the PG taught course MSc courses in Applied Mathematics, Pure Mathematics.

MSci Fourth Year students are expected normally to offer all of their course choices in M4 form. It is expected that, with Departmental permission, there may be dispensation with respect to one M3 course, but only if a corresponding M4 course is not available. Fourth Year students do, of course, complete in addition a research project.

M4 COURSES

It is not possible to be completely specific at this stage about courses expected to run in the next session. The teaching plans are usually reaching a conclusion at about Easter for the following session, in the light of staffing availability and student choice.

The current FOURTH YEAR structure is contained in the appropriate MSci 4 document from the Maths department. Current Sectional policy is to have available a good range of high level options each session, so that MSci students should have a good choice open to them in **each** of their last two years of study.

Below is a list of suggested Mathematics courses for joint Mathematics and Computing students in J4. If however, a JMC student wishes to take another M4 course and the timetable allows it, they may take that course as one of their half units (please let the JMC Course Director know first). In taking one of these unlisted M4 courses, you may have to do extra preparatory work to cover material that may not have been in J1—J3.

NUMERICAL/APPLIED ANALYSIS	
M4N7	Numerical Solution of Ordinary Differential Equations with Advanced Study
M4N10	Computational Partial Differential Equations I with Advanced Study

PURE MATHEMATICS	
GEOMETRY	
M4P5	Geometry of Curves and Surfaces with Advanced Study
M4P20	Geometry I: Algebraic Curves with Advanced Study
M4P21	Geometry II: Algebraic Topology with Advanced Study
M4P51	Riemannian Geometry
M4P52	Manifolds

M4P54	Differential Topology
M4P57	Complex Manifolds
ALGEBRA AND DISCRETE MATHEMATICS	
M4P8	Algebra III with Advanced Study
M4P10	Group Theory with Advanced Study
M4P11	Galois Theory with Advanced Study
M4P12	Group Representation Theory with Advanced Study
M4P17	Algebraic Combinatorics with Advanced Study
M4P23	Computational Algebra and Geometry with Advanced Study*
M4P34	Groups and Representations
M4P46	Lie Algebras
M4P55	Commutative Algebra
NUMBER THEORY	
M4P14	Elementary Number Theory with Advanced Study
M4P15	Algebraic Number Theory with Advanced Study
M4P16	Analytic Number Theory with Advanced Study
M4P32	Number Theory: Elliptic Curves
OTHER	
M4P6	Probability Theory with Advanced Study*
M4P7	Functional Analysis with Advanced Study*
M4P18	Fourier Analysis and Theory of Distributions with Advanced Study*
M4P19	Measure and Integration with Advanced Study*
M4T	Communicating Mathematics

STATISTICS	
M4S1	Statistical Theory I with Advanced Study
M4S2	Statistical Modelling II with Advanced Study
M4S4	Applied Probability with Advanced Study
M4S7	Statistical Pattern Recognition with Advanced Study
M4S8	Time Series with Advanced Study
M4S9	Stochastic Simulation with Advanced Study
M4S11	Games, Risks and Decisions
M4S14	Survival Models and Actuarial Applications with Advanced Study
M4S16	Credit Scoring I with Advanced Study
M4S17	Credit Scoring II with Advanced Study
M4A42	Applied Stochastic Processes

C4 COURSES

Students should select Computing courses from the following list. Recommended prerequisites are indicated with the syllabus details in Appendix C below; see Note (1) below.

C320	Complex Systems		Term 1
C330	Network Security	Recommended: Distributed Systems (C335)	Term 1
C332	Advanced Computer Architecture		Term 2
C382	Type Systems for Programming Languages		Term 1
C417	Advanced Graphics and Visualisation	Recommended: Graphics (C317)	Term 2
C418	Computer Vision	Recommended: Graphics (C317)	Term 1
C421	Computational Neurodynamics		Term 1
C422	Computational Finance	Recommended: Operations Research (C343)	Term 2

C424	Machine Learning and Neural Computation		Term 1
C429	Parallel Algorithms	Recommended: Distributed Systems (C335) or Simulation & Modelling (C337)	Term 2
C436	Performance Analysis	Recommended: Simulation & Modelling (C337). Exclusive with M3S4 Applied Probability I	Term 2
C437	Distributed Algorithms	Recommended: Concurrency (C223), Distributed Systems (C335)	Term 2
C438	Complexity	Recommended: Models of Computation (C240) or (M20D)	Term 2
C470	Program Analysis		Term 2
C471	Advanced Issues in Object Oriented Programming	Recommended: Type Systems for Programming Languages (C382)	Term 1
C474	Multi-agent Systems	Recommended: Introduction to AI (C231) & Prolog	Term 2
C475	Software Engineering for Industry		Term 1
C477	Computing for Optimal Decisions	Recommended: Operations Research (C343) or (M3N3)	Term 1
C480	Automated Reasoning	Recommended: Prolog	Term 1
C481	Models of Concurrent Computation	Recommended: Concurrency (C223)	Term 1
C484	Quantum Computing		Term 2
C491	Knowledge Representation		Term 2
C493	Intelligent Data and Probabilistic Inference		Term 2
C499	Modal and Temporal Logic		Term 1

NOTE:

(1) Recommended prerequisites in Computing are given as a guide to the background material required. Without these, in some cases you may need to do some extra background reading in order to fully understand the material. However you may still take a course without sitting the recommended prerequisite.

(2) C320, C330, C332, C382 cannot be taken in both years 3 and 4

(3) C382 cannot be taken in Year 4 if C482 has been taken in Year 3

Some further information is given below. Students may well also find it useful to consult overall and year-specific course documents for other degree codings within the two Departments and the College Calendar.

Students should certainly feel free to seek advice from Personal Tutors and Senior Tutors, together with the Course Directors and Directors of Studies in the two Departments.

APPENDIX (M)

M2PM5 METRIC SPACES AND TOPOLOGY

Dr. Krasovsky **Term 2**

Metric spaces. Convergence and continuity. Examples (Euclidean spaces, function spaces; uniform convergence). The open sets in a metric space; equivalent metrics. Convergence and continuity in terms of open sets: topological spaces. Subspaces. Hausdorff spaces. Sequential compactness; compactness via open covers; compact spaces; determination of compact subspaces of R^n. Completeness in metric spaces. Relationship between compactness and completeness. Connected and path connected spaces; equivalence of these notions for open sets in Rn. Winding numbers, definition of fundamental group, its computation for the circle. Example: proof of fundamental theorem of algebra.

M2AA1 DIFFERENTIAL EQUATIONS

Dr. Van Strien **Term 2**

Differential equations and systems have a rich structure, which is explored in this course using ideas from analysis, algebra and geometry. There is a similar theory for discrete systems (maps).

Scalar, system; linear, nonlinear; first-order, higher-order
Homogeneous, autonomous
1st order constant coefficient systems:
Matrix exponential
Laplace transform
1st order linear systems: Fundamental solution matrix; periodic problems, Floquet theory
Power series solutions
1st order nonlinear systems: Contractions, existence/uniqueness
Numerical methods
Boundary value problems: Sturm-Liouville, Green's functions
Dynamical systems (1st order autonomous systems): stationary solutions, periodic orbits; hyperbolicity, stability; bifurcations.

NUMERICAL/APPLIED ANALYSIS

M3N7 NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS
Also M4N7

Dr. D.R. Moore
3 (Term 1)

An analysis of methods for solving ordinary differential equations. Wholly examined by project.

Runge-Kutta, extrapolation and linear multistep methods. Analysis of stability and convergence.
Error estimation and automatic step control. Introduction to stiffness.
Boundary and eigenvalue problems. Solution by shooting and finite difference methods.
Introduction to deferred and defect correction.

M3N10 COMPUTATIONAL PARTIAL DIFFERENTIAL EQUATIONS I
Also M4N10

Dr. Lushi **Term 2**

An introduction to the finite element method for approximating PDEs.

Topics:
Variational principles, weak formulation. Essential and natural boundary conditions. Galerkin method. Conforming elements in one and higher space dimensions. Approximation and error analysis. Functional analysis setting. Computational aspects. Variational crimes.
Non-self adjoint problems. Time dependent problems. Stability analysis

PURE MATHEMATICS

GEOMETRY

M3P5 GEOMETRY OF CURVES AND SURFACES
Also M4P5

Dr. A. Neves **Term 2**

The main object of this course is to understand what is the curvature of a surface in 3-dimensional space.

Topological Surfaces: Defintion of an atlas; the prototype definition of a surface; examples.

The topology of a surface; the Hausdorff condition, the genuine definition of a surface.

Orientability, compactness. Subdivisions and the Euler characteristic.

Cut-and-paste technique. The classification of compact surfaces. Connected sums of surfaces.

Smooth Surfaces: Definition of a smooth atlas, a smooth surface and of smooth maps into and out of smooth surfaces. Surfaces in IR^3**,**tangents, normals and orientability.

The first fundamental form, lengths and areas, isometries.

The second fundamental form, principal curvatures and directions.

The definition of a geodesic, existence and uniqueness, geodesics and coordinates.

Gaussian curvature, definition and geometric interpretation, Gauss curvature is intrinsic, surfaces with constant Gauss curvature. The Gauss-Bonnet theorem

(Not examinable and in brief) Abstract Riemannian surfaces, metrics.

M3P20 GEOMETRY I: ALGEBRAIC CURVES
Also M4P20

Dr. Hein **Term 1**

Plane algebraic curves including inflection points, singular and non-singular points, rational parametrisation, Weierstrass form and the Group Law on non-singular cubics. Abstract complex manifolds of dimension 1 (Riemann surfaces); elliptic curves as quotients of C by a lattice. Elliptic integrals and Abel’s theorem.

M3P21 GEOMETRY II: ALGEBRAIC TOPOLOGY
Also M4P21

Dr. Kaloghiros **Term 2**

Homotopies of maps and spaces. Fundamental group. Covering spaces, Van Kampen (only sketch of proof). Homology: singular and simplicial (following Hatcher’s notion of Delta-complex). Mayer-Vietoris (sketch proof) and long exact sequence of a pair. Calculations on topological surfaces. Brouwer fixed point theorem.

M3P23 COMPUTATIONAL ALGEBRA AND GEOMETRY
Also M4P23

Dr. Kasprzyk **Term 1**

Syllabus not available.

ALGEBRA AND DISCRETE MATHEMATICS

M3P6 PROBABILITY THEORY

Dr. Zegarlinski **Term 2**

Probability measures. Random variables Independence. Sums of independent random variables; weak and strong laws of large numbers. Weak convergence, characteristic functions, central limit theorem. Elements of Brownian motion.

M3P7 FUNCTIONAL ANALYSIS

Dr. Laptev **Term 1**

This course brings together ideas of continuity and linear algebra. It concerns vector spaces with a distance, and involves linear maps; the vector spaces are often spaces of functions.
Vector spaces. Existence of a Hamel basis. Normed vector spaces. Banach spaces. Finite dimensional spaces. Isomorphism. Separability. The Hilbert space. The Riesz-Fisher Theorem. The Hahn-Banach Theorem. Principle of Uniform Boundedness. Dual spaces. Operators, compact operators. Hermitian operators and the Spectral Theorem.

M3P8 ALGEBRA III

Professor K. Buzzard **Term 1**

Rings, integral domains, unique factorization domains
Modules, ideals homomorphisms, quotient rings, submodules quotient modules
Fields, maximal ideals, prime ideals, principal ideal domains
Euclidean domains, rings of polynomials, Gauss's lemma, Eisenstein's criterion
Field extensions
Noetherian rings and Hilbert's basis theorem
Dual vector space, tensor algebra and Hom
Basics of homological algebra, complexes and exact sequences

M3P10 GROUP THEORY
Also M4P10

Dr. N. Britnell **Term 1**

An introduction to some of the more advanced topics in the theory of groups.

Composition series, Jordan-Hölder theorem, Sylow's theorems, nilpotent and soluble groups
Permutation groups. Types of simple groups

M3P11 GALOIS THEORY
Also M4P11

Professor M.W. Liebeck **Term 1**

The formula for the solution to a quadratic equation is well known. There are similar formulae for cubic and quartic equations, but no formula is possible for quintics. The course explains why this happens.

Irreducible polynomials. Field extensions, degrees and the tower law. Extending isomorphisms
Normal field extensions, splitting fields, separable extensions. The theorem of the primitive element
Groups of automorphisms, fixed fields. The fundamental theorem of Galois theory
The solubility of polynomials of degree at most 4. The insolubility of quintic equations

M3P12 GROUP REPRESENTATION THEORY
Also M4P12

Dr. E. Segal **Term 2**

Although the underlying theory in this course concerns the ways in which a group can be written as a set of matrices, the main results involve group characters.

Some elementary group theory; examples of groups; conjugacy classes; Cayley's theorem
Representations of groups: definitions and basic properties. Maschke's theorem, Schur's lemma.
Representations of abelian groups. The character of a group representation. The group algebra and the regular character. Orthogonality relations
Character tables of various interesting groups
Construction of representations by using tensor products

M3P17 ALGEBRAIC COMBINATORICS
(Previously DISCRETE MATHEMATICS)
Also M4P17

Professor A.A. Ivanov **Term 2**

An introduction to a variety of combinatorial techniques which have wide applications to other areas of mathematics.

Elementary coding theory. The Hamming metric, linear codes and Hamming codes
Combinatorial structures: block designs, affine and projective planes. Construction of examples using finite fields and vector spaces. Steiner systems from the Golay code. Basic theory of incidence matrices
Strongly regular graphs: examples, basic theory, and relationship with codes and designs

M3P18 FOURIER ANALYSIS AND THOERY OF DISTRIBUTIONS
Also M4P18

Dr. Cuenin **Term 2**

Spaces of test functions and distributions, Fourier Transform (discrete and continuous), Bessel's, Parseval's Theorems, Laplace transform of a distribution, Solution of classical PDE's via Fourier transform, Basic Sobolev Inequalities, Sobolev spaces.

M3P19 MEASURE AND INTEGRATION
Also M4P19

Dr. Krasovsky **Term 1**

Lebesgue theory of measure of integration. Outer measure, Basic theorems of integration theory. Integration versus differentiation. Fubini-Tonelli Theorem and applications, Convergence almost everywhere. Borel-Cantelli Lemma, L_p spaces. Ergodic Theorem.

NUMBER THEORY

M3P14 NUMBER THEORY
Also M4P14

Dr. P. Cascini **Term 1**

The course is concerned with properties of natural numbers, and in particular of prime numbers, which can be proved by elementary methods.

Division and factorisation, fundamental theorem of arithmetic, Euclid's algorithm
Arithmetic functions, multiplicative functions, perfect numbers, Möbius inversion, Dirichlet convolution
Congruences, Fermat-Euler theorem, Chinese remainder theorem, Lagrange's theorem
Wilson's theorem
Primitive roots, Gauss's theorem, indices
Quadratic residues, Euler's criterion, Gauss's lemma, law of quadratic reciprocity, Jacobi symbol. Sums of squares. Distribution of quadratic residues and non-residues
Irrationality, Liouville's theorem, construction of a transcendental number

M3P15 ALGEBRAIC NUMBER THEORY
Also M4P15

Dr. Pal **Term 2**

An introduction to algebraic number theory, with emphasis on quadratic fields. In such fields the familiar unique factorisation enjoyed by the integers may fail, but the extent of the failure is measured by the class group.

The following topics will be treated with an emphasis on quadratic fields $Q(\sqrt{d})$:

Field extensions, minimum polynomial, algebraic numbers, conjugates and discriminants, Gaussian integers, algebraic integers, integral basis, quadratic fields, cyclotomic fields, norm of an algebraic number, existence of factorisation.Factorisation in $Q(\sqrt{d})$. Ideals, Z-basis, maximal ideals, prime ideals, unique factorisation theorem of ideals and consequences, relationship between factorisation of numbers and of ideals, norm of an ideal.
Ideal classes, finiteness of class number, computations of class number.

M3P16 ANALYTIC NUMBER THEORY
Also M4P16

Professor N. Bingham **Term 2**

Uniform distribution, Weyl's criterion, the sequence $\{na\}$.
Euler's proof that there are infinitely many primes. Von Mangoldt's function, Chebychev's Inequality, Merten's theorem.
Dirichlet series, generating functions.
Gauss sums, second proof of law of quadratic reciprocity.
Dirichlet characters. Dirichlet's theorem on primes in arithmetic progression.

Selberg’s upper bound sieve. Prime number theorem.

STATISTICS

M2S1 PROBABILITY AND STATISTICS II

Dr. Van Dyke **Term 1**

This course extends the ideas met in M1S and introduces the distributional results needed for the study of statistical inference and applied probability.
Probabilities for events. Random variables. Poisson process. Probability generating functions. Probability densities. Moment generating functions. Limit theorems. Inequalities. Joint distributions. Multinomial and bivariate normal distributions. Transformations. Order statistics. Ideas of statistics.

M2S2 STATISTICAL MODELLING I

Dr. A. Gandy **Term 2**

This course leads on from the Probability and Statistics covered in M2S1. Traditional concepts of statistical inference, including maximum likelihood, hypothesis testing and interval estimation are developed and then applied to the linear model which arises in many practical situations. The course also acts as an introduction to the course M3S7 (Applied Statistics).

Revision and extension of M2S1 statistical material - maximum likelihood estimation, likelihood ratio tests and their properties, confidence intervals.
Linear models (including non-full rank models): estimation, confidence intervals and hypothesis testing. The analysis of variance.

M3S1 STATISTICAL THEORY I
Also M4S1

Dr. R. Coleman **Term 1**

This course deals with the criteria and the theoretical results necessary to develop and evaluate optimum statistical procedures in hypothesis-testing, point and interval estimation. Knowledge of M2S1 is assumed and its results will be used.
Theories of estimation and hypothesis-testing, including sufficiency, completeness, exponential families, Cramér-Rao, Blackwell-Rao and Neyman-Pearson results. Elementary decision theory.

M3S2 STATISTICAL MODELLING II

Dr. C. Anagnostopoulos **Term 2**

This course leads on from the linear models covered in M2S2 and Probability and Statistics covered in M2S1. The Generalised Linear Model is introduced from a theoretical and practical viewpoint and various aspects are explained.

Generalised Linear Model, as a unifying statistical framework – linear models and quantitative responses. Nonlinear regression. Generalised Additive Models, Kernel Regression. Fixed and random effect models.

The R statistical package will be used to expose how the different models can be applied on example data.

M3S4 APPLIED PROBABILITY
Also M4S4

Dr. A. Veraart **Term 1**

This course aims to give students an understanding of the basics of stochastic processes. The theory of different kinds of processes will be described, and will be illustrated by applications in several areas. The groundwork will be laid for further deep work, especially in such areas as genetics, finance, industrial applications, and medicine.

Revision of basic ideas of probability. Important discrete and continuous probability distributions. Random processes: Bernoulli processes, simple birth processes, branching processing. Point processes. Stationarity. Non-homogeneous, compund, and doubly stochastic Poisson processes. Autocorrelation functions. Perturbed regular processes. Probability generating functions and how to use them. Branching processes and their properties. Galton-Watson model. Random walks. Gambler's ruin. Absorbing and reflecting barriers. Markov chains. Chapman-Kolmogorov equations. Recurrent, transient, periodic, aperiodic chains. Returning probabilities and times. The basic limit theorem. Birth processes.
Differential difference equations and pgfs. Finding pgfs. Birth, death, immigration processes. Embedded processes. Time to extinction. Queues. Specification. Long term behaviour. Traffic intensity. Waiting times. Steady states.

M3S7 STATISTICAL PATTERN RECOGNITION
Also M4S7

Dr. Cohen **Term 1**

A fundamental aim in statistics is classifying things. For example, this occurs in medicine, where the aim is to assign people to diseases, speech recognition, where the aim is to assign words to meanings, banking, where the aim is to assign customers to risk classes, and in a vast number of other areas. This course provides a modern view of methods for performing such so-called pattern recognition tasks.
Totally examined by projects.

Discriminant analysis, generalised linear models, nearest neighbour methods, recursive partitioning methods, trees, ensemble classifiers, and other tools, The areas of pre-processing, feature selection and classification performance assessment will also be explored, and an introduction to unsupervised pattern recognition methods will be given.

M3S8 TIME SERIES
Also M4S8

Professor A.T Walden **Term 1**

An introduction to the analysis of time series (series of observations, usually evolving in time) is given which gives weight to both the time domain and frequency domain viewpoints. Important structural features (e.g. reversibility) are discussed, and useful computational algorithms and approaches are introduced. The course is self-contained.

Discrete time stochastic processes and examples. ARMA processes. Trend removal and seasonal adjustment. General linear process. Invertibility. Directionality and reversibility in time series. Spectral representation. Aliasing. Generating functions. Estimation of mean and autocovariance sequence. The periodogram. Tapering for bias reduction. Parametric model fitting. Forecasting.

M3S9 STOCHASTIC SIMULATION
Also M4S9

Dr. Bottolo **Term 2**

Computational techniques are becoming increasingly important in statistics, especially in the simulation of stochastic systems (for example for testing new estimation methods). This course provides a modern view of such simulation methods covering areas from basic random variate generation to ways of describing images. Material from M2S1 would form a firm foundation.

Random numbers. Congruential and other generators. Techniques of random variate generation. Computer experiments. Monte Carlo integration. Variance reduction techniques. Markov random fields. Gibbs sampling. Applications in statistics, optimisation and operations research.

M3S11 GAMES, RISKS AND DECISIONS
Also M4S11

Dr. L. White **Term 2**

Simple probabilistic and mathematical tools are used to study the theory of games where two opponents are in conflict, including the celebrated Prisoners' Dilemma problem. Utilities, based on (apparently) reasonable axioms are introduced, leading to a study of decision theory.
Game Theory. Randomized strategies.. Admissibility. Bayes strategies. Utility theory. Risks and risk aversion. Decision Theory. Sequential Decision-making. Applications to problems in Economics, Biology, Medicine, Insurance and Investment.

M3S14 SURVIVAL MODELS AND ACTUARIAL APPLICATIONS
Also M4S14

Dr. N.A. Heard **Term 2**

Survival models are fundamental to actuarial work, as well as being a key concept in medical statistics. This course will introduce the ideas, placing particular emphasis on actuarial applications.

Explain concepts of survival models, right and left censored and randomly censored data. Describe estimation procedures for lifetime distributions: empirical survival functions, Kaplan-Meier estimates, Cox model.
Statistical models of transfers between multiple states, maximum likelihood estimators. Binomial model of mortality. Estimation of transition intensities which depend on age. Testing crude estimates for consistency. Discussion of simple assurance and annuity contracts. Computation of expected present values and variances of simple benefits. Description and calculation of net premiums and net premium policy values of simple assurance and annuity contracts.

M3S16 CREDIT SCORING I
Also M4S16

Dr. T. Bellotti **Term 1**

Introduction and background: Aims and objectives of scoring, legislative and commercial aspects.
Consumer credit data: characteristics, transformations, data quality, transaction types, challenges. Notions of statistical scorecards
Basic models: application and behavioural model types and characteristics, including segmented, dynamic, and stochastic process models

Estimation and model fitting: basic principles and model selection. Issues of model search space and selectivity bias.
Special models and methods: tailored methods for specific problems, such as existing model favouritism and the needle in the haystack problem in fraud detection
Performance assessment and monitoring: measures and estimation, method comparison

M3S17 CREDIT SCORING II
Also M4S17

Dr. T. Bellotti **Term 2**

Syllabus not available.

OTHER MATHEMATICAL COURSE

M3T COMMUNICATING MATHEMATICS

Dr. E. McCoy/Dr. L.V. White **Term 2/3**

This course will give students the opportunity to observe and assist with teaching of Mathematics in local schools. Entry to the course is by interview in the preceding June and numbers will be limited. For those selected there will follow a one-day training course in presentation skills and other aspects of teaching. Students will be assigned to a school where they will spend 10 half-days in the Spring Term, under the supervision of a teacher. Assessment will be based on a portfolio of activities in the school, a special project, evaluation by the school teacher and an oral presentation.

APPENDIX (C)

C302 SOFTWARE ENGINEERING – METHODS
[Part of Group Project]

Dr. M.R. Huth ***Term 1***

Learning Outcomes
- Students will understand the core values and practices of key agile and iterative development methods, be familiar with their history, and appreciate the advantages and trade-offs of these methods.
- Students will be competent in judging which agile development methods can be mixed, and to what extent, on a successfully managed project, with a particular emphasis on their own Group Project.
- Students will know what support is currently available for software quality assurance in implementation work. They will gain experience in writing and using unit testing for software validation.
They will be able to formulate and, with clients, conduct acceptance tests.
- Students will learn how to write project progress reports addressed to 'managers'. Students will learn how to effectively present a finished software product to a technical but general audience. They will learn how to write a final project report that competently, and in detail, describes their completed system and its evaluation.

Syllabus
This course features state-of-the-art methods in software engineering practices from a managerial, technical, and process perspective. This course is firmly integrated with a Group Project.
Agile Development: appreciation of agile methods, understanding of common threats to success of IT projects, study of agile methods such as Scrum and Extreme Programming, discussion of how one may blend which agile methods into suitable mix for Group Project.
Software Testing: verification versus validation, acceptance testing, unit testing, black-box testing, grey-box testing, integration testing, regression testing, interface testing, automated testing, test coverage.

Model-based Testing: static versus dynamic analysis, model-based development, benefits of models in software lifecycle, behavioral models, criteria for model choice, heuristics for model construction, model-based test-case generation.
Technical writing: how to write and structure reports globally and at the level of chapters, sections, subsections, and paragaphs; effective use of tables and figures; discussion and evaluation of findings; proper citations and use of references.

Technical speaking: how to give a technical talk, respecting you and your audience, balancing technical details with the big picture, slide etiquette, choice of fonts and animations, example layouts of talks.

C303 SOFTWARE ENGINEERING – SYSTEMS VERIFICATION

Term 2

Aims

The course aims at introduce students to the area of formal methods for system specification and verification. Particular prominence will be given to logic-based formalisms and techniques for system specification and verification and in particular to symbolic model checking.

Syllabus

Part 1: Introduction. Why verification and validation. Verification in the SE life-cycle. Correctness, Faults and Errors. Lessons learnt: Arianne V, London Ambulance Service, etc. Formal verification: pros and cons. Model checking pros and cons.

Part 2: Logics and specifications: Modal logic: syntax and semantics. Temporal logic. LTL. CTL. Expressive power. Epistemic logic, interpreted systems. Specifications in temporal and epistemic logic. Examples: mutual exclusion, bit transmission problem, lift systems.

Part 2: Logics and verification. Theorem proving and model checking. Explicit and symbolic model checking. Boolean formulas. Ordered-binary-decision-diagrams. Fixed point operators. Protocol design and verification. Verification for logics for knowledge.

Part 3. Model checking packages: The nuSMV verification toolkit. MCMAS. SMV and ISPL syntax. Lab classes. Debugging. Countermodels.

Part 4. Verification Scenarios. Verification of PCI bus, verification of security protocols (anonymity and authentication).

Part 5. Advanced techniques. Limitations of current OBDD-based techniques. Experimental results. Bounded model checking. Unbounded model checking. The Verics model checking toolkit.

Learning outcomes:

Upon successful completion of this module the student should be able to:
1) Explain the basics of logic as a formal specification language.
2 Use the main verification algorithms for verification of computing systems such as the

ones based on ordered-binary-decision diagrams.
4) Understand the architecture and be able to use
a main tool (such as nuSMV) to model check specifications.
5) Appreciate the limitations of the current tools and techniques and develop a basic understanding of current research directions.

C317 GRAPHICS

Professor D. Gillies/Professor D. Ruckert ***Term 2***

Syllabus
This course covers the fundamental principles of computer graphics and their use in prominent applications. The lectures include:
Device independent graphics: Raster and vector devices, world coordinates, the normalisation transformation, output primitives, input primitives. Polygon rendering: 3D data base representation, projection onto a viewing surface, transformation of graphical scenes, homogenous co-ordinates, affine transformations for animation. 3D geometry:
clipping and containment in 3D convex objects, splitting concave objects. Texture mapping and anti-aliasing. Shading planar polygons:
Gouraud shading, Phong shading. Representing Colours: tri-stimulus model, RGB model, CIE definition space, perceptual colour spaces. Ray Tracing: Ray/object intersection calculations, secondary rays, shadows, reflection and refraction, object space coherence, ray space coherence.
Radiosity: Modeling ambient light, form factors, specular effects, shooting patches, computational efficiency. Geometric Warping and Morphing. Special Visual Effects: particle systems for fire smoke and water, inverse kinematics in animation, non-photorealistic rendering.

C318 CUSTOM COMPUTING

Professor W. Luk/Dr. T. Todman ***Term 2***

Content
Overview: motivations; features and examples of custom computers; summary of development methods and tools.
Design: parametrised description of leaf components and composite structures; resource and performance characterisation; high-level design tools.
Optimisation: techniques for improving design efficiency such as pipelining, serialisation, transposition and their combinations.
Realisation: bit-level designs, data refinement, FPGA-based implementations.
System-on-chip: architectures, technology trade-offs, design and optimisation methods.
Examples will be selected from a number of application areas, including digital signal processing, computer arithmetic and non-numerical operations.

C320 COMPLEX SYSTEMS

Professor A. Edalat ***Term 1***

Content
In the course of the past decade complexity has become a major scientific field in its own right, which is sometimes considered the ultimate in interdisciplinary fields (EPSRC has named complexity among the five priority research areas along with energy, information communication technologies, nano-technology and health care). It relies on concepts in mathematics, physics and computer science on the one hand, and the rapid growth in computational power and information processing capabilities on the other.
This course aims to introduce students to this exciting new field, underlining its theoretical underpinnings, while engaging them in hands-on laboratory exercises in order to develop their practical skills and emphasise the applicability of this subject in the modern world.

Learning Outcomes

By the end of the course the students will be able to:

- Understand the theoretical foundations of complexity: mathematical, computational and information-theoretic;
- Reason about complexity with fluency and using scientific terminology; appreciate it as a post-Newtonian paradigm for approaching a large body of phenomena;
- Apply the terminology and methodology of nonlinear dynamics and chaos to study the time-evolution of complex systems;
- Describe some of the well-known models of complex systems, such as artificial life and Self-Organised Criticality (SOC);
- Construct new variations on models for complex systems using the modelling techniques taught in this course;
- Implement these models elegantly and efficiently using Java or MATLAB,;
- Appreciate the interdisciplinary nature of complexity and the pivotal role that computer scientists play in its study.

Syllabus

- Complexity theory: definition of complex systems, models, and dynamical systems; examples of complex systems (e.g. neural networks; traffic flow; artificial life; financial markets and human societies).
- Fundamental theoretical concepts: information, computational complexity, Kolmogorov complexity.
- Discrete dynamical systems: iterative maps; orbits; fixed and periodic points; graphical analysis; phase portraits; bifurcations; chaos. The quadratic family as an illustration of the phenomenology of dynamical systems.
- Cellular automata: cellular automaton rules; number-conserving cellular automata; generalised cellular automata. Cellular automata as models of complex systems: artificial life; stock market dynamics (e.g. Sornette's model of market fluctuations); sandpile model; percolation (e.g. forest fires).
- Complex networks: the small-world phenomenon; graphs; random networks; small-world networks; scale-free networks. WWW as a complex network.
- Hopfield model: associative memories; discrete Hopfield model; applications.
- Game theory: theory of choice; representations of games; repeated games. Economic applications of game theory. Competition and cooperation: stag hunt and prisoner's dilemma; repeated prisoner's dilemma.

C330 NETWORK SECURITY

[Recommended: Distributed Systems (C335)]

Dr. M. Huth/Dr. E. Lupu ***Term 1***

Aims

To survey the principles and practice of network security. The emphasis of the course is on the underlying principles and techniques of network security with examples of how they are applied in practice.

Learning Outcomes

At the end of the course, a student will have an understanding of: the themes and challenges of network security, the role of cryptography, the techniques for access control and intrusion detection, the current state of the art. The student will have developed a critical approach to the analysis of network security, and will be able to bring this approach to bear on future decisions regarding network security. Practical skills will include the implementation of a security protocol.

Outline Syllabus

Introduction: assets, threats, countermeasures; network security models, security functions: confidentiality, authentication, integrity, nonrepudiation, access control, availability, passive and active attacks, end-to-end vs link-to-link encryption.
Classical Cryptography: key ideas, steganography, codes, one-time pad, substitution and transposition ciphers, cryptanalysis, cryptographic strength.
Symmetric-Key cryptography: Feistel cipher; DES: basics, rounds, e-box, s-box, p-box, key

box; Modes of Operation: ECB, CBC, CFB, OFB; Double DES, Triple DES, IDEA, RC5, AES, problems with symmetric key cryptography.
Public-Key cryptography: requirements, confidentiality, authentication, modular arithmetic, Diffie-Hellman key exchange, RSA, attacks against RSA, hybrid cryptosystems, Elliptical Curve, Quantum Cryptography.
Digital Signatures: characteristics, MAC's, one-way hash functions, signing and verification, birthday attack, public-key certificates, disavowed signatures, arbitrated digital signatures, chaffing & winnowing.
Mutual Authentication: basics, replay attacks, man-in-the-middle, interlock protocol, Andrew Secure RPC, Needham Schroeder, Wide-Mouth Frog, Neuman-Stubblebine, Woo-Lam.
BAN-Logic
Key Management: distribution, KDC, announcements and directories, public key certificates, X509 certification authorities, PGP web of trust, control vectors, key generation and destruction, key backup.
Intruders and Programmed Threats: host access, password systems and attacks, one-time passwords, token cards, biometrics, trapdoors, programmed threats: trapdoors, logic bombs, trojan horses, viruses, worms, countermeasures, intrusion-direction.
Firewalls: internet security policies, firewall design goals, firewall controls, TCP/IP, packet filtering routers, application-level gateways, circuit-level gateways, firewall architectures, VPNs.
Web Security: WWW, web servers, CGI, active content, Java applets, Java security model: sandbox, class loaders, bytecode verification, security manager, Java attacks, bypassing Java, mobile code cryptography.

C332 ADVANCED COMPUTER ARCHITECTURE

Professor P. Kelly ***Term 2***

This is a third-level course that aims to develop a thorough understanding of high-performance and energy-efficient computer architecture, as a basis for informed software performance engineering and as a foundation for advanced work in computer architecture, compiler design, operating systems and parallel processing.

Topics include:

- Pipelined CPU architecture. Instruction set design and pipeline structure. Dynamic scheduling using scoreboarding and Tomasulo's algorithm; register renaming. Software instruction scheduling and software pipelining. Superscalar and long-instruction-word architectures (VLIW, EPIC and Itanium). Branch prediction and speculative execution.
- Simultaneous multithreading ('hyperthreading'), and vector instruction sets (such as SSE and AVX).
- Caches: associativity, allocation and replacement policies, sub-block placement. Multilevel caches, multilevel inclusion. Cache performance issues. Uniprocessor cache coherency issues: self-modifying code, peripherals, address translation.
- Dependence in loop-based programs; dependence analysis, and iteration-space transformations - to enable automatic parallelisation, vectorisation (eg for SSE), and for memory hierarchy optimisations such as tiling.
- Implementations of shared memory: the cache coherency problem. Update vs invalidation. The bus-based 'snooping' protocol design space.
- Scalable shared memory using directory-based cache coherency. How shared memory supports programmability; OpenMP and MPI.
- Graphics processors and 'manycore' architectures: SIMT ('single instruction multiple thread), and the CUDA and OpenCL programming models. Decoupling, latency tolerance, throughput-intensive memory system architecture.

Further details are available from the course web site: http://tinyurl.com/kelly-c332

C333 ROBOTICS

Dr. A. Davison ***Term 2***

Learning Outcomes
At the end of the course students will be familiar with the key concepts related to the building and programming of small robots. They will understand the different kinds of locomotion and sensor systems that can be used and how they can be integrated electronically, and the principles behind the programming of simple reactive behaviours. They will understand different ways in which reactive behaviours can be programmed and combined for controlling a robot in an unknown but fixed type of environment. They will have been introduced to advanced techniques in probabilistic filtering to permit simultaneous localisation and mapping, and aspects of motion planning.

Syllabus
This course will focus on the field of mobile robotics, at the exciting time when cutting-edge robots are beginning to leave the research laboratory to tackle real-world tasks --- in space, in the desert or in your living room. A robot is a physical device where sensing and actuation must be linked by efficient, robust and flexible on-board computing. We will emphasize both theoretical and practical aspects of the field, with hands-on tutorial sessions giving the opportunity to build and program robots.
We will discuss the various approaches to locomotion commonly used in mobile robots, and look at the problems and advantages of various wheeled configurations as well as alternatives such as legs or flight. We will also look in depth at the variety of sensors available, from simple touch sensors to advanced computer vision. It will be shown that the combination of simple sensing and motion devices using reactive loops permits behaviours such as obstacle avoidance and line-following.
Having introduced the notion of a behaviour, we will look in more depth at how behaviours can be designed and combined. We will examine the concept of Braitenberg vehicles, which clarifies the direct coupling of sensor and action in interesting behaviours. We will then examine a rule based programming notion called Teleo-Reactive programs for realising goal direction behaviour with sequence of sensor/action response rules. This leads naturally into the examination of two different architectural principles that have been proposed for designing reactive mobile robots - the Brooks subsumption architecture and the behaviour based architecture based on vector fields. We will then discuss the use of finite state machines for sequencing lower level behaviours.
We will look at how probabilistic filtering techniques are the key to processing uncertain data from complex sensors such as cameras, and discuss how modern robots localise by building a map of landmarks using Simultaneous Localisation and Mapping (SLAM). Finally the course will look at a case study of an unmanned autonomous vehicle competition winner and cover the path planning - the construction of a sequence paths for the robot to follow that will enable it to navigate from a given start location to a given destination location in a known environment of obstacles.
Learning Outcomes
At the end of the course students will be familiar with the key concepts related to the building and programming of small robots. They will understand the different kinds of locomotion and sensor systems that can be used and how they can be integrated electronically, and the principles behind the programming of simple reactive behaviours. They will understand different ways in which reactive behaviours can be programmed and combined for controlling a robot in an unknown but fixed type of environment. They will have been introduced to advanced techniques in probabilistic filtering to permit simultaneous localisation and mapping, and aspects of motion planning.

C335 DISTRIBUTED SYSTEMS
[Recommended: Operating Systems

This course is a prerequisite for Distributed Algorithms (C437)]

Professor M. Sloman ***Term 1***

Aims
The objective is to give students a clear overview of the problems and issues that must be dealt with in constructing secure and flexible distributed applications. The emphasis is on

the conceptual basis for distributed and networked systems rather than a detailed study of particular systems and standards. Concepts will be illustrated with examples from practical systems.

Learning Outcomes
A sound understanding of the principles and concepts involved in designing distributed systems and Internet applications
Ability to implement a distributed application using Java RMI and datagrams
Understand the design issues relating to publish-subscribe, peer-to-peer networks

Syllabus
Overview of Distributed System Architecture: motivation, system structures, architecture, ODP Reference model and distribution transparencies, design issues.
Interaction Primitives: message passing, remote procedure call and Interface definition language, objects and remote object invocation (RMI)
Interaction implementation: message passing, RPC, concurrency and threads, heterogeneity of systems and languages.
Time Synchronisation
Distributed systems management: SNMP management model & MIBs, management and security policy.
Ubiquitous computing: grand challenges, engineering issues, policy-based management and self-managed cells
Publish-Subscribe systems: subscription classification, composite events, case studies - Siena, SMC, Hermes, event routing
Peer-to-peer Networking: structure, case studies - Gnutella, Freenet, BitTorrent, JXTA.
Sensor Networks: history and deployment, case studies, future directions.

C337 SIMULATION AND MODELLING
[Exclusive with M3S9 Stochastic Simulation]

Dr. J. Bradley/Dr. A.J. Field ***Term 1***

Objectives
This course provides an introduction to system modelling using both computer simulation and mathematical techniques. A range of case studies are examined, both in the lectures and tutorial exercises. The application areas considered are wide-ranging, although the emphasis is on the analysis of computer and communication systems using a variety of modelling paradigms such as simulation, queueing theory, stochastic process algebras and stochastic Petri nets. The course is self-contained, both in terms of notes and supporting software.

Learning Outcomes
By the end of the course students will be able to apply the fundamental laws of performance analysis to establish the relationships between workload parameters and system performance for a given system. They will also be able to develop performance models for simple real-world systems and will be able to solve those models to obtain meaningful performance measures. They will thus be able to analyse system responsiveness, scalability etc. as a function of workload.

Content
Introduction and motivation.
Modelling principles: Fundamental laws, Monte Carlo simulation, stochastic state transition systems.
Discrete-event simulation (DES): Principles of DES, formalisation as a Generalized Semi-Markov Processes (GSMPs), random number generation, distribution sampling, analysis of simulation output.
Markov Processes (MPs): Numerical solution of MPs, analytical solution of MPs.
High-level formalisms, tools and techniques: Stochastic Petri Nets (SPNs) and Stochastic Process Algebras (SPAs), generation of the underlying MP, Queueing Networks.
Stochastic simulation in biochemical modelling.

C341 INTRODUCTION TO BIOINFORMATICS

Professor Y. Guo/Dr. N. Przulj ***Term 2***

Course Overview and Goals

Vast amounts of biological network data have recently been generated due to advances in experimental biology. These data sets are increasingly beings studied to obtain systems-level understanding of biological structures and processes. Various mathematical and computational tools are being used and developed to analyze and model these data aiming to achieve a better description and understanding of biological processes, disease, and contribute to the time and cost effectiveness of biological experimentation.

This course will give an overview of the existing types of biological network data, point to sources of errors and biases in the data, and introduce the current methods, models and literature on graph theoretic modeling and discrete algorithmic analyses applied to these data. The course will also present an overview of the works of several major network biology labs around the world (e.g., U. Alon, M. Vidal, M. Tyers, M. Stumpf, J. Doyle, A.-L. Barabasi etc.).

Topics Outline

The course will cover the following topics:
a) Types of biological networks: metabolic, signaling, protein-protein interaction, etc.
b) Major databases storing biological network data (e.g. MINT, DIP, HPRD, GRID, MIPS, KEGG).
c) Sources of noise and biases in various types of the biological network data (e.g., biotechnological biases and limitations, effects of sampling).
d) Computational challenges in network analysis: introduction to basic graph theoretic and computational complexity concepts such as subgraph isomorphism and NP-completeness.
e) Properties of large networks: global (e.g., degree distribution, clustering coefficient, average diameter) and local (e.g., network motifs and graphlets).
f) Network models: various types of random graphs (e.g., Erdos-Renyi, small-world, scale-free, hierarchical, geometric) and network growth models (e.g., preferential attachment).
g) Network motifs: techniques for their detection (exhaustive and heuristic network search algorithms) and biological function (e.g., feed-forward loops in transcriptional regulation networks).
h) Interplay of network topology and function (e.g., “lethality” and “centrality,” “synthetic lethality” and network “redundancy,” graph theoretic pathway models).
i) From models to heuristic algorithms (e.g., exploiting network model properties for “optimal” walks through a network, or detection of small network substructures).
j) Graph alignment heuristics (e.g., PathBLAST, IsoRank, GRAAL).
k) Network evolution (e.g. gene duplication and divergence in biological network growth models).
l) Clustering problems in biological networks (e.g., detection of protein complexes).
m) Software tools and libraries for network analysis (e.g., LEDA, Cytoscape, Pajek).

C343 OPERATIONS RESEARCH
[Exclusive with M3N3 Optimisation]

Dr. D. Kuhn/Professor B. Rustem ***Term 1***

Aims

The purpose of this course is to study the basic tools for quantitative methods for decision making. The emphasis is on solution methods and strategies.

Learning outcomes

The student will understand the underlying algorithms and be able to interpret the results. He/She will be able to formulate problems as abstract models which can be solved by generic algorithms.

Syllabus

To introduce optimal decision making processes in design and management. To give the

necessary mathematical background and its application to solving a selection of constrained optimisation problems with special reference to computation.
Preview: optimal policy in design and management: mathematical models.
Linear programming: The Simplex method, two-phase Simplex method, duality, shadow prices.
Linear integer programming: Gomory's cutting plane methods for pure and mixed linear integer programming. Search methods; branch and bound algorithms.
Game theory: two person non-co-operative games. Saddle points. Matrix games.

C382 TYPE SYSTEMS FOR PROGRAMMING LANGUAGES

Dr. S. van Bakel ***Term 1***

Content
The course sets of with the presentation of the Lambda Calculus, on which most functional languages are based. We will introduce the Curry Type Assignment System for this calculus, for which we show that types are respected by reduction, and there exists a notion of principality on types, i.e. all types for a program can be constructed from the 'principal' type.
We will then focus on how to extend the language and the system of types in order to deal with Polymorphism. To this end, Combinator Systems are introduced as a slight extension of LC. In order to study how to deal with Recursion, the language of CS is extended to allow for recursive definitions, and we will discuss ways of typing this extension.
We will then look at Milner's basic ML system, and show that, in this calculus, a solution is present for all the issues discussed separately before. A disadvantage of ML is that it is, syntactically, very distant from actual programming languages, and it is not easy to see that all properties shown for ML translate to the 'real' languages. In particular, definitions of functions by 'cases' (Pattern Matching) are not present in ML. In order to come to a formal notion of types that relates more closely to actual programming languages, we introduce Term Rewriting Systems, which are much closer to actual programs, and show that we can obtain much of the desired properties. To conclude, we will have a brief visit to Intersection Type Assignment, which allows to prove strong properties. The aim of this course is to lay out in detail the design of type assignment systems for programming languages and focus on the importance of a sound theoretical framework, in order to be able to reason about the properties of a typed program. Students will study and compare a variety of systems and languages.

Lambda Calculus: terms, abstraction, application, reduction, normal form, normalisation, head normal form, head normalisation. The Curry type assignment System.

Types, type assignment rules, type substitution, unification, subject reduction. The Principle type for Curry's System.

Recursion and Polymorphism: the need for a recursor, how to type recursion, Milner's let, the basic ML system, the principal type property for the ML system.

Church's typed lambda calculus: type checking versus type inference.

The Polymorphic Lambda calculus. Strong normalisation for the polymorphic Lambda Calculus.

Intersection type assignment: types, type assignment rules, subject expansion, undecidability, filter semantics.

Patterns: term rewriting systems, weak reduction, normal forms, strong normalisation. Dealing with patterns, recursion, subject expansion and reduction.

A decidable restriction of the intersection system.

C395 MACHINE LEARNING
[Recommended: Introduction to Artificial Intelligence (C231)]

Professor S. Muggleton/Mr. L. Dickens/Professor M. Pantic Term 1

Machine Learning (course 395) is envisioned to be an introductory course for several groups of students including MSc Advanced Computing students, Fourth Year Information Systems Engineering students, and Third/Fourth Year Mathematics and Computer Science students.

Course Aims
students should be familiar with some of the foundations of the Machine Learning (ML), students should have an understanding of the basic ML concepts and techniques:

- Concept Learning (to be taught by Maja Pantic)
- Decision Trees (to be taught by Maja Pantic)
- Artificial Neural Networks (to be taught by Maja Pantic)
- Instance Based Learning (to be taught by Maja Pantic)
- A Genetic Algorithms (to be taught by Maja Pantic)
- Evaluating Hypotheses (to be taught by Maja Pantic)
- Inductive Logic Programming (to be taught by Mr. L. Dickens)

students should gain programming skills using Matlab with an emphasis on ML and they should learn how to design, implement and test ML systems
students should enhance their skills in project planning, working with dead-lines, and reflecting on their own involvement in the teamwork

Course Material
Book: Machine Learning by Tom Mitchell, McGraw-Hill Press, 1997 (chapters: 1-5, 8, 9)
Computer Based Coursework (CBC) Manual
Syllabus on Case-based Reasoning
Lecture notes on Inductive Logic Programming (pdf file)

Course Schedule
The curriculum schedules 18 class meetings of one hour each. The CBC for this course will mainly be devoted to coursework (+/- 80 hours per group of 4-5 students).

Attendance to the CBC is mandatory and accounts for 10% of the final grade for the Machine Learning Course. Students are allowed to miss at most 2 hours. Failure to attend due to medical reasons requires a letter from your GP.

Computer Based Coursework (CBC)

CBC CONTENT:
The project is designed to build on lectures by teaching students how to apply ML techniques about which they have been lectured to real-world problems.

The project will consist of five assignments. The first four assignments will focus on emotion recognition from data on displayed facial expression using decision trees, neural networks, case-based reasoning, and genetic algorithms. The last assignment will focus on evaluating (by means of paired t tests and ANOVA test) which of these ML techniques is more suitable for the problem in question in the case of clean data and in the case of noisy data.

CBC ASSESSMENT:
Assessment of the project work will be conducted based upon the following:
· the quality of the delivered code as measured by the clarity, effectiveness and efficacy of the delivered code when tested on real (previously unseen) data,
· the quality of the delivered project report (+/- 2 pages) as measured by the correctness, depth and breadth of the provided discussion on the evaluation of the performance of the developed ML systems for emotion recognition,
· personal involvement and contribution to the group’s results (to be judged based upon a final interview with each of the groups).

C526 DATABASES

Dr. P. McBrien ***Term 2***

Objectives
To provide an introduction to database systems, in general. To investigate the theory and practice of the relational model in detail, including design, database languages, transaction management and recovery and concurrency. To provide an introduction to entity-relationship modelling and translation to the relational model.

Content
Introduction to databases, including data modelling, database management, data dictionary, query formulation and evaluation. Relational databases: design, functional dependencies, normalisation up to and including the fourth normal form, losslessness, dependency preservation.
Relational database languages, including relational algebra and relational tuple calculus. Views and database integrity.
Transaction management and recovery: transaction atomicity, database log, commit and rollback, recovery from system and media failure, deferred and immediate database modifications, checkpoints.
Concurrency: including conflict serialisability, conflict equivalence, precedence graphs, serialisability, locking, two phase locking protocol, deadlock.
Entity-relationship modelling and translation to the relational model.
The course will be supported by laboratory sessions using the relational database system, INGRES, and the language SQL.

C527 COMPUTER NETWORKS AND DISTRIBUTED SYSTEMS

Dr. P. Pietzuch/Professor M. Sloman ***Term 2***

Aims
The objective is to give students a clear overview of the problems and issues that must be dealt with in constructing robust and flexible distributed applications as well as the underlying network protocols needed to support them. The emphasis is on the conceptual basis for distributed and networked systems rather than a detailed study of particular systems and standards. Concepts will be illustrated with examples from practical systems.

Learning Outcomes
- A sound understanding of the principles and concepts involved in designing distributed systems and Internet applications
- Ability to implement a distributed application using Java RMI
- Ability to understand and evaluate network and security solutions

Content
Network Overview: interfaces, protocols and services, connection-oriented and connectionless services, OSI & TCP/IP Reference Models.
Local Area Networks: Topologies - star, bus, ring, media access control - deterministic and probabilistic, IEEE 802.x. wireless networking.
Data Link Protocols: framing and data transparency, error detection & correction, flow control.
Interconnecting Networks: transparent and source routing bridges, switches. Routers - adaptive and non-adaptive routing protocols.
Internet Protocols: IP Addressing, ARP & RARP, IP & ICMP, UDP & TCP.
Overview of distributed system architecture: motivation, system structures, ODP Reference model and distribution transparencies, design issues.
Interaction primitives: message passing, remote procedure call, remote object invocation.
Interaction implementation: message passing, RPC, concurrency and threads.
Security: threat analysis, security policies - military (Bell Lapadula) versus commercial models; access control concepts - identification, authentication, authorisation and delegation; authorisation policy: access matrix, access rules and domains; access control lists, capabilities, secret and public key encryption, digital signatures, authentication, Kerberos.

C528 CONCURRENT PROGRAMMING

Dr. N. Yoshida ***Term 2***

Syllabus not available at time of writing. Email yoshida@doc.ic.ac.uk for further information on this module.