

COMPUTER SCIENCE COURSE HANDBOOK

2011/2012

Edited by C. J. Whyley
August 2011

Department of Computer Science
Swansea University
Singleton Park
Swansea, SA2 8PP
Wales

Telephone: (+44) 01792 295651
Facsimile: (+44) 01792 295708
Email: csdept@swansea.ac.uk
WWW:<http://www.swan.ac.uk/compsci/>

difficilia quae pulchra

things that are excellent are difficult

This Handbook is intended for staff and students of Computer Science at Swansea University. The information contained in this Handbook is, to the best of our knowledge, correct at the time of printing. The Department retains the right to change details it contains, but every effort will be made to notify students and other interested parties of such changes. Where discrepancies arise University Regulations and Procedures will apply.

This is the Twenty First Edition of the New Series of Department Course Handbooks. Any corrections or suggestions for improvement of this Handbook should be communicated to the Editor, as should any broken hyperlinks.

CONTENTS

Introduction to the College of Science
Staff of the College
College Committees
Introduction to the Department of Computer Science
Computer Science Course Structure
Staff of the Department
ITWales Staff
Swansea University History of Computing Collection
Contact Points: Department
Contact Points: University
University Support for Students
Departmental Committees
Staff Directory
ITWales Directory
General Information

Degree Programmes

Introduction
Computer Science Modules and Educational Aims
Computer Science Modules for programmes
BSc Computer Science
MEng Computing
BSc Computing and Communications
BSc Mathematics for Computer Science
BSc Computing with Finance
BSc Computer Science with a Modern Language
BSc Computer Science and Pure Mathematics
BSc Computer Science and Geoinformatics
BSc Computer Science and Psychology
BSc Computer Science and Physics
BA Modern Language(s) with Computer Studies
Level Three Optional Modules
MSc Computer Science
MSc Computing and Software Technology
MSc Computing and Future Interaction Technology
MRes Schemes
MSc Advanced Computer Science wsi Software Technology
MSc Advanced Computer Science wsi Human Computer Interaction
MSc Advanced Computer Science wsi Visual Computing
MSc Advanced Computer Science wsi Safe and Secure Systems
MSc Advanced Computer Science wsi Web Science
MSc Advanced Computer Science (Open)
UCAS and UWS Route Codes

Level Zero Modules

CS-061 Introduction to Computing I

Level One Modules

CS-108 Computers and Computing
CS-110 Programming I
CS-113 From Languages to Hardware
CS-115 Programming II
CS-130 Professional Issues I: Computers and Society
CS-135 Professional Issues II: Software Development
CS-150 Concepts of Computer Science I
CS-155 Concepts of Computer Science II
CS-170 Modelling Computer Systems I
CS-175 Modelling Computer Systems II
Level One Awayday

Level Two Modules

CS-205 Declarative Programming
CS-210 Concurrency
CS-215 Logic Programming and Artificial Intelligence
CS-217 Computer Graphics I: Image Processing and Synthesis
CS-219 Database Systems
CS-230 Software Engineering I
CS-235 Software Engineering II
CS-236 Language and Computation
CS-242 Algorithms
CS-250 Database Systems
CS-255 Computer Graphics
CS-270 Algorithms
CS-275 Automata and Formal Language Theory
Level Two Awayday

Level Three Modules

CS-307 Computer Graphics II: Modelling and Rendering
CS-311 Concepts of Programming Languages
CS-313 High Integrity Systems
CS-318 Cryptography and IT Security
CS-337 Data Visualisation
CS-338 Internet Computing
CS-344 Project Implementation and Dissertation
CS-345 Artificial Intelligence Applications
CS-348 Building Reliable Web Applications
CS-349 Mobile Interaction Design
CS-354 Project Specification and Development
CS-358 High-Performance Computing in C/C++
CS-364 Software Testing
CS-368 Embedded Systems
CS-371 Design Patterns and Generic Programming
CS-375 Logic for Computer Science
CS-377 Computer Vision and Pattern Recognition
MS-306 New Screen Technologies
MS-353 Digital Philosophy: The Roots of the Virtual
HIP300 The Classical Tradition in the Sciences
HIP301 From Natural Philosophy to Science

Undergraduate Computer Science Colloquium
Project Demonstration Fair

Level M Modules

CS-M00 Research Methodology
CS-M04 Group Project
CS-M05 Advanced Topics in Logic and Computation
CS-M07 Data Visualisation
CS-M08 Future Interaction Technologies: MSc Project
CS-M09 Future Interaction Technologies: MRes Project
CS-M10 Computer Science Project Development
CS-M12 Software Concepts and Efficiency
CS-M13 Critical Systems
CS-M14 Industrial Project
CS-M15 Directed Studies in Logic and Computation
CS-M18 IT Security: Theory and Practice
CS-M19 Interactive Systems Design
CS-M20 MSc Project
CS-M24 Software Team Project
CS-M29 Mobile Interaction Design
CS-M35 Logic and Computation MRes Project
CS-M39 Interaction Technologies: Seminars and Reading
CS-M40 Software Technology Project Development
CS-M41 Programming in Java
CS-M49 Interaction Technologies: Lab and Field Work
CS-M50 Safe and Secure Systems Project Development
CS-M53 Computer Science Concepts
CS-M58 Distributed O-O Programming
CS-M59 Relational and Object-Oriented Database Systems
CS-M61 Concepts of Programming Languages
CS-M65 Artificial Intelligence Applications
CS-M67 Graphics Processor Programming
CS-M68 Writing Web and Web Service Applications
CS-M69 Interaction Technologies: Information Retrieval
CS-M70 Visual Computing Project Development
CS-M71 Design Patterns and Generic Programming
CS-M75 Logic for Computer Science
CS-M77 Computer Vision and Pattern Recognition
CS-M78 High Performance Computing in C/C++
CS-M79 Interaction Technologies: Hardware and Devices
CS-M80 Web Science Project Development
CS-M84 Software Testing
CS-M85 Modelling and Verification Techniques
CS-M87 Computer Graphics MRes Visual Computing Project
CS-M88 Embedded Systems
CS-M89 Advanced Database Systems
CS-M90 Human Computer Interaction Project Development
CS-M94 Software Engineering Principles

MSc Modules from Other Colleges/Departments

MSDM01 Thinking about Digital Media

ASCM17 Understanding Crime

GEGM13 Urban Networks and the Knowledge Economy

LALM121 Contract and E-Commerce Law

PO-M52 Rethinking Global Politics

EBCM10 E-Business

ITWales

Introduction

Student and Graduate Placements

itwales.com

Software Alliance Wales

SAW Deliverables

Technocamps

Other Information

General Reading List

Tutorials

Assessment and Progression

Employability

British Computer Society Accreditation

General Information

Student Obligations - University

Student Obligations - Department

2011/2012 Session Calendar

INTRODUCTION TO THE COLLEGE OF SCIENCE

MISSION STATEMENT OF THE COLLEGE OF SCIENCE:

To create an environment where academic research excellence can flourish, hand in hand with high quality undergraduate and postgraduate courses, to enrich our knowledge economy and external engagement strategies.

THE COLLEGE'S AIMS AND OBJECTIVES:

To provide an academic environment in which all staff, students and visitors are able to excel.

WELCOME FROM THE HEAD OF THE COLLEGE OF SCIENCE:

It gives me great pleasure to extend a warm greeting to new and returning students of the College of Science.

The newly formed College brings together the Departments of Biosciences, Computer Science, Geography, Mathematics and Physics in an environment that enables us to collaborate in enhancing our teaching and research environment, and to undertake initiatives that provide new and challenging opportunities for our students.

**Prof Steve Wilks
Head of the College of Science
Swansea University**

1st August 2011

STAFF OF THE COLLEGE

COLLEGE STAFF

Head of College	Professor Steve Wilks, BSc, PhD, FinstP
Deputy Head of College	Professor Marcus Doel, BSc, PhD
College Manager	Mr. Steven Walmsley
College Secretary	Mrs Leighan Evans

COLLEGE COORDINATORS

College Chair of Learning and Teaching	Dr. Sharp
College Unfair Practice Representatives	
Biosciences	Professor Andrew Rowley
Computer Science	Mr. Chris Whyley
Geography	Professor Alayne Street-Perrott
Mathematics	<i>tbc</i>
Physics	Dr. Warren Perkins
Recruitment and Admissions Officer	Mrs. Sandra Kramcha
Departmental Programme Directors	
Biosciences	Dr. Dan Forman
Computer Science	Dr. Neal Harman
Geography	Dr. Kevin Rees
Mathematics	Dr. Ian Davies
Physics	Professor Simon Hands
Safety Officer	College Technical Manager, <i>tbc</i>
Chair of Postgraduate Affairs	Dr. Adrian Luckman (Geography)
Welsh Language Officer	<i>tbc</i>
Disability Officer	Mrs. Linda Andrews
Examinations Officer	Mr. Steven Walmsley, College Manager
Emergency Management (Business Recovery)	Mr. Steven Walmsley
Heads of Department	
Biosciences	Professor Kevin Flynn
Computer Science	Professor Matt Jones
Geography	Professor Stefan Doerr
Mathematics	Professor Niels Jacob
Physics	Professor Graham Shore

COLLEGE COMMITTEES

LEARNING AND TEACHING COMMITTEE

Professor Marcus Doel	Deputy Head of College
Dr. John Sharp (Computer Science)	Head of Learning and Teaching
Programme Directors of the five constituent departments	
Dr. Dan Forman	Biosciences
Dr. Neal Harman	Computer Science
Dr. Kevin Rees	Geography
Dr. Ian Davies	Mathematics
Professor Simon Hands	Physics
Mrs. Linda Andrews	Administrative Team Leader

other members of the teaching administrative staff

RESEARCH COMMITTEE

Professor Steven Wilks	Head of College
Professor Marcus Doel	Deputy Head of College
Professor Tavi Murray	Director of Research (Chair)
Heads of Department	
Dr. Adrian Luckman	
Professor David Dunbar	
Professor Harold Thimbleby	

other representatives

INTRODUCTION TO THE DEPARTMENT OF COMPUTER SCIENCE

This Handbook contains a great deal of information about the programmes and courses, teaching and assessment methods, etc., of the Department of Computer Science at Swansea University. It is intended as a convenient reference for the students and staff of the Department; it may also be consulted by many other people interested in our educational programmes.

Our education in Computer Science is characterised by nine **Educational Aims** (displayed below). The first three Educational Aims are focused on knowledge of the subject of Computer Science, the next three are concerned with related knowledge and experience, and the last three with personal competence.

EDUCATIONAL AIMS

The aims of our education in Computer Science are to provide our students with

1. practical experience and theoretical understanding of design methods for the specification, programming and analysis of a wide range of computing systems;
2. a fundamental understanding of the scope and limits of Computer Science and Artificial Intelligence, and of their applications;
3. knowledge of the history and present state of Computer Science, and an insight into future technologies and their role in applications and society;
4. the ability to plan and accomplish a substantial project;
5. relevant mathematical knowledge and experience in its applications;
6. experience in co-operative working through team projects, with their demands on the management of partners and time;
7. skills in written and oral communication;
8. skills in locating information, and the ability to read critically, to précis and to judge information; and
9. the ability and confidence to learn, unaided, complex new subjects.

A quick impression of our educational programmes is given by the picture of our current scheme for the BSc in Computer Science on Page 6. The Handbook provides the details of this and all our other schemes and courses.

The Department received the highest possible rating of *Excellent* in the most recent Teaching Quality Assessment, and our Computer Science programme is accredited by the British Computer Society.

The work of the Department is not just its teaching. We are dedicated to studying and advancing Computer Science and its applications. We play our part in the international community of computer scientists. We run a programme called *ITWales* for industry and the community which links the skills, experience and needs of industry to the knowledge, expertise and resources of members of the Computer Science Department, especially its students through an optional *Student Placement Scheme*.

The Department has set itself the following Aims:

MISSION

1. To accomplish outstanding research in Computer Science and its applications that is fundamental and useful.
2. To provide an excellent education in Computer Science for undergraduate and postgraduate students that is of the highest international standards.
3. To recognise and respond to the changing needs of society through specific high-quality research, teaching programmes and initiatives.

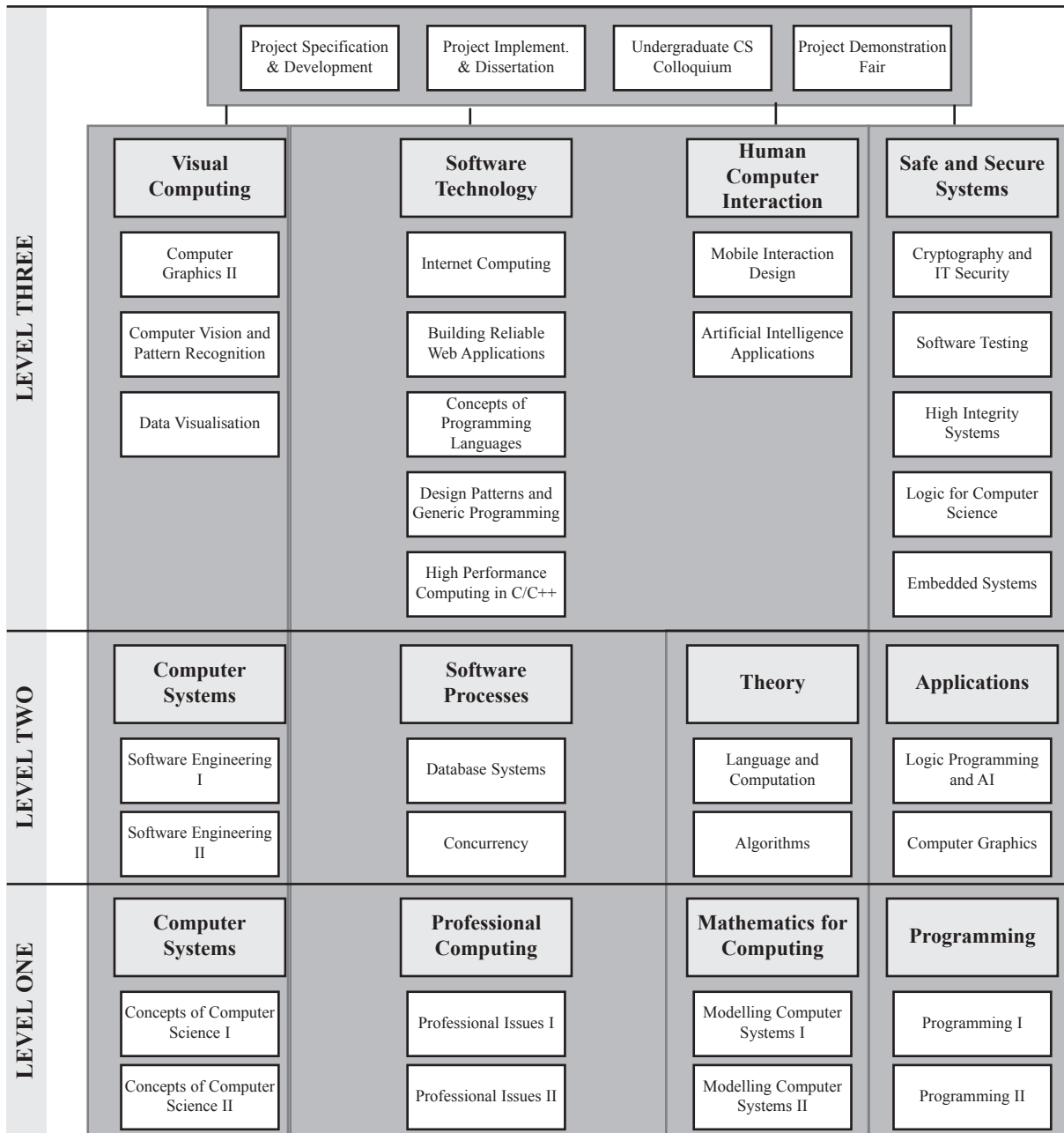
Thus, to appreciate fully our educational work, an understanding of our other aims and activities may be needed. Some further information can be obtained from brochures and reports, and from the extensive Internet pages that we publish. The best way to discover Computer Science at Swansea is to visit and engage in conversation with us!



Prof. Matt Jones
Head of Department of
Computer Science

1st August 2011

COMPUTER SCIENCE COURSE STRUCTURE



Note:

1. The above diagram illustrates the scheme for the *BSc in Computer Science* from 2012/2013 onwards. During the transitional change, modules for 2011/2012 may differ. See page 40 for details.
2. *The Undergraduate Computer Science Colloquium* is not an independently-assessed module, though it contains some assessed components.

STAFF OF THE DEPARTMENT

ACADEMIC STAFF

Head of Department

M. Jones, BSc, MPhil, PhD, MACM, FBCS CITP, CEng, FRSA

Theories, tools and methods for future interaction technology design. Particular interest in mobile and ubiquitous computing; digital libraries; and, software engineering that accommodates social and ethical concerns.

Deputy Head of Department

N. A. Harman, BSc, PhD

Algebraic specification of microprocessors and other digital hardware. Algebraic specification of languages and environments. Formal specification methodologies.

Deputy Pro-Vice Chancellor for High Performance Computing

J. V. Tucker, BA, MSc, PhD, CEng, FBCS

Mathematical theories of specification and computation. Topological data types. Synchronous concurrent algorithms and their applications. Logic, algebra and applications. Historical and social aspects of technology

Professors

F. G. Moller, BSc, MMath, PhD, CITP, CMath, CSci, FBCS, FIMA

Models of concurrent computation. Modal and temporal logic. Equivalence and model checking of infinite state systems.

P. D. Mosses, BA, MSc, DPhil

Concepts and semantics of programming languages. Compiler generation. Algebraic specification.

H.W. Thimbleby, BSc, MSc, PhD, Hon.FRSA

Human-computer interaction. Future interaction technologies.

Readers

A. Beckmann, Dipl-Math, Dr rer nat, PD

Logic, proof theory and applications. Bounded arithmetic. Propositional proof complexity.

U. Berger, Dipl-Math, Dr rer nat, PD

Logic, proof theory and applications. Domain theory. Theorem provers and program synthesis.

A. G. Setzer, Dipl-Math, Dr rer nat, docent

Logic, proof theory, type theory, esp. Martin-Löf type theory and programming with dependent types. Theory of object oriented programming. Interactive and machine-assisted theorem proving.

Senior Lecturers

P. W. Grant, BSc, DPhil, FBCS, CITP

Logic programming and expert systems. Multimedia communications. Parallel processing. Evolutionary computing.

M. W. Jones, BSc, PhD

Computer graphics. Volume visualisation and rendering. Image processing.

O. Kullmann, Dipl-Math, Dr phil nat

Complexity theory. Algorithms for hard problems. Combinatorial optimisation. Combinatorics. Propositional proof systems. Satisfiability problems and generalisations.

R. S. Laramée, BSc, MSc, PhD

Data visualisation, data analysis, computer graphics. Human-computer interaction.

M. Roggenbach, Dipl-Inform, Dr rer nat

Mathematical theories of specification. Algebraic specification. Process algebra. Tool support for specification languages.

Lecturers

O. Arandjelovic

Computer Vision and machine learning, and their interdisciplinary applications

J. E. Blanck, MSc, PhD

Theoretical computer science. Logic and computability theory. Domain theory. Continuous data types.

R. Borgo, BSc, MSc, PhD

Scientific visualization, information visualization, and visual analytics. Human factors in visualization. Multimedia processing and visualization. High performance computing. Functional programming

P. Eslambolchilar, BEng, MEng, PhD

Mobile Human-Computer Interaction, Gesture recognition, Multimodal interfaces, Dynamic and continuous interaction, Persuasive Technologies

B. Mora, PhD

Computer graphics and visualisation. Volume rendering. Computational geometry.

M. Seisenberger, Dipl-Math, Dr rer nat

(on leave 2011/2012)
Logic. Infinitary combinatorics. Proof theory. Interactive theorem proving. Formal methods. Specification and verification.

J. A. Sharp, BSc, PhD

Data flow computing. Program design environments. Parallel processing. Functional programming.

M. L. Wilson, MEng, PhD

Human computer interaction. Future interaction technologies. Information Science. Web Science

RCUK Fellow

X. Xie, BSc, MSc, PhD, MIEEE

Computer vision, image processing, video analysis, medical imaging technology.

Senior Tutor

R. D. Stein, BSc, PhD, MBCS, CEng, CSci, CITP

Database systems. Software development and management.

Tutor

C. J. Whyley, BSc, MPhil

Programming. Design Patterns. Compilers. Plagiarism detection and deterrence.

CAST/RIVIC KNOWLEDGE TRANSFER

A. Morris

ADMINISTRATORS

J. Edwards, ECDL

Teaching Administration

V. Hurst

Research PA

J. Lewis

Research Administration

RESEARCH STAFF

C. Monroy Aceves, BSc MA PhD

Medical device design.

R. Byrne, BSc

Human computer interaction.

D. Chen, BSc, PhD

Computer vision and medical image analysis.

D. H. S. Chung, BSc

Video processing and visualization.

B. C. Daubney, BSc, PhD

Image and video processing, computer vision.

M. W. Davies, BSc, MSc

HCI and rural computing

H. Fang, BSc, PhD

Image and video processing, computer vision.

A. M. Gimblett, BA, MPhil

Formal methods for interface specification and usability. Theory of programming and specification languages. Programming techniques. Functional programming.

Y. Guo, BEng, MSc

HCI and energy sustainability.

D. Hegde, BSc

Software engineering, Business computing.

M. Jiang, BSc, PhD

Computer vision, image and video processing

R. Kammaje, BSc, PhD

Computer graphics.

P. A. Legg, BSc, PhD	Computer vision, medical imaging, video processing and visualization. K. Li, BSc MSc PhD Medical device design.
D. Lipsa, BSc	Visualisation.
P. Oladimeji, BSc, MRes	Medical device design.
M. Parry, BSc	Video processing and visualization.
J. Pearson, BSc, MSc	HCI, digital document manipulation, digital libraries, rural computing.
S. N. W. Robinson, BSc, MSc	Multimodal negotiated interaction in mobile scenarios. Mobile devices and human computer interaction. Digital Divide.
B. Spencer, BSc, PhD	Computer graphics, visualisation.
S. Walton, BSc, PhD	Computer graphics, visualisation, database management.
V. Wang, BSc PhD	Cyber communities. Cybercrime. Technology studies

ASSOCIATE ACADEMIC STAFF
Honorary Professor, Honorary Fellow

P. Townsend, BSc, PhD, CEng, MBCS Computational fluid dynamics. Computer graphics. Human computer interaction.

Honorary Visiting Professors

J. A. Bergstra, MSc, PhD
Professor of programming,
University of Amsterdam, The Netherlands. Abstract data types. Concurrent process theory.

R Beale, BSc DPhil FBCS FRGS MACM CITP
School of Computer Science
University of Birmingham

M. D. Harrison, BA DPhil MBCS CEng
Professor of Informatics
Newcastle University

C. M. N. Tofts, PhD, MBA, ScD, FBCS, FIMA
Chief Mathematics Officer, Concinnitas Service Science

J. I. Zucker, BSc, PhD
Department of Computer Science
McMaster University, Canada

Theory of computation.

Honorary Lecturer

K. Stephenson, BSc, PhD
Qinetiq

High integrity systems. Programming and specification languages.

Honorary Research Fellow

R Morris, MPhil FRPS

History of visual images. History of science in Wales.

ITWALES STAFF

Software Alliance Wales (SAW)

Dr. N. Harman	Director
Ms. S. Govindasamy	Project Manager
Mr. A. Richards	Operations Manager
Mr. M. Kiddell	Technical Services Manager
Mrs. S. Earls	Communications Manager
Dr. M. Roach	Business Services Manager
Prof. M. Moller	Regional Interface Officer
Mrs. C. Spearing	Communications Coordinator
Mrs. H. Keller (Robinson)	Communications Assistant
Mrs. E. Waters	Finance Support Assistant
Mrs. H. Doidge	Admin Support Assistant

Technocamps

Prof. F. Moller	Director
Mr. S. Toomey	Project Manager
Mr. A. Richards	Operations Manager
Mr. M. Kiddell	Technical Services Manager
Mrs. S. Jones	Communications Manager
Miss. A. Dignam	Workshop Developer
Dr. R. Pau	Workshop Developer
Mr. E. O'Regan	Creative Design Officer
Mr. M. Erskine	Communications Coordinator
Dr. V. Wang	Admin Support Officer
Miss. J. O'Byrne O'Keefe	Finance Support Assistant

Go Wales

Dr. N. Harman	Director
Mrs. L. Nash	Placement Officer
Miss. K. Birdsall	Placement Assistant

Swansea University History of Computing Collection

Prof. J. V. Tucker
Dr. S. R. Williams
Dr. T. Davies
Dr. C. Evans
Dr. P. W. Grant

Chair
Library Information Services
Engineering
Physics
Computer Science

Computers have transformed many aspects of science, society and culture since the 1940s. In each decade, the forces of change have intensified and show no sign of relaxing. The transformation is phenomenal and difficult to comprehend: there is a great need to study these technologies, their development and impact, from different points of view. The history of the transformation is not well documented, let alone analysed. There is a great need to rescue this history for our contemporaries and future generations to reflect upon. The purpose of the collection is to

1. Rescue, collect, preserve, document, and make available materials charting the history of computing, including books, papers, ephemera, images, videos, manuals, software and hardware.
2. Work in partnership with museums, archives and learned societies to inform and educate scholars and students on the nature and history of technology.
3. Maintain a working collection that can be of use in connection with legacy systems and obsolete digital media.
4. Seek funding to grow and sustain the collection and associated activities, including commercial exploitation.
5. Promote research on the local history of computing.
6. Educate students in the History of Computing from technical, social, cultural and economic perspectives.
7. Engage the interest of the public.

The Collection will focus on:

- Developments in hardware;
- Developments in programming and programming languages;
- Developments in software;
- Developments in theoretical understanding;
- Legacy systems and obsolete media;
- The impact of computing on Science, Industry, Business, Society and Culture;
- The history of computing and its influence in Wales and, especially, Swansea Bay.

Its users will be professional scholars, local historians, and postgraduate and undergraduate students.

More information about the collection can be found at the website: <http://hocc.swan.ac.uk/>

CONTACT POINTS: DEPARTMENT

Head of Department:	Prof. Jones
Deputy Head of Department:	Dr. Harman
Director of Finance:	Dr. Grant
Director of Research:	Prof. Mosses
Undergraduate Recruitment, Admissions and Retention:	
Director	Dr. Jones
Applications	Mr. Whyley, Dr. Stein
Clearing	Dr. Stein, Mr. Whyley
Visit / Open Day Coordinator	Dr. Wilson
Visit / Open Day Team	Dr. Eslambolchilar, Mr. Whyley, Dr. Arandjelovic, Dr. Borgo
Enrolment and Induction Coordinator	Dr. Mora
Enrolment and Induction Team	Dr. Laramee, Dr. Blanck, Mr. Whyley, Heads of Year
Postgraduate Recruitment, Admissions and Retention:	
Director	Dr. Setzer
PhD Applications	Dr. Kullmann
Postgraduate Research Progression/Exams/Year Head	Dr. Berger
MSc Advanced Computer Science Coordinator <i>(including admissions/induction)</i>	Dr. Setzer
MSc Computer Science Coordinator <i>(including admissions/induction)</i>	Dr. Sharp
Research Studentships	Dr. Berger
Research:	
Director	Prof. Mosses
Group Seminar Coordinators	Dr. Borgo, Dr. Blanck, Dr. Wilson
Distinguished Lecture Series	Prof. Mosses
Teaching:	
Director	Dr. Harman
Year Heads	
Foundation	Dr. Xie
Year One	Dr. Berger
Year Two	Dr. Laramee
Year Three	Dr. Roggenbach
M. Eng	Dr. Eslambolchilar
PhD/MPhil	Dr. Berger
MSc Advanced Computer Science Coordinators	
HCI	Dr. Wilson
Web Science	Dr. Wilson
Visual Computing	Dr. Xie
Safe and Secure Systems	Dr. Setzer
Software Technology	Dr. Stein
Open Specialisation	Dr. Stein

MRes. Coordinators	
Computing and FIT	Dr. Wilson
Visual Computing	Dr. Xie
Logic and Computation	Dr. Kullmann
Timetables / Lecture Rooms	Dr. Laramee
Examinations and Assessments	
Chair of Exam Boards	Dr. Harman
College Liaison	Dr. Jones
Exam Papers	Dr. Blanck, (Dr. Seisenbeger)
Exam Results	Dr. Blanck, Dr. Jones
Student Monitoring	Mr. Whyley
College Liason: Special Needs	Dr. Mora
Events and Communications	
Coordinator	Prof. Moller
Handbook	Mr. Whyley
BSc Project Handbook	Dr. Roggenbach
MEng Project Handbook	Dr. Eslambochilar
MSc Project Handbook	Dr. Stein
First Year Events	Mr. Whyley, Dr. Berger
Second Year Events	Mr. Whyley, Dr. Laramee
Undergraduate Colloquium	Dr. Roggenbach, KE2
Project Demonstration Fair	Dr. Roggenbach, KE2
Infrastructure	
Director	Dr. Beckmann
Deputy Director	Dr. Jones
Department Website	Prof. Mosses
KE2	
Director	Prof. Tucker
SAW	Dr. Harman
Technocamps	Prof. Moller
Teaching and Industry Impact	Dr. Roggenbach
Task and Finish Work	
BCS Accreditation	Dr. Harman, Dr. Eslambolchilar
Erasmus Programme, Departments, external Schools and Colleges	Director of Programmes and Delegates
Library Liaison	Dr. Grant

CONTACT POINTS: UNIVERSITY

Emergency Services	333	
Admissions Office		
<i>Undergraduate Admissions</i>	5111	admissions@swansea.ac.uk
<i>Postgraduate Admissions</i>	5358	postgraduate.admissions@swansea.ac.uk
Careers Centre	3266	careers@swansea.ac.uk
Chaplaincy	4442	
Dental Surgery	2222	dentist@swansea.ac.uk
(Out of hours emergency: 0845 4647)		
Health Centre	5321	
(Out of hours emergency: 0845 8501362)		
Research and Innovation Office	5412	
Library and Information Services		
<i>Information Desk</i>	5697	library@swansea.ac.uk
<i>IT Support</i>	5060	itsupport@swansea.ac.uk
Safety Office	5152	safety@swansea.ac.uk
Shops and Services		
<i>Level 2 Shop</i>	5473	
<i>Post Office</i>	4373	
<i>Students' Union Travel Shop</i>	5476	
<i>Waterstone's Bookshop</i>	4374	enquiries@swan-uni.waterstones.co.uk
<i>Nursery</i>	3151	nursery@swansea.ac.uk
Student Volunteering Swansea (Discovery)	5743	discovery@swansea.ac.uk
Taliesin Arts Centre		
<i>Box Office</i>	2060	
<i>Gallery</i>	5526	
<i>Egypt Centre</i>	5960	
University Sports Centre		sportcentre@swansea.ac.uk
<i>Reception</i>	3555	
<i>Department of Sport and Physical Recreation</i>	3552	
<i>Wales National Pool</i>	3513	

UNIVERSITY SUPPORT FOR STUDENTS

ACADEMIC REGISTRY

Tel: 01792 513546

Fax: 01792 295157

Email: academic.registry@swansea.ac.uk

Web: <http://www.swan.ac.uk/registry/>

Opening times: Monday to Friday 10am-4pm

The Academic Registry, located in the Stable Block of Singleton Abbey, administers the following key areas that may directly affect your time at Swansea:

Appeals; Assessment; Attendance Monitoring; Complaints/Disciplinary; Council Tax Exemption; Diploma Supplements; Examinations; Graduation Ceremonies; Programme Specifications; Providing confirmation of student status; Suspension/Withdrawal of Studies; Transcripts; Transfers of Programmes/Modules; Unfair Practice/Fitness to Practise; Visa Extensions; Welsh Language Provision.

In addition, any queries that might result in having to change your personal or academic details may have to be referred to the Academic Registry.

INTERNATIONAL DEVELOPMENT OFFICE

Tel: 01792 602365

Email: international@swansea.ac.uk / erasmus@swansea.ac.uk

Web: <http://www.swan.ac.uk/internationalisation/>

The International Development Office contains the Study Abroad Unit which provides support, information and advice to students on international study and work placement opportunities available to them, including Swansea University's range of semester and year-long student exchange and study abroad programmes, the Erasmus scheme, and short term programmes including Study in India, Study in China and other summer schools.

STUDENT SUPPORT SERVICES

Student Support Services is part of the Student Services Directorate alongside Residential Services and the English Language Training Service (ELTS).

Tel: 01792 602000

Email: student.services@swansea.ac.uk

Web: <http://www.swan.ac.uk/study/current/StudentSupportServices/>

Student Support Services can be found in the Keir Hardie Building, Ground Floor. It offers a 'one-stop shop' to provide information, advice and support to students and staff working with them. It comprises:

The Disability Office

The International Student Advisory Service (ISAS)

The Money Advice and Support Office

The Student Wellbeing Service

Students can be referred to the department if they have a general enquiry, specific problem or just need to talk things through. Enquiries from staff seeking to support students are also welcome.

Support Services Diary

A Support Services Diary will be provided to all new students on their arrival at the University which provides a comprehensive list of support service contact information as well guidance on key dates and issues that may arise during the academic year.

Study Support

With the help and support of several academic departments, the subject teams in LIS and the Web Office, a list of on-line resources for Study Skills support for students has been established at

<http://www.swansea.ac.uk/study/current/StudentSupportServices/StudyAdvice/>

Disability Office

Tel: 01792 602000

Fax: 01792 295090

Minicom/text: 01792 513200/295090

Email: disability@swansea.ac.uk

Web: <http://www.swansea.ac.uk/study/current/StudentSupportServices/DisabilityOffice/>

The Disability Office both provides and co-ordinates support for students with disabilities. Reference to 'disability' includes those with any form of disability/ specific need/medical condition that requires support. This short note aims to give some basic information about the services available.

Support available

- Disability Caseworkers – provide a comprehensive advice and information service for students from initial enquiries prior to application and throughout a chosen course of study
- Specialist tuition – a team of Specialist Tutors provide group and individual support sessions
- IT training & technical support – training and technical support is available from experienced staff on all assistive software packages
- Support Schemes – Notetakers, Readers and Support Workers can all be arranged through the Disability Office
- Close links with the Well Being Service for students affected by mental health issues
- Assessment of needs / Disabled Students Allowance (DSA) – an Assessment Officer will undertake a comprehensive assessment of your needs to identify and recommend the best available technology and human support
- Alternative examination provision – in conjunction with the Examinations Office extra time, use of a computer & assistive software such as screen readers can be arranged
- Internal liaison – to establish and maintain support and adjustments by Academic Departments, the Accommodation Office, the Recording for the Blind Centre and the Examinations Office
- External liaison – to facilitate the necessary support from agencies, such as Local Education Authority, the RNIB, RNID, National Assembly
- Educational psychologist – the Disability Office can arrange for an assessment with an Educational Psychologist for students who suspect they may be dyslexic

Assessment and Training Centre for Students with Disabilities (ATC), Grove Building Extension.

The University established an Assessment and Training Centre for students with disabilities in the summer of 2002. All IT Training and Assessment of Needs/DSA Assessments are undertaken in this fully accessible and air-conditioned facility. The Assessment Officer can demonstrate the latest equipment and software to ensure students make better-informed decisions about the forms of support that will most appropriately meet their needs.

Transcription Centre

Tel: 01792 295912

Email: Braille@swansea.ac.uk

Web: <http://www.swan.ac.uk/lis/library/sutc/>

The Transcription Centre (formerly the Recording Centre for the Blind) works closely with

colleagues in the Disability Office and Library and Information Centre to provide materials in Braille, large print and/or tape.

For further information about the range of services on offer, please visit the website for a summary.

International Student Advisory Service (ISAS)

Tel: 01792 602000

Email: isas@swansea.ac.uk

Web: <http://www.swan.ac.uk/study/current/StudentSupportServices/ISAS/>

The International Student Advisory Service (ISAS) provides information, advice and support on non-academic matters to all applicants, students, staff members, visitors to the University, ethnic minority UK residents and their dependants. The service operates according to the UKCISA/AISA Code of Ethics for those advising international students (www.ukcisa.org.uk/join/code_of_ethics.php) and the Rules and Code of Standards of the Office of the Immigration Services Commissioner (www.oisc.gov.uk/). Common areas of advice include UK immigration and visa extensions, financial hardship, employment regulations and dependants.

ISAS operates the Home Office Batch Scheme to assist students with extending their visas, and arranges Police Registration on campus. Specific induction sessions are arranged for international students.

Students can drop in for advice on simple matters or make an appointment with an adviser to discuss more complicated matters in detail. The drop in service operates each weekday morning and appointments are scheduled in the afternoons.

International Student Handbook

The International Student Handbook is produced each year and sent to prospective students. It is also available on the University website at:

<http://www.swansea.ac.uk/study/current/StudentSupportServices/ISAS/Pre-ArrivalInformation/>

Money Advice and Support Office

Tel: 01792 602000

Email: moneydoctors@swansea.ac.uk

Web:

<http://www.swansea.ac.uk/study/current/StudentSupportServices/MoneyAdviceandSupport/>

Opening times: 9.00am-5.00pm (drop-in sessions Monday to Friday, 10.00am-12.30pm, except Wednesday)

The Money Advice and Support Office is on hand to provide information, advice and guidance in all areas of student funding, money management and financial assistance. The service is available to all Swansea University students who are facing financial difficulties or wish to manage their finances effectively. Staff within the office can offer advice and assistance with the following:-

- Money Doctors (including budgeting and debt advice)
- Applying for Support from your Local Authority (LA) or Student Finance England (SFE)
- Student Loan/Grant Queries
- Short-term Emergency Loans (for students whose loans are late through no fault of their own)
- Part-time Funding Queries
- Professional and Career Development Loans
- Entitlement to Benefits
- Banking Difficulties

- Application to the Financial Contingency Fund (FCF's)
- International Student Crisis Fund
- All aspects of Student Funding

The advice is free of charge and officers will provide open, honest and non-judgmental advice whilst taking the issue of confidentiality seriously.

In addition the Money Advice and Support Office has produced a number of leaflets that may be to your benefit, these are available on the website.

All contacts and communications made between student and advisors will remain in the strictest of confidence

Wellbeing Services

Tel: 01729 295592

Email: wellbeing@swansea.ac.uk

Web: <http://www.swansea.ac.uk/counselling/>

Opening times: 8.30am – 4.30pm (drop-in sessions 10.00am-12.15pm, 2.00pm-3.15pm, term-time Monday-Friday except Wednesday morning)

The Wellbeing Unit was created within Student Support Services in 2009 to help people feel better within themselves and function more effectively as members of the University.

Services available include:

- Counselling for staff and students
- Wellbeing advice and support

For more information on these Services visit the Wellbeing website.

ENGLISH LANGUAGE TRAINING SERVICES (ELTS)

Tel: 01792 295391

Email: elts@swansea.ac.uk

Web: www.swansea.ac.uk/elts

The University recognises that students from other countries whose first language is not English may need additional information and support.

English Language Training Services (ELTS) provides a variety of English language programmes for international students both before and during their studies. Students who do not meet the required English language entry level for their degree programme can attend a course in English for University Study for up to 12 months.

We recommend all international students whose first language is not English to attend a 5 week pre-session course to familiarise themselves with using English in an academic setting on a daily basis before they commence their undergraduate or postgraduate programmes.

For students who have already enrolled on a degree programme, there are free daily Academic English Language Support classes and grammar, speaking and writing workshops.

DATA PROTECTION

The University's procedures comply with the principles of the Data Protection Act 1998.

The responsibilities of students in relation to the provision of personal data can be found on the Records Management web pages:

<http://www.swan.ac.uk/university/Administration/RecordsManagement/>.

Students as data subjects have a right to request from the University a copy of their own personal data. A standard form must be completed and a fee of £10 is charged for each request. The University's registration number with the Information Commissioner is Z6102454.

DEPARTMENTAL COMMITTEES

Departmental Management Committee:

<i>Head of Department</i>	Prof. Jones (Chair)
<i>Deputy Head of Department and Programme Director</i>	Dr. Harman
<i>Director of Research</i>	Prof. Mosses
<i>Director of Finance</i>	Dr. Grant
<i>Coordinator of Undergraduate Admission</i>	Dr. Jones
<i>Director of Postgraduate Admission</i>	Dr. Setzer
<i>Director of Support Services</i>	Dr. Beckmann

Co-opted Members

<i>College Teaching Director</i>	Dr. Sharp (Secretary)
<i>College KE2 Director</i>	Prof. Tucker
<i>College Interdisciplinary</i>	Prof. Thimbleby
<i>Technocamps / Outreach</i>	Prof. Moller

Departmental Research Committee:

<i>Director</i>	Prof. Mosses (Chair)
<i>Representatives from Research Groups</i>	Prof. Tucker, Dr. Jones, Prof. Thimbleby
<i>PhD / MPhil Year Head</i>	Dr. Berger (Secretary)
<i>Head of Department</i>	Prof. Jones
<i>Director of Finance</i>	Prof. Jones
<i>Research Staff Representative</i>	Mr. B. Daubney
<i>Research Students Representative</i>	Mr. Edmunds
<i>Admin Support</i>	Janice Lewis
<i>Co-opted Members:</i>	
<i>Director of Support Services</i>	Dr. Beckmann

Departmental Board of Studies:

<i>Programme Director</i>	Dr. Harman (Chair)
<i>Coordinator of Undergraduate Admission</i>	Dr. Jones
<i>Coordinator of Postgraduate Admission</i>	Dr. Setzer
<i>Student Monitoring</i>	Mr. Whyley
<i>Disability Officer</i>	Dr. Berger
<i>Head of Department</i>	Prof. Jones
<i>Admin Support</i>	Mrs. Edwards
<i>Co-opted Members</i>	All other teaching staff College Director of Teaching and Learning

Ethics and Risk Assessment Committee

Prof. Jones (Chair)
Dr. Laramee, Dr. Wilson
Dr. Mora

Staff/Student Consultative Committee:

Dr. Jones (Chair)
Mrs. Edwards (Admin Support)
All Teaching Staff
All Support Staff
Student Representatives
LIS Representatives

Infrastructure Committee:

Director of Support Services
Director of Finance
Head of Department
Co-opted

Dr. Beckmann (Chair)
Dr. Grant
Prof. Jones
College Computing Specialist
Staff

STAFF DIRECTORY

	Room	Tel. Ext. No.	Email address
Head of Department			
Prof. M Jones	304	29/5259	matt.jones@swansea.ac.uk
Deputy Head of Department			
Dr. N. A. Harman	207	29/5394	n.a.harman@swansea.ac.uk
Deputy Pro-Vice Chancellor for High Performance Computing			
Prof. J. V. Tucker	309	29/5649	j.v.tucker@swansea.ac.uk
Professors			
Prof. F. G. Moller	213	29/5160	f.g.moller@swansea.ac.uk
Prof. P. D. Mosses	507	60/2249	p.d.mosses@swansea.ac.uk
Prof. H. W. Thimbleby	505	60/2299	h.thimbleby@swansea.ac.uk
Academic Staff			
Dr. O. Arandjelovic	Talbot 58	3383	
Dr. A. Beckmann	511	60/2370	a.beckmann@swansea.ac.uk
Dr. U. Berger	306	51/3380	u.berger@swansea.ac.uk
Dr. J. E. Blanck	301	60/2917	j.e.blanck@swansea.ac.uk
Dr. R. Borgo	Talbot 59	51/4534	r.borgo@swansea.ac.uk
Dr. P. Eslambolchilar	509	60/2658	p.eslambolchilar@swansea.ac.uk
Dr. P. W. Grant	312	29/5396	p.w.grant@swansea.ac.uk
Dr. M. W. Jones	305	51/3391	m.w.jones@swansea.ac.uk
Dr. O. Kullmann	212	51/3369	o.kullmann@swansea.ac.uk
Dr. R. S. Laramée	508	51/2609	r.s.laramée@swansea.ac.uk
Dr. B. Mora	303	29/5575	b.mora@swansea.ac.uk
Dr. M. Roggenbach	210	51/3578	m.roggenbach@swansea.ac.uk
Dr. M. Seisenberger	208	60/2131	m.seisenberger@swansea.ac.uk
Dr. A. G. Setzer	211	51/3368	a.g.setzer@swansea.ac.uk
Dr. J. A. Sharp	307	29/5560	j.a.sharp@swansea.ac.uk
Dr. R. D. Stein	302	51/3026	r.d.stein@swansea.ac.uk
Mr. C. J. Whyley	209	51/3576	c.j.whyley@swansea.ac.uk
Dr. M.L. Wilson	510	60/2611	m.l.wilson@swansea.ac.uk
Dr. X. Xie	Talbot 56	60/2916	x.xie@swansea.ac.uk
Support Staff			
Mrs. J. Edwards	206	29/5561	j.edwards@swansea.ac.uk
V. Hurst	505	60/2299	
Mr. A. Morris	Talbot 55a	29/5416	a.morris@swansea.ac.uk
Mrs. J. D. Pellard	206	4048	j.d.pellard@swansea.ac.uk
Mr. P. H. Roberts-Davies	403	29/5007	p.roberts-davies@swansea.ac.uk
Others			
General Student Enquiries	206	29/5651	cshelp@swansea.ac.uk
Departmental Fax		29/5708	
FIT Laboratory		2693	
Research Laboratory	402	4534/4566	
Research Laboratory(HPC)	215	4533	
Research Laboratory(5th Floor)	500	4026/4027	
Server Room		4060	
Visitor	308	3213	

ITWALES DIRECTORY

	Room	Tel. Ext. No.	Email address
Miss. K. Birdsall	215	60/6660	k.a.birdsall@swansea.ac.uk
Miss. A. Dignam	215	60/6891	a.j.dignam@swansea.ac.uk
Mrs. H. Doidge	215	60/5881	h.f.doidge@swansea.ac.uk
Mrs. S. Earls	215	60/3382	s.e.m.earls@swansea.ac.uk
Mr. M. Erskine	215		michael.erskine@technocamps.com
Ms. S. Govindasamy	215	60/6892	s.govindasamy@swansea.ac.uk
Dr. N. Harman	207	29/5394	n.a.harman@swansea.ac.uk
Mrs. S. Jones	215	60/6652	Sian.Jones@swansea.ac.uk
Mr. M. Kiddell	215	4532	m.kiddell@swansea.ac.uk
Mrs. H. Keller (Robinson)	215	606663	h.robinson@swansea.ac.uk
Prof. F. Moller	213	29/5160	f.g.moller@swansea.ac.uk
Dr. M. Moller	214	60/6659	M.Moller@swansea.ac.uk
Mrs. E. Nash	215	60/6658	e.nash@swansea.ac.uk
Miss. J. O'Byrne O'Keefe	214	60/6868	e.j.okeefe@swansea.ac.uk
Dr. R. Pau	215		reena.pau@technocamps.com
Mr. E. O'Regan	215	60/6893	e.oregan@swansea.ac.uk
Mr. A. M. C. Richards	214	29/5806	a.m.c.richards@swansea.ac.uk
Dr. M. Roach	215	60/6662	m.j.roach@swansea.ac.uk
Mrs. C. Spearing	215	59/2702	c.e.spearing@swansea.ac.uk
Mr. S. Toomey	215	60/6889	s.j.toomey@swansea.ac.uk
Dr. V. Wang	215		q.wang@swansea.ac.uk
Mrs. E. Waters	214	60/6657	e.m.waters@swansea.ac.uk

GENERAL INFORMATION

Contact Details

Staff are willing to make themselves available for individual consultation with students. In order to arrange a convenient time students are advised to either speak to the relevant member of staff at the end of a lecture or teaching session, or to send an email. For an appointment with the Head of College or Head of Department students are advised to contact Mrs. Edwards in Room 206.

Each student is assigned an individual e-mail address when they enrol. This is usually of the form 123456@swansea.ac.uk where 123456 is the student's university identity number. Staff will regularly e-mail students with information about the department/courses and will only use these university addresses. It is therefore very important that students read their e-mail accounts and the departmental notice boards daily. In addition, an increasing amount of official University information from your College, from the Administration and from Library and Information Services is sent only by email. Such official electronic communication from the University will only be sent to your University email account.

The main port of call for most problems and queries is the Student Enquiry Office in room 206. If the staff there cannot solve the problem they will direct students to someone who can. They also sell CDs and folders, etc, for the convenience of students.

The LIS science and engineering team can help with project related queries, e.g. bibliographic databases or difficulty in finding books. They can be contacted at scieng@swan.ac.uk.

Access and Communications

The department has two open access laboratories for undergraduate use. These are located on the second floor. Access to these is restricted to those students doing work required by the Computer Science Department only. Access hours are from 0830 hrs to 1800 hrs Monday to Friday. If a card is lost, stolen or fails to work it must be reported to the Enquiry Office **immediately**. Students found misusing the laboratories or allowing access to students from other departments will face severe disciplinary action. Notice boards for each level can also be found on the second floor. These contain coursework results and notices of general relevance for students.

Feedback

The Department allows several different methods by which students can express opinion/ask questions, etc, about wider ranging aspects of the Department.

1. Students are encouraged to discuss any matters during their regular tutorial meetings.
2. Each lecturer hands out a questionnaire at the end of each module. Students are encouraged to fill these in and comment on the courses anonymously.
3. Students elect two representatives from each level to represent them at regular staff/student consultative committee meetings.
4. A staff/student meeting is held for the entire first year at an Awayday, usually during semester one.
4. A staff/student meeting is held for the entire second year at an Awayday, usually during semester two.
5. A staff/student meeting is held for the entire third year at the department's Undergraduate Colloquium at Gregynog.

Communications

The Academic Registry produces and sends results letters to students. These letters will be sent

to your home address as captured on the central computer system. It is your responsibility to ensure that this information is up to date. If you require your results to be sent to an address other than your home address then you must submit an envelope to the Academic Registry by 17th June 2012 detailing the necessary address, your student number and your level of study.

Learning and Professional Development

During your time at University one of the most important skills you will develop is being able to take responsibility for, and manage, your own learning and development. Since the academic year 2005/06 all universities have been required to provide opportunities for their students to undertake this type of personal development activity throughout their degree programme. At Swansea we call this activity Learning and Professional Development. By taking advantage of the opportunities offered for LEAP you can make the most of your time at university in both your academic work and other activities.

In practical terms LEAP for students means being able to review how and what you are learning and then plan how you are going to use this knowledge to consolidate and improve your performance.

Resources to help you with Learning and Professional Development are provided through the Pebble Pad E-portfolio platform at <http://pebblepad.swansea.ac.uk>.

There are also online courses to support your LEAP and Employability available in the Virtual Learning Environment Blackboard. You will find these in the 'My Courses' section once you have logged into Blackboard. These courses link closely to PebblePad.

Extra-curricular activities, student jobs, and placements will also provide opportunities to use and further develop skills of interest to employers and you should use the LEAP process to review them regularly.

Through undertaking the LEAP process you will build up knowledge about yourself which as well as being useful in helping you to take responsibility for your own learning and make progress academically, will also help you to make decisions about your future after university and to be able to build an effective and informative CV.

DEGREE PROGRAMMES

INTRODUCTION

Educational Aims

The subject of Computer Science involves scientific foundations, new technologies, and advanced applications. Computer Science education at Swansea equips individual students with the necessary knowledge, skills and experience to embark on an intellectually satisfying career. In particular, it emphasises scientific curiosity, problem solving, rigorous thinking and an interest in engaging the world's work. In 1992, the Department made explicit its *nine educational aims* (see Page 2) for the benefit of applicants, students, staff, and the world at large.

Degree courses in Computer Science at Swansea are designed to achieve these aims in a coherent way. Naturally, all the modules contribute to our broad educational objectives. Some modules are directly relevant to a specific aim, whilst the others support several aims. An impression of how individual modules relate to the nine aims can be obtained from the tables on Pages 32 - 35.

Learning Outcomes

The learning outcomes associated with each module are the knowledge and capability that a typical student might reasonably be expected to achieve and demonstrate if he or she takes full advantage of the learning opportunities that are provided. A student who obtains a high (first or upper second) mark would be expected to have a greater understanding and appreciation of the material covered than suggested by the simple learning outcomes listed, whereas a student who is awarded a tolerated failure in a module may well not achieve all the listed outcomes.

Transferable Skills

Transferable skills are skills that may be acquired in one field (e.g., the academic field of Computer Science) and adapted to the demands of another field (e.g., some profession or field of employment). Many transferable skills are ones that all students can be expected to gain from following any degree course. Examples include managing one's time and the ability to find and assimilate information from many sources. Furthermore a student's ability to manage their learning would naturally be expected to increase significantly as they progress through a programme.

In a degree involving a significant Computer Science component it is reasonable to assume that virtually all modules studied will enhance a student's general IT skills, problem solving abilities, abstract modelling skills and formal reasoning abilities. The first two are especially associated with modules which involve practical coursework, and the latter two more with modules covering the more theoretical aspects of the subject.

In the module descriptions that follow we have attempted to list only those transferable skills which would be especially enhanced for a typical student taking that module. The transferable skills considered include:

1. *Communication* — (a) written communication and documentation, (b) oral presentation, (c) interactive discussions;
2. *Collaboration and Management* — (a) teamwork, (b) time management, (c) project management;
3. *Mathematical Skills* — (a) general mathematical discipline, (b) mathematical modelling and analysis;
4. *IT Skills* — (a) general IT skills, (b) ability to learn and use computer systems and software packages effectively, (c) ability to evaluate and deploy new technologies;
5. *Problem Solving* — (a) problem identification, (b) problem analysis, (c) abstract modelling, (d) formal reasoning, (e) solution formulation;
6. *Self-learning* — (a) information retrieval, (b) ability to read critically, to précis and judge information, (c) ability to manage learning processes.

Employment Skills

Our degree programmes educate individuals. They provide students with an opportunity to gain a deep knowledge and understanding of Computer Science, and to develop other intellectual and personal capabilities. This is evident from our educational aims (see above). Our degree programmes are intellectually demanding. Students should stretch themselves. As graduates, our students should be proud of their academic achievements and should be well prepared to make a strong contribution in any number of possible careers.

The aspirations of our students are diverse and change in the course of their studies. Students are advised to reflect on their personal progression, on their aspirations, and their prospects in employment. For help with the last, students are welcome to consult with the staff of the Careers Centre.

The world's work is diverse, but Computer Science is influential in many fields. Employers are diverse, but commonly they find the following properties attractive in potential colleagues:

1. Intellectual skills and academic ability;
2. Key skills in communication, co-operative working, numeracy, IT, problem solving and self-improvement;
3. Personal attributes such as self-reliance, confidence, drive, flexibility and creativity;
4. Knowledge of how organisations function.

Our degree programmes address the properties 1 and 2 explicitly, of course. They also provide an environment, and time, that can nurture some of the personal qualities mentioned in property 3. Finally, for property 4, our ITWales programme offers students excellent opportunities to gain experience with organisations, through summer work placements and industrially related academic projects.

Credit Points, Modules and Levels

Full-time students will normally be expected to pursue 120 credits of study each year. The University has introduced regulations to allow students to study on a part-time basis which would mean that they would study fewer than 120 credits in a year. Further details are available from the Academic Registry.

Students taking a three year honours degree will normally study 120 credits at Level 1 in their first year, 120 credits at Level 2 in their second year and 120 credits at Level 3 in their final year. Students who enrol on the Foundation year will normally take 120 credits at Level 0 before following the usual pattern of study for a three year degree. Students on a four year Advanced Initial Degree, e.g., an MEng, will take a further 120 credits at Level M in their fourth year of study.

For taught Masters degrees, 180 credits must be taken including the project which is usually 60 credits for MSc degrees and 120 credits for MRes degrees.

Throughout this handbook 10 credits from Swansea University are equivalent to 5 ECTS (European Credits Transfer Scheme) credits.

Module Numbering Scheme

All course modules within the University are allocated a unique six character code according to a University-wide convention.

The first two characters will indicate the department offering the module (CS in the case of Computer Science). The third character can be used to indicate a sub-department, and is currently unused for Computer Science modules. The fourth character indicates the level of the module, that is,

- 1 — Level One;
- 2 — Level Two;
- 3 — Level Three;
- M — taught masters, MEng or graduate modules.

The last two characters are for the use of the department and are used to uniquely identify modules.

COMPUTER SCIENCE MODULES AND EDUCATIONAL AIMS LEVELS ZERO, ONE AND TWO

Module Code	Module Name	Educational Aims									Module Code
		1	2	3	4	5	6	7	8	9	
CS-061	Introduction to Computing I	✓✓	✓	✓		✓					CS-061
CS-108	Computers and Computing	✓✓	✓					✓	✓		CS-108
CS-110	Programming I	✓✓	✓	✓		✓					CS-110
CS-113	From Languages to Hardware	✓	✓	✓							CS-113
CS-115	Programming II	✓✓	✓	✓		✓					CS-115
CS-130	Professional Issues I: Computers and Society		✓	✓✓		✓		✓✓	✓		CS-130
CS-135	Professional Issues II: Software Development	✓✓	✓	✓		✓		✓	✓		CS-135
CS-150	Concepts of Computer Science I	✓	✓	✓							CS-150
CS-155	Concepts of Computer Science II	✓	✓	✓							CS-155
CS-170	Modelling Computer Systems I	✓✓	✓	✓		✓✓					CS-170
CS-175	Modelling Computer Systems II	✓✓	✓	✓		✓✓					CS-175
CS-205	Declarative Programming	✓✓	✓✓	✓		✓					CS-205
CS-210	Concurrency	✓✓	✓	✓		✓					CS-210
CS-215	Logic Programming and Artificial Intelligence	✓	✓✓	✓		✓					CS-215
CS-217	Computer Graphics I: Image Processing and Synthesis	✓				✓				✓	CS-217
CS-219	Database Systems	✓				✓			✓	✓	CS-219
CS-230	Software Engineering I	✓✓	✓	✓		✓	✓✓	✓	✓	✓	CS-230
CS-235	Software Engineering II	✓✓	✓	✓		✓	✓✓	✓	✓	✓	CS-235
CS-236	Language and Computation	✓✓	✓		✓	✓✓	✓✓	✓	✓	✓	CS-236
CS-242	Algorithms	✓✓	✓	✓		✓✓		✓		✓	CS-242
CS-250	Database Systems	✓	✓	✓		✓			✓	✓	CS-250
CS-255	Computer Graphics	✓				✓				✓	CS-255
CS-270	Algorithms	✓✓	✓	✓		✓✓		✓		✓	CS-270
CS-275	Automata and Formal Language Theory	✓✓	✓		✓	✓✓		✓	✓	✓	CS-275

1. Keys:[✓✓ in white box] strongly directly relevant; [✓ in grey box] directly relevant; [dark grey box] not so directly relevant as a main educational aim.

2. The description of the nine educational aims is given on Page 2 of this handbook.

COMPUTER SCIENCE MODULES AND EDUCATIONAL AIMS

LEVEL THREE

Module Code	Module Name	Educational Aims									Module Code
		1	2	3	4	5	6	7	8	9	
CS-307	Computer Graphics II: Modelling and Rendering	✓✓	✓	✓	✓✓	✓✓			✓	✓	CS-307
CS-311	Concepts of Programming Languages	✓✓	✓✓	✓		✓✓					CS-311
CS-313	High Integrity Systems	✓✓	✓✓	✓		✓					CS-313
CS-318	Cryptography and IT Security	✓✓	✓	✓✓		✓✓		✓		✓	CS-318
CS-337	Data Visualisation	✓✓	✓	✓	✓		✓	✓	✓	✓	CS-337
CS-338	Internet Computing	✓✓	✓	✓✓							CS-338
CS-344	Project Implementation and Dissertation	✓	✓✓	✓	✓✓			✓✓	✓✓	✓✓	CS-344
CS-345	Artificial Intelligence Applications	✓✓	✓✓	✓✓		✓✓				✓	CS-345
CS-348	Building Reliable Web Applications	✓✓	✓	✓✓							CS-348
CS-349	Mobile Interaction Design	✓✓	✓	✓					✓	✓	CS-349
CS-354	Project Specification and Development	✓	✓✓	✓	✓✓			✓✓	✓✓	✓✓	CS-354
CS-358	High-Performance Computing in C/C++	✓✓	✓✓	✓✓	✓	✓					CS-358
CS-364	Software Testing	✓✓	✓	✓		✓			✓	✓✓	CS-364
CS-368	Embedded Systems	✓✓	✓✓	✓✓		✓			✓		CS-368
CS-371	Design Patterns and Generic Programming	✓✓	✓	✓		✓			✓	✓	CS-371
CS-375	Logic for Computer Science	✓	✓✓	✓		✓✓		✓	✓	✓	CS-375
CS-377	Computer Vision and Pattern Recognition	✓✓	✓✓	✓	✓	✓✓			✓	✓	CS-377
	Undergraduate Computer Science Colloquium	✓	✓	✓✓	✓		✓	✓✓	✓	✓	
	Project Demonstration Fair	✓	✓	✓✓	✓			✓✓	✓	✓	

1. Keys:[✓✓ in white box] strongly directly relevant; [✓ in grey box] directly relevant; [dark grey box] not so directly relevant as a main educational aim.

COMPUTER SCIENCE MODULES AND EDUCATIONAL AIMS

LEVEL M

Module Code	Module Name	Educational Aims									Module Code
		1	2	3	4	5	6	7	8	9	
CS-M00	Research Methodology	✓✓	✓	✓	✓✓	✓		✓✓	✓✓	✓✓	CS-M00
CS-M04	Group Project	✓✓	✓		✓✓		✓✓	✓	✓		CS-M04
CS-M05	Advanced Topics in Logic and Computation	✓	✓	✓				✓✓	✓✓	✓✓	CS-M05
CS-M07	Data Visualisation	✓✓	✓	✓	✓		✓	✓		✓	CS-M07
CS-M08	Future Interaction Technologies: MSc Project	✓✓	✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M08
CS-M09	Future Interaction Technologies: MRes Project	✓✓	✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M09
CS-M10	Computer Science Project Development	✓✓	✓	✓	✓✓	✓		✓✓	✓✓	✓✓	CS-M10
CS-M12	Software Concepts and Efficiency	✓	✓			✓✓				✓	CS-M12
CS-M13	Critical Systems	✓✓	✓✓	✓		✓			✓	✓	CS-M13
CS-M14	Industrial Project	✓✓	✓	✓	✓✓	✓✓		✓✓	✓✓	✓✓	CS-M14
CS-M15	Directed Studies in Logic and Computation	✓	✓	✓	✓✓	✓✓		✓✓	✓✓	✓✓	CS-M15
CS-M18	IT Security: Theory and Practice	✓✓	✓	✓✓		✓✓		✓	✓	✓	CS-M18
CS-M19	Interactive Systems Design	✓✓	✓✓		✓	✓✓		✓	✓	✓✓	CS-M19
CS-M20	MSc Project	✓	✓✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M20
CS-M24	Software Team Project	✓✓	✓		✓✓		✓✓	✓			CS-M24
CS-M29	Mobile Interaction Design	✓✓	✓	✓	✓✓				✓	✓	CS-M29
CS-M35	Logic and Computation MRes Project	✓	✓✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M35
CS-M39	Interaction Technologies: Seminars and Readings	✓✓	✓✓	✓✓	✓✓	✓	✓✓	✓✓	✓✓	✓✓	CS-M39
CS-M40	Software Technology Project Development	✓	✓✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M40
CS-M41	Programming in Java	✓✓	✓								CS-M41
CS-M49	Interaction Technologies Lab and Field Work	✓✓	✓✓		✓✓	✓	✓✓	✓✓		✓✓	CS-M49
CS-M50	Safe and Secure Systems Project Development	✓	✓✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M50
CS-M53	Computer Science Concepts	✓✓	✓✓	✓✓		✓			✓		CS-M53

1. Keys:[✓✓ in white box] strongly directly relevant; [✓ in grey box] directly relevant; [dark grey box] not so directly relevant as a main educational aim.

2. The description of the nine educational aims is given on Page 2 of this handbook.

COMPUTER SCIENCE MODULES AND EDUCATIONAL AIMS LEVEL M (CONTINUED)

Module Code	Module Name	Educational Aims									Module Code
		1	2	3	4	5	6	7	8	9	
CS-M58	Distributed O-O Programming	✓✓	✓	✓✓					✓	✓	CS-M58
CS-M59	Relational and Object-Oriented Databases	✓✓	✓			✓			✓	✓	CS-M59
CS-M61	Concepts of Programming Languages	✓✓	✓✓	✓		✓✓			✓	✓	CS-M61
CS-M65	Artificial Intelligence Applications	✓✓	✓✓	✓✓		✓			✓	✓	CS-M65
CS-M67	Graphics Processor Programming	✓✓	✓	✓✓		✓		✓	✓	✓✓	CS-M67
CS-M68	Writing Web and Web Service Applications	✓✓	✓	✓✓					✓	✓	CS-M68
CS-M69	Interaction Technologies: Information Retrieval	✓✓	✓✓	✓✓	✓	✓✓	✓✓	✓	✓	✓✓	CS-M69
CS-M70	Visual Computing Project Development	✓✓	✓		✓✓	✓✓		✓✓	✓	✓✓	CS-M70
CS-M71	Design Patterns and Generic Programming	✓✓	✓	✓		✓			✓	✓	CS-M71
CS-M75	Logic for Computer Science	✓✓	✓			✓✓				✓	CS-M75
CS-M77	Computer Vision and Pattern Recognition	✓	✓✓	✓	✓	✓✓			✓	✓	CS-M77
CS-M78	High Performance Computing in C/C++	✓✓	✓✓	✓✓	✓	✓					CS-M78
CS-M79	Interaction Technology: Hardware and Devices	✓✓	✓✓	✓✓	✓	✓	✓	✓	✓	✓✓	CS-M79
CS-M80	Web Science Project Development	✓✓	✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M80
CS-M84	Software Testing	✓✓	✓			✓✓			✓	✓	CS-M84
CS-M85	Modelling and Verification Techniques	✓✓	✓			✓✓				✓	CS-M85
CS-M87	Computer Graphics, MRes Visual Computing Project	✓✓	✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M87
CS-M88	Embedded Systems	✓✓	✓✓	✓✓		✓			✓		CS-M88
CS-M89	Advanced Database Systems	✓	✓	✓		✓			✓		CS-M89
CS-M90	Human Computer Interaction Project Development	✓✓	✓		✓✓	✓✓		✓✓	✓✓	✓✓	CS-M90
CS-M94	Software Engineering Principles	✓✓	✓✓	✓	✓	✓	✓✓	✓	✓	✓	CS-M94

COMPUTER SCIENCE MODULES FOR PROGRAMMES LEVELS ONE AND TWO

Module Code	Module Name	Semester	G400	G401	G403	G420	G4GC	G4L1	G4Rn	G4G1	GF48	CG84	FG34	RnG4	Module Code
CS-108	Computers and Computing	S1				1		1	1		1				CS-108
CS-110	Programming I	S1	1	1	1	1	1	1	1	1	1	1	1	1	CS-110
CS-113	From Languages to Hardware	S1				1		1	1					1	CS-113
CS-115	Programming II	S2	1	1	1	1	1	1	1	1	1	1	1		CS-115
CS-130	Professional Issues I	S1	1	1	1	1		1	1			1			CS-130
CS-135	Professional Issues II	S2	1	1	1	1		1							CS-135
CS-150	Concepts of Computer Science I	S1	1	1	1		1			1			1		CS-150
CS-155	Concepts of Computer Science II	S2	1	1	1		1			1			1	1	CS-155
CS-170	Modelling Computer Systems I	S1	1	1	1										CS-170
CS-175	Modelling Computer Systems II	S2	1	1	1										CS-175
CS-205	Declarative Programming	S2				2			2				2	2	CS-205
CS-210	Concurrency	S1	2	2	2	2	2	2	2	2	2	2	2	2	CS-210
CS-215	Logic Programming and Artificial Intelligence	S2	2	2	2		2			2	2	2			CS-215
CS-217	Computer Graphics I: Image Processing and Synthesis	S2					2				2	2	2	2	CS-217
CS-219	Database Systems	S1					2	2			2	2	2	2	CS-219
CS-230	Software Engineering I	S1	2	2	2	2	2	2	2		2	2	2	2	CS-230
CS-235	Software Engineering II	S2	2	2	2	2	2	2	2		2	2		2	CS-235
CS-236	Language and Computation	S2	2	2	2		2			2	2	2	2	2	CS-236
CS-242	Algorithms	S1					2				2	2	2	2	CS-242
CS-250	Database Systems	S1	2	2	2	2	2		2		2	2	2	2	CS-250
CS-255	Computer Graphics	S2	2	2	2	2	2		2	2	2	2		2	CS-255
CS-270	Algorithms	S1	2	2	2	2	2	2	2	2	2	2		2	CS-270
CS-275	Automata and Formal Language Theory	S2				2	2		2	2	2	2		2	CS-275

1. Keys:[S1]Semester 1;[S2]Semester 2;[WS]whole session;[white box]compulsory; [grey box]optional;[dark grey box]not available;[the number in a box]the level at which a module may be taken.
2. Programmes involving subjects other than Computer Science will have additional compulsory modules.

COMPUTER SCIENCE MODULES FOR PROGRAMMES LEVEL THREE

Module Code	Module Name	Semester														Module Code
			G400	G401	G403	G420	G4GC	G4L1	G4Rn	GC41	GF48	CG84	FG34	RnG4		
CS-307	Computer Graphics II: Modelling and Rendering	S1	3	3	3	3	3		3	3	3	3	3	3		CS-307
CS-311	Concepts of Programming Languages	S2	3	3	3	3	3		3	3	3	3	3	3		CS-311
CS-313	High Integrity Systems	S1	3	3	3	3	3	3	3	3	3	3	3			CS-313
CS-318	Cryptography and IT Security	S2	3	3	3	3	3	3	3	3	3	3	3			CS-318
CS-337	Data Visualisation	S1	3	3	3	3	3	3	3	3	3	3	3	3		CS-337
CS-338	Internet Computing	S1	3	3	3	3	3	3	3	3	3	3	3	3		CS-338
CS-344	Project Implementation and Dissertation	WS	3	3	3	3	3	3	3							CS-344
CS-345	Artificial Intelligence Applications	S2	3	3	3	3	3	3	3	3	3		3	3		CS-345
CS-348	Building Reliable Web Applications	S2	3	3	3	3	3	3	3	3	3	3	3	3		CS-348
CS-349	Mobile Interaction Design	S2	3	3	3	3	3		3	3	3	3	3	3		CS-349
CS-354	Project Specification and Development	WS	3	3	3	3	3	3	3							CS-354
CS-358	High-Performance Computing in C/C++	S2	3	3	3	3	3		3	3	3	3	3	3		CS-358
CS-364	Software Testing	S2	3	3	3	3	3		3	3	3	3	3	3		CS-364
CS-368	Embedded Systems	S1	3	3	3	3	3	3	3	3	3	3	3	3		CS-368
CS-371	Design Patterns and Generic Programming	S2	3	3	3	3	3		3	3	3	3	3	3		CS-371
CS-375	Logic for Computer Science	S2	3	3		3	3	3	3	3	3	3	3	3		CS-375
CS-377	Computer Vision and Pattern Recognition	S1	3	3		3	3	3	3	3	3	3	3	3		CS-377

1 Keys:[S1]Semester 1;[S2]Semester 2;[WS]whole session;[V]variable semester;[white box]compulsory; [grey box]optional;[dark grey box]not available;[the number in a box]the level at which a module may be taken.

2. Programmes involving subjects other than Computer Science will have additional compulsory modules.

COMPUTER SCIENCE MODULES FOR PROGRAMMES LEVEL M

Module Code	Module Name	Semester	MEng	MSc CS	MSc CAST	MSC CFTT	MRes CFTT	MRes VC	MRes LaC	MSc AST	MSc AHCI	MSc AVC	MSc ASSC	MSc AWS	MSc AO	Module Code
CS-M00	Research Methodology	S1														CS-M00
CS-M04	Group Project	WS														CS-M04
CS-M05	Advanced Topics in Logic and Computation	S1														CS-M05
CS-M07	Data Visualisation	S1														CS-M07
CS-M08	Future Interaction Technologies: MSc Project	S2														CS-M08
CS-M09	Future Interaction Technologies: MRes Project	S2														CS-M09
CS-M10	Computer Science Project Development	S2														CS-M10
CS-M12	Software Concepts and Efficiency	S2														CS-M12
CS-M13	Critical Systems	S1														CS-M13
CS-M14	Industrial Project	WS														CS-M14
CS-M15	Directed Studies in Logic and Computation	S2														CS-M15
CS-M18	IT Security: Theory and Practice	S2														CS-M18
CS-M19	Interactive Systems Design	WS														CS-M19
CS-M20	MSc Project	SV														CS-M20
CS-M24	Software Team Project	S2														CS-M24
CS-M29	Mobile Interaction Design	S2														CS-M29
CS-M35	Logic and Computation: MRes Project	WS														CS-M35
CS-M39	Interaction Technologies: Seminars and Reading	WS														CS-M39
CS-M40	Software Technology Project Development	S2														CS-M40
CS-M41	Programming in Java	S1														CS-M41
CS-M49	Interaction Technologies: Lab and Field Work	WS														CS-M49
CS-M50	Safe and Secure Systems Project Development	S2														CS-M50
CS-M53	Computer Science Concepts	S1														CS-M53

Key:[S1]Semester 1;[S2]Semester 2;[WS]whole session;[SV]Summer Vacation;[white box]compulsory; [grey box]optional;[dark grey box]not available;[blue box]pathway module.

BSc COMPUTER SCIENCE

G400

	credit	page
Level One		
CS-110	Programming I	15
CS-115	Programming II	15
CS-130	Professional Issues I: Computers and Society	15
CS-135	Professional Issues II: Software Engineering	15
CS-150	Concepts of Computer Science I	15
CS-155	Concepts of Computer Science II	15
CS-170	Modelling Computer Systems I	15
CS-175	Modelling Computer Systems II	15
Level Two		
CS-210	Concurrency	15
CS-215	Logic Programming and Artificial Intelligence	10
CS-230	Software Engineering I	15
CS-235	Software Engineering II	15
CS-236	Language and Computation	20
CS-250	Database Systems	15
CS-255	Computer Graphics	15
CS-270	Algorithms	15
Level Three - 120 credit points must be taken.		
<i>Compulsory</i>		
CS-344	Project Implementation and Dissertation (Core)	20
CS-354	Project Specification and Development	20
<i>Optional</i>		
(80 credits must be taken from the list on page 51)		

MENG COMPUTING

G403

	credit points	page no.
Level One		
Same as BSc Computer Science (G400)	120	
Level Two		
Same as BSc Computer Science (G400)	120	
Level Three		
Same as BSc Computer Science (G400)	120	
Level Four - 120 credit points must be taken		
<i>Compulsory:</i>		
CS-M04 Group Project	20	
CS-M14 Industrial Project (core)	40	
 <i>60 credits from: other MSc modules listed on pages 129 - 173 and Level Four Business Options</i>		+

✦ For further details of Business modules consult the School of Business and Economics

BSC COMPUTING AND COMMUNICATIONS

G420

	credit points	page no.
Level One		
CS-108 Computers and Computing	10	
CS-110 Programming I	15	
CS-113 From Languages to Hardware	10	
CS-115 Programming II	15	
CS-130 Professional Issues I: Computers and Society	15	
CS-135 Professional Issues I: Software Development	15	
EG-143 Digital System Design	10	♦
EG-150 Signals and Systems B	10	♦
EG-189 Engineering Analysis I	10	♦
EG-190 Engineering Analysis II	10	♦
Level Two		
CS-210 Concurrency	15	
CS-230 Software Engineering I	15	
CS-235 Software Engineering I	15	
CS-270 Algorithms	15	
EG-245 Communications	10	♦
EG-247 Signals and Systems	10	♦
EE-259 Web Applications Technology	10	♦
<i>Two modules from</i>		
CS-205 Declarative Programming	15	
CS-250 Database Systems	15	
CS-255 Computer Graphics	15	
CS-275 Automata and Language Theory	15	
Level Three - 120 credit points must be taken		
<i>Compulsory:</i>		
CS-338 Internet Computing	10	
CS-344 Project Implementation and Dissertation (core)	20	
CS-354 Project Specification and Development	20	
<i>30 - 40 credit points from Computer Science modules listed on page 51</i>		
<i>30 - 40 credit points from suitable Engineering modules:</i>		
		♦

♦ For further details of electronics modules contact the College of Engineering.

BSC MATHEMATICS FOR COMPUTER SCIENCE

G4GC

	credit points	page no.
Level One		
CS-110 Programming I	15	
CS-115 Programming II	15	
CS-150 Concepts of Computer Science I	15	
CS-155 Concepts of Computer Science II	15	
<i>60 credits compulsory from Mathematics</i>	60	♦
Level Two		
CS-210 Concurrency	15	
CS-219 Database Systems	10	
CS-236 Language and Computation	20	
CS-270 Algorithms	15	
<i>60 credits from appropriate Computer Science and Mathematics modules</i>		
Level Three - 120 credit points must be taken		
<i>Compulsory</i>		
CS-344 Project Implementation and Dissertation (core)	20	
CS-354 Project Specification and Development	20	
<i>80 credit points from modules listed on page 51 and Appropriate Mathematics Options</i>		
		♦

♦ For details of mathematics modules contact the Department of Mathematics.

BSc COMPUTING WITH FINANCE

G4L1

	credit points	page no.
Level One		
CS-108	Computers and Computing	10
CS-110	Programming I	15
CS-113	From Languages to Hardware	10
CS-115	Programming II	15
CS-130	Professional Issues I	15
CS-135	Professional Issues II	15
EC-112	Current Issues in Economics	20 ♦
<i>One of the following modules depending upon A-level qualifications:</i>		
EC-100	Principles of Economics A (core)	20 ♦
EC-103	Principles of Economics B (core)	20 ♦
Level Two		
CS-210	Concurrency	15
CS-219	Database Systems	10
CS-230	Software Engineering I	15
CS-235	Software Engineering II	15
CS-270	Algorithms	15
EC-200	Intermediate Microeconomics	20 ♦
EC-244	Managerial Economics	20 ♦
EBG202	Statistics for Business and Economics	10 ♦
Level Three - 120 credit points must be taken		
<i>Compulsory:</i>		
CS-344	Project Implementation and Dissertation (core)	20
CS-354	Project Specification and Development	20
EC-338	Financial Economics	10 ♦
EC-339	Financial Market Efficiency	10 ♦
<i>40 - 60 credits from:</i>		
CS-313	High Integrity Systems	10
CS-318	Cryptography and IT Security	10
CS-338	Internet Computing	10
CS-345	Artificial Intelligence Applications	10
CS-348	Building Reliable Web Applications	10
CS-349	Mobile Interaction Design	10
<i>Up to 20 credits from appropriate Economics modules</i>		♦

- ♦ For details of Economics modules consult the Department of Economics
- ♦ For details of Finance modules consult the Department of Finance

BSC COMPUTER SCIENCE WITH A MODERN LANGUAGE

Single language may be **French (G4R1)**, **German (G4R2)**, **Italian (G4R3)**, **Spanish (G4R4)**, or **Welsh (G4Q5)**

	credit points	page no.
Level One		
CS-108 Computers and Computing	10	
CS-110 Programming I	15	
CS-113 From Languages to Hardware	10	
CS-115 Programming II	15	
CS-130 Professional Issues I: Computers and Society	15	
CS-135 Professional Issues II: Software Development	15	

Modern Language	40	♦
-----------------	----	---

*Beginners in **Welsh** and those taking Welsh as a second language will be required to complete the Transitional Year in that language before entering Level Two.*

Level Two - 120 credit points must be taken

Compulsory:

CS-230 Software Engineering I	15	
CS-235 Software Engineering II	15	
CS-250 Database Systems	15	
CS-270 Algorithms	15	

Two modules from

CS-210 Concurrency	15	
CS-215 Declarative Programming	15	
CS-255 Computer Graphics	15	
CS-275 Automata and Language Theory	15	
Modern Language	30	♦

Level S/E (Intercalary or third year of study)

This year is usually spent at an appropriate institution in the appropriate country

Level Three (fourth year of study) - 120 credit points must be taken

Compulsory:

CS-344 Project Implementation and Dissertation (core)	20	97
CS-354 Project Specification and Development	20	102
Modern Language	30	♦

50 credits from remaining Level Three Computer Science modules listed on page 51

♦ For details of Language modules consult the College of Arts and Humanities

BSC COMPUTER SCIENCE AND PURE MATHEMATICS GG41

		credit points	page no.
Level One			
CS-110	Programming I	15	
CS-115	Programming II	15	
CS-150	Concepts of Computer Science I	15	
CS-155	Concepts of Computer Science II	15	
MA-101	Introductory Calculus	15	♦
MA-102	Introductory Analysis	15	♦
MA-111	Foundations of Algebra	15	♦
MA-112	Introductory Linear Algebra	15	♦

Level Two

Compulsory:

CS-210	Concurrency	15	
CS-215	Logic Programming and Artificial Intelligence	10	
CS-236	Language and Computation	20	
CS-270	Algorithms	15	

25 credits from

CS-250	Database Systems	10 or 15	
CS-255	Computer Graphics I	10 or 15	

Level Three - *120 credit points must be taken, including at least 40 Computer Science and 40 Mathematics credit points*

Compulsory:

MAC321	Analysis 3	10	♦
MAP345	Rings, Modules and Categories	10	♦

Selection of between 40 and 80 credits from level 3 Computer Science modules from page 51, and the remainder chosen from Mathematics

♦ For details of mathematics modules contact the Department of Mathematics.

BSC COMPUTER SCIENCE AND GEOINFORMATICS

GF48

	credit	page
Level One		
CS-108 Computers and Computing	10	
CS-110 Programming I	15	
CS-115 Programming II	15	
GEG101 Earth in action: an introduction to Earth Surface Processes	10	♦
GEG102 Earth	10	♦
GEG103 Global environmental change: The human impact	10	♦
GEG106 Sustainability in a fragile world	10	♦
GEG107 Geographical methods 1: Practical Skills	10	♦
GEG108 Geographical Methods II	10	♦
Level Two		
CS-215 Logic Programming and Artificial Intelligence	10	
<i>60 credit points from Geoinformatics</i>		
<i>50 credit points from Computer Science</i>		
Level Three - 120 credit points must be taken		
<i>Compulsory:</i>		
Geoinformatics	60	♦
<i>60 credit points from Computer Science modules listed on page 51</i>		

♦ For details of topographic science modules contact the Geography Department.

BSC COMPUTER SCIENCE AND PSYCHOLOGY

CG84

	credit	page no.
Level Three		
<i>Compulsory:</i>		
CS-345 Artificial Intelligence Applications	10	99
PS-325 Project for Joint Honours	15	♦
PS-395 Brain & Behaviour	15	♦
PS-396 Individual Differences (B)	15	♦
<i>55 credit points from suitable Computer Science and Psychology modules</i>		
Computer Science modules are listed on page 51		

Note: This course is no longer available to new applicants

♦ For details of psychology modules contact the College of Health and Human Sciences.

BSc COMPUTER SCIENCE AND PHYSICS

FG34

	credit points	page no.
Level One		
CS-110	Programming I	15
CS-115	Programming II	15
CS-150	Concepts of Computer Science I	15
CS-155	Concepts of Computer Science II	15
PH-101	Dynamics I	10
PH-102	Vibrations and Waves?????	10
PH-104	Introduction to Astronomy and Cosmology	10
PH-108	Experimental Techniques II?????	20
	<i>Other Level One Modules</i>	10
Level Two		
CS-230	Software Engineering I	15
CS-235	Software Engineering II	15
CS-255	Computer Graphics	15
CS-270	Algorithms	15
PH-203	Statistical Physics	10
PH-205	Quantum Mechanics I	10
PH-206	Mathematical Methods in Physics I	10
PH-210	Laboratory Physics and Group Projects B	10
PH-227	Mathematical Methods in Physics II	10
PH-229	Particle Physics I	10
Level Three - 120 credit points must be taken		
<i>Compulsory:</i>		
PH-207	Condensed Matter Physics I	10
PH-221	Electromagnetism and Special Relativity I	10
PH-222	Electromagnetism I I	10
PH-302	Quantum Mechanics II	10
PH-306	Atomic Physics I	10
PH-311	Project	20
Plus one further Level 3 Physics module (10 credits)?????		
Plus Level 3 Computer Science modules totalling 50 credits?????		
Plus Level 3 Computer Science or Physics modules totalling a further 20 credits?????		
<i>(for Computer Science options see page 51)</i>		

- ◆ For details of physics modules contact the Department of Physics.

BA ONE MODERN LANGUAGE WITH COMPUTER STUDIES

BA TWO MODERN LANGUAGES WITH COMPUTER STUDIES

The single language studied may be French (R1G4), German (R2G4), Italian (R3G4), Spanish (R4G4), or Welsh (Q5G4)

The two languages studied may be French and German (RRD2), French and Italian (RRD3), French and Spanish (RRD4), French and Welsh (RQC5), German and Italian (RRG3), German and Spanish (RRG4), German and Welsh (RQG5), Italian and Spanish (RRJ4), Italian and Welsh (RQH5), or Spanish and Welsh (RQK5).

Level One	credit points	page no.
One or Two Modern Languages	80	♦
CS-110 Programming I	15	
CS-113 From Languages to Hardware	10	
CS-155 Concepts of Computer Science II	15	

Beginners in Welsh and those taking Welsh as a second language will be required to complete the Transitional year in that language before entering Level Two.

Level Two (second year of study)

30 - 40 credits from Computer Science modules for which the pre-requisites are satisfied, to include

CS-250 Database Systems	15
-------------------------	----

(Intercalary or third year of study)

This year is usually spent at an appropriate institution in the appropriate country

Level Three (fourth year of study)

120 credit points including at least 30 Computer Science credit points and no more than 20 credit points at Level Two

✦

♦For details of language modules contact the School of Arts.

✦For details of computer science modules see list on page 51.

LEVEL THREE MODULES FOR VARIOUS DEGREE PROGRAMMES

Level Three		points	no.
CS-307	Computer Graphics II: Modelling and Rendering	10	
CS-311	Concepts of Programming Languages	10	
CS-313	High Integrity Systems	10	
CS-318	Cryptography and IT Security	10	
CS-337	Data Visualisation	10	
CS-338	Internet Computing	10	
CS-345	Artificial Intelligence Applications	10	
CS-348	Building Reliable Web Applications	10	
CS-349	Mobile Interaction Design	10	
CS-358	High-Performance Computing in C/C++	10	
CS-364	Software Testing	10	
CS-368	Embedded Systems	10	
CS-371	Design Patterns and Generic Programming	10	
CS-375	Logic for Computer Science	10	
CS-377	Computer Vision and Pattern Recognition	10	
MS-306	New Screen Technologies	20	
MS-353	Digital Philosophy: The Roots of the Virtual	20	
HIP300	The Classical Tradition in the Sciences	10	
HIP301	From Natural Philosophy to Science	10	

Students may take a maximum of 30 credits from MS-306, MS-353, HIP300 and HIP301

Level three Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

MSC COMPUTER SCIENCE

	credit points	page no.
Compulsory Modules		
Part 1		
Semester 1		
CS-M00 Research Methodology	10	
CS-M53 Computer System Concepts	10	
Semester 2		
CS-M10 Computer Science Project Development	10	
CS-M12 Software Concepts and Efficiency	10	
Whole Session		
CS-M94 Software Engineering Principles	10	
Part 2 (Summer vacation)		
CS-M20 MSc Project	60	
Optional Modules (70 credits must be taken)		
Semester 1		
CS-M07 Data Visualisation	10	
CS-M13 Critical Systems	10	
CS-M58 Distributed Object-Oriented Programming	10	
CS-M59 Relational and Object-Oriented Databases	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M75 Logic for Computer Science	10	
CS-M77 Computer Vision and Pattern Recognition	10	
CS-M88 Embedded Systems	10	
Semester 2		
CS-M18 IT Security: Theory and Practice	10	
CS-M29 Mobile Interaction Design	10	
CS-M61 Concepts of Programming Languages	10	
CS-M68 Writing Web and Web Service Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M78 High Performance Computing in C/C++	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	
CS-M84 Software Testing	10	
CS-M85 Modelling and Verification Techniques	10	
CS-M89 Advanced Database Systems	10	
Whole Session		
CS-M19 Interactive Systems Design	10	
CS-M67 Graphics Processor Programming	10	
<i>Optional modules may include up to 20 credits from</i>		
MSDM01 Thinking about Digital Media	20	
ASCM17 Criminology/Understanding Crime	20	
GEGM13 Urban Networks and the Knowledge Economy	20	
LALM121 Contract and E-commerce Law	10	
PO-M52 Rethinking Global Politics	20	
EBCM10 E-Business	10	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

MSC / DIPLOMA / CERTIFICATE

COMPUTING AND SOFTWARE TECHNOLOGY

	credit points	page no.
Compulsory Modules		
Part 1		
Semester 1		
CS-M00 Research Methodology	10	
Semester 2		
CS-M24 Software Team Project	10	
CS-M40 Software Technology Project Development	10	
Part 2 (Summer vacation)		
CS-M20 MSc Project	60	
Optional Modules (90 credits must be taken)		
Semester 1		
CS-M07 Data Visualisation	10	
CS-M13 Critical Systems	10	
CS-M58 Distributed Object-Oriented Programming	10	
CS-M59 Relational and Object-Oriented Database Systems	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M77 Computer Vision and Pattern Recognition	10	
CS-M88 Embedded Systems	10	
Semester 2		
CS-M18 IT Security: Theory and Practice	10	
CS-M29 Mobile Interaction Design	10	
CS-M61 Concepts of Programming Languages	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M68 Writing Web and Web Service Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M75 Logic for Computer Science	10	
CS-M78 High Performance Computing in C/C++	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	
CS-M84 Software Testing	10	
CS-M85 Modelling and Verification Techniques	10	
CS-M89 Advanced Database Systems	10	
Whole Session		
CS-M19 Interactive Systems Design	10	
CS-M67 Graphics Processor Programming	10	
<i>Optional modules may include up to 20 credits from</i>		
MSDM01 Thinking about Digital Media	20	
ASCM17 Criminology/Understanding Crime	20	
GEGM13 Urban Networks and the Knowledge Economy	20	
LALM121 Contract and E-commerce Law	10	
PO-M52 Rethinking Global Politics	20	
EBCM10 E-Business	10	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

MSc / DIPLOMA / CERTIFICATE

COMPUTING AND FUTURE INTERACTION TECHNOLOGY

	credit points	page no.
Compulsory Modules		
CS-M08 MSc Project	10	
CS-M19 Interactive Systems Design		
CS-M25 Research Methodology and Project Specification		
CS-M29 Mobile Interaction Design		
CS-M39 Interaction Technologies: Seminars and Readings		
CS-M49 Interaction Technologies: Lab and Field Work		
CS-M69 Interaction Technologies: Information Retrieval		
CS-M79 Interaction Technologies: Hardware and Devices		
Optional Modules (<i>180 credits must be taken in total</i>)		
CS-M07 Data Visualisation	10	
CS-M13 Critical Systems	10	
CS-M18 IT Security Theory and Practice		
CS-M58 Distributed Object-Oriented Programming	10	
CS-M61 Concepts of Programming Languages		
CS-M65 Artificial Intelligence Applications		
CS-M68 Writing Web and Web Service Applications		
CS-M71 Design Patterns and Generic Programming		
CS-M77 Computer Vision and Pattern Recognition		
CS-M89 Advanced Database Systems	10	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

MRES / CERTIFICATE

COMPUTING & FUTURE INTERACTION TECHNOLOGIES

	credit points	page no.
Compulsory Modules		
CS-M00 Research Methodology	10	
CS-M09 Research Project (includes 10 credits from lab & field work)	120	
CS-M39 Interaction Technologies: Seminars & Readings	10	
CS-M49 Interaction Technologies: Lab & Field Work	10	
CS-M90 Human Computer Interaction Project Development	10	
Optional Modules - students take 180 credits in total		
CS-M19 Interactive Systems Design	10	
CS-M29 Mobile Interaction Design	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	

MRES/CERTIFICATE

VISUAL COMPUTING

	credit points	page no.
Compulsory Modules		
CS-M00 Research Methodology	10	
CS-M07 Data Visualisation	10	
CS-M57 Computer Graphics: MRes Visual Computing Project	120	
CS-M67 Graphics Processor Programming	10	
CS-M70 Visual Computing Project Development	10	
CS-M77 Computer Vision and Pattern Recognition	10	
Optional Modules - students take 180 credits in total		
CS-M58 Distributed Object-Oriented Programming	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M78 High Performance Computing in C/C++	10	
CS-M79 Interaction Technologies: Hardware and Devices	10	

MRES/CERTIFICATE

LOGIC AND COMPUTATION

Compulsory Modules		
CS-M05 Advanced Topics in Logic and Computation	10	
CS-M15 Directed Studies in Logic and Computation	10	
CS-M35 Logic and Computation Project	120	
Optional Modules - students take 180 credits in total		
CS-M13 Critical Systems	10	
CS-M18 IT Security: Theory and Practice	10	
CS-M61 Concepts of Programming Languages	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M75 Logic for Computer Science		
CS-M85 Modelling and Verification Techniques		

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them

MSC ADVANCED COMPUTER SCIENCE WITH SPECIALISATION IN SOFTWARE TECHNOLOGY

	credit points	page no.
Compulsory Modules		
CS-M00 Research Methodology	10	
CS-M24 Software Team Project	10	
CS-M40 Software Technology Project Development		
At least one of		
CS-M58 Distributed Object-Oriented Programming, or	10	
CS-M78 High Performance Computing in C/C++		
At least 20 credits from Pathway Modules		
CS-M13 Critical Systems	10	
CS-M19 Interactive Systems Design	10	
CS-M58 or CS-M78 (if not already taken)	10	
CS-M68 Writing Web and Web Service Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M84 Software Testing	10	
CS-M89 Advanced Database Systems	10	
Optional Modules - students must take 120 credits in total		
Semester 1		
CS-M07 Data Visualisation	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M77 Computer Vision and Pattern Recognition	10	
CS-M88 Embedded Systems	10	
Semester 2		
CS-M18 IT Security: Theory and Practice	10	
CS-M29 Mobile Interaction Design	10	
CS-M61 Concepts of Programming Languages	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M75 Logic for Computer Science	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	
CS-M85 Modelling and Verification Techniques	10	
Whole Session		
CS-M67 Graphics Processor Programming	10	
<i>Optional modules may include up to 20 credits from</i>		
MSDM01 Thinking about Digital Media	20	
ASCM17 Criminology/Understanding Crime	20	
GEGM13 Urban Networks and the Knowledge Economy	20	
LALM121 Contract and E-commerce Law	10	
PO-M52 Rethinking Global Politics	20	
EBCM10 E-Business	10	
Part 2 (Summer vacation)		
CS-M20 MSc Project	60	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

MSC ADVANCED COMPUTER SCIENCE

WITH SPECIALISATION IN HUMAN-COMPUTER INTERACTION

	credit points	page no.
Compusory Modules		
CS-M00 Research Methodology	10	
CS-M39 Interaction Technologies: Seminars & Reading	10	
CS-M49 Interaction Technologies: lab & field work	10	
CS-M90 Human Computer Interaction Project Development	10	
At least 20 credits from Pathway Modules		
CS-M19 Interactive Systems Design	10	
CS-M29 Mobile Interaction Design	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	
Optional Modules - students must take 120 credits in total		
Semester 1		
CS-M07 Data Visualisation	10	
CS-M13 Critical Systems	10	
CS-M58 Distributed Object-Oriented Programming	10	
CS-M77 Computer Vision and Pattern Recognition	10	
CS-M88 Embedded Systems	10	
Semester 2		
CS-M18 IT Security: Theory and Practice	10	
CS-M61 Concepts of Programming Languages	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M68 Writing Web and Web Service Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M75 Logic for Computer Science	10	
CS-M78 High Performance Computing in C/C++	10	
CS-M84 Software Testing	10	
CS-M85 Modelling and Verification Techniques	10	
CS-M89 Advanced Database Systems	10	
Whole Session		
CS-M67 Graphics Processor Programming	10	
<i>Optional modules may include up to 20 credits from</i>		
MSDM01 Thinking about Digital Media	20	
ASCM17 Criminology/Understanding Crime	20	
GEGM13 Urban Networks and the Knowledge Economy	20	
LALM121 Contract and E-commerce Law	10	
PO-M52 Rethinking Global Politics	20	
EBCM10 E-Business	10	
Part 2 (Summer vacation)		
CS-M20 MSc Project	60	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them

MSC ADVANCED COMPUTER SCIENCE WITH SPECIALISATION IN VISUAL COMPUTING

	credit points	page no.
Compusory Modules		
CS-M00 Research Methodology	10	
CS-M07 Data Visualisation	10	
CS-M70 Visual Computing Project Development	10	
CS-M77 Computer Vision and Pattern Recognition	10	
At least 20 credits from Pathway Modules		
CS-M67 Graphics Processor Programming	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M78 High Performance Computing in C/C++	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	
Optional Modules - students must take 120 credits in total		
Semester 1		
CS-M13 Critical Systems	10	
CS-M58 Distributed Object-Oriented Programming	10	
CS-M88 Embedded Systems	10	
Semester 2		
CS-M18 IT Security: Theory and Practice	10	
CS-M29 Mobile Interaction Design	10	
CS-M61 Concepts of Programming Languages	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M68 Writing Web and Web Service Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M75 Logic for Computer Science	10	
CS-M84 Software Testing	10	
CS-M85 Modelling and Verification Techniques	10	
CS-M89 Advanced Database Systems	10	
Whole Session		
CS-M19 Interactive Systems Design	10	
<i>Optional modules may include up to 20 credits from</i>		
MSDM01 Thinking about Digital Media	20	
ASCM17 Criminology/Understanding Crime	20	
GEGM13 Urban Networks and the Knowledge Economy	20	
LALM121 Contract and E-commerce Law	10	
PO-M52 Rethinking Global Politics	20	
EBCM10 E-Business	10	
Part 2 (Summer vacation)		
CS-M20 MSc Project	60	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

MSC ADVANCED COMPUTER SCIENCE WITH SPECIALISATION IN SAFE AND SECURE SYSTEMS

	credit points	page no.
Compusory Modules		
CS-M00 Research Methodology	10	
CS-M13 Critical Systems	10	
CS-M18 IT Security: Theory and Practice	10	
CS-M50 Safe and Secure Systems Project Development	10	
At least 20 credits from Pathway Modules		
CS-M68 Writing Web and Web Service Applications	10	
CS-M75 Logic for Computer Science	10	
CS-M84 Software Testing	10	
CS-M85 Modelling and Verification Techniques	10	
CS-M88 Embedded Systems	10	
Optional Modules - students must take 120 credits in total		
Semester 1		
CS-M07 Data Visualisation	10	
CS-M58 Distributed Object-Oriented Programming	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M77 Computer Vision and Pattern Recognition	10	
Semester 2		
CS-M29 Mobile Interaction Design	10	
CS-M61 Concepts of Programming Languages	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M78 High Performance Computing in C/C++	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	
CS-M89 Advanced Database Systems	10	
Whole Session		
CS-M19 Interactive Systems Design	10	
CS-M67 Graphics Processor Programming	10	
<i>Optional modules may include up to 20 credits from</i>		
MSDM01 Thinking about Digital Media	20	
ASCM17 Criminology/Understanding Crime	20	
GEGM13 Urban Networks and the Knowledge Economy	20	
LALM121 Contract and E-commerce Law	10	
PO-M52 Rethinking Global Politics	20	
EBCM10 E-Business	10	
Part 2 (Summer vacation)		
CS-M20 MSc Project	60	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

MSC ADVANCED COMPUTER SCIENCE WITH SPECIALISATION IN WEB SCIENCE

	credit points	page no.
Compusory Modules		
CS-M00 Research Methodology	10	
CS-M68 Writing Web and Web Service Applications	10	
CS-M80 Web-Science Project Development	10	
30 - 40 credits from Pathway Modules		
MSDM01 Thinking about Digital Media (20 credits)	10	
GEGM13 Urban Networks and the Knowledge Economy (20 credits)	10	
ASCM17 Criminology/Understanding Crime (20 credits)	10	
LALM121 Contract and E-commerce Law	10	
PO-M52 Rethinking Global Politics (20 credits)	10	
EBCM10 E-Business	10	
Optional Modules - students must take 120 credits in total		
Semester 1		
CS-M07 Data Visualisation	10	
CS-M13 Critical Systems	10	
CS-M58 Distributed Object-Oriented Programming	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M77 Computer Vision and Pattern Recognition	10	
CS-M88 Embedded Systems	10	
Semester 2		
CS-M18 IT Security: Theory and Practice	10	
CS-M29 Mobile Interaction Design	10	
CS-M61 Concepts of Programming Languages	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M75 Logic for Computer Science	10	
CS-M78 High Performance Computing in C/C++	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	
CS-M84 Software Testing	10	
CS-M85 Modelling and Verification Techniques	10	
CS-M89 Advanced Database Systems	10	
Whole Session		
CS-M19 Interactive Systems Design	10	
CS-M67 Graphics Processor Programming	10	
Part 2 (Summer vacation)		
CS-M20 MSc Project	60	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

MSC ADVANCED COMPUTER SCIENCE (OPEN)

	credit points	page no.
Compusory Modules		
CS-M00 Research Methodology	10	
Optional Modules - 60 credits from		
Semester 1		
CS-M07 Data Visualisation	10	
CS-M13 Critical Systems	10	
CS-M58 Distributed Object-Oriented Programming	10	
CS-M69 Interaction Technologies: Information Retrieval	10	
CS-M77 Computer Vision and Pattern Recognition	10	
CS-M88 Embedded Systems	10	
Semester 2		
CS-M18 IT Security: Theory and Practice	10	
CS-M29 Mobile Interaction Design	10	
CS-M61 Concepts of Programming Languages	10	
CS-M65 Artificial Intelligence Applications	10	
CS-M68 Writing Web and Web Service Applications	10	
CS-M71 Design Patterns and Generic Programming	10	
CS-M75 Logic for Computer Science	10	
CS-M78 High Performance Computing in C/C++	10	
CS-M79 Interaction Technologies: Hardware & Devices	10	
CS-M84 Software Testing	10	
CS-M85 Modelling and Verification Techniques	10	
CS-M89 Advanced Database Systems	10	
Whole Session		
CS-M19 Interactive Systems Design	10	
CS-M67 Graphics Processor Programming	10	
<i>Optional modules may include up to 20 credits from</i>		
MSDM01 Thinking about Digital Media	20	
ASCM17 Criminology/Understanding Crime	20	
GEGM13 Urban Networks and the Knowledge Economy	20	
LALM121 Contract and E-commerce Law	10	
PO-M52 Rethinking Global Politics	20	
EBCM10 E-Business	10	
Part 2 (Summer vacation)		
CS-M20 MSc Project	60	

Level M Computer Science optional modules may not run in any year depending upon the number of students enrolled upon them.

UCAS AND SWANSEA UNIVERSITY ROUTE CODES

UCAS Code	programme	UWS Route Code
G400	BSc Computer Science	XCSCS
G401	BSc Computer Science (with foundation year)	FCSCS
G403	MEng Computing	4CSPS
G420	BSc Computing and Communications	XCSOS
G4GC	BSc Mathematics for Computer Science	XMACS
G4R1	BSc Computer Science with a Modern Language (French)	XCSCWFRE
G4R2	BSc Computer Science with a Modern Language (German)	XCSCWGRM
G4R3	BSc Computer Science with a Modern Language (Italian)	XCSCWITA
G4R4	BSc Computer Science with a Modern Language (Spanish)	XCSCWHSS
G4Q5	BSc Computer Science with a Modern Language (Welsh)	T/XCSCWCYM
GG41	BSc Computer Science and Pure Mathematics	XCSCSMAP
GF48	BSc Computer Science and Geoinformatics	XCSCAGEI
CG84	BSc Computer Science and Psychology	XCSCAPSS
FG34	BSc Computer Science and Physics	XCSCAPHY
G4L1	BSc Computing with Finance	XCSHS
R1G4	BA One Modern Language with Computer Studies (French)	SFREWCSS
R2G4	BA One Modern Language with Computer Studies (German)	SGRMWCSS
R3G4	BA One Modern Language with Computer Studies (Italian)	SITAWCSS
R4G4	BA One Modern Language with Computer Studies (Spanish)	SHSSWCSS
Q5G4	BA One Modern Language with Computer Studies (Welsh)	XCYMWCSS
RRD2	BA Two Modern Languages with Computer Studies (French and German)	SFREWGRMWCSS
RRD3	BA Two Modern Languages with Computer Studies (French and Italian)	SFREWITAWCSS
RRD4	BA Two Modern Languages with Computer Studies (French and Spanish)	SFREWHSSWCSS
RQC5	BA Two Modern Languages with Computer Studies (French and Welsh)	SFREWCYMWCSS
RRG3	BA Two Modern Languages with Computer Studies (German and Italian)	SGRMWITAWCSS
RRG4	BA Two Modern Languages with Computer Studies (German and Spanish)	SGRMWHSSWCSS
RQG5	BA Two Modern Languages with Computer Studies (German and Welsh)	SGRMWCYMWCSS
RRJ4	BA Two Modern Languages with Computer Studies Italian and Spanish)	SITAWHSSWCSS
RQH5	BA Two Modern Languages with Computer Studies Italian and Welsh)	SITAWCYMWCSS
RQK5	BA Two Modern Languages with Computer Studies (Spanish and Welsh)	SHSSWCYMWCSS

LEVEL ZERO MODULES

INTRODUCTION TO COMPUTING I

Synopsis:

This module gives an overview of the main topics and questions in Computer Science and enables students who are not majoring in computer science to reach a level of skill in programming such that they will be able to apply their computing knowledge to their other studies. It can also provide (along with other Level 0 modules) a suitable preparation for Level 1 Computer Science.

Syllabus:

General introduction to Computer Science.

Introduction to programming in Java. Program design techniques, simple and structured data types, control structures, object-oriented programming concepts, and applications.

Learning Outcomes:

Students will have gained an insight into the main topics and questions of Computer Science. They will have sufficient programming skills to develop applications for their (non-Computer Science) major. Along with other Level 0 modules they will have gained background knowledge to enable them to proceed to level 1 Computer Science.

Transferable Skills:

Problem solving. Ability to learn and use computer systems and software packages effectively.

Reading:

C. Horstmann, *Java for Everyone*, Wiley, 2010.

J. G. Brookshear, *Computer Science, An Overview*, 9th Ed, Addison-Wesley, 2007.

Code:	CS-061
Credit Points:	10
Taught:	Semester 1
Contact Hours:	30 lectures/laboratory classes
Lecturers:	tbc
Assessment:	100% Continuous Assessment <i>(Laboratory exercises, Open book tests, Coursework)</i>

LEVEL ONE MODULES

COMPUTERS AND COMPUTING

Synopsis:

This module will introduce a variety of contemporary computer platforms. Commencing from the user interface, the module will explain the structure of the underlying operating and storage systems. Students will also explore the various types of networks that underlie the internet.

Syllabus:

What is a computer: Structure; organisation; the ubiquity of computing devices.

Interacting with computers: GUIs and CLIs; specialised interfaces (eg, for mobile phones and digital cameras); new OS interface concepts; protection, management and allocation of resources.

What is an Operating System: Kernels; applications; device drivers; APIs.

Command Line Interfaces: Concepts; Linux shells; text editors; Regular expressions.

File Systems: Files and file types; hierarchical structure; ownership and permissions.

Networking: Protocols; network addresses; DNS; history and development of the internet; WANs, MANs, LANs, SANs and PANs.

Learning Outcomes:

Students will understand the user-level structure of a computer; have an appreciation of the roles of different interface mechanisms; have an understanding of the decomposition of complex computer systems; and appreciate different types of communications networks.

Transferable Skills:

Ability to use a variety of computing platforms.

Reading:

N. Dale and J. Lewis, *Computer Science Illuminated*, 4th Edition, Jones and Bartlett, 2011.

Code:	CS-108
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures,
Lecturers:	Dr. Sharp
Assessment:	30% written and computer based exercises, 70% written examination (in January)

PROGRAMMING 1

Code:	CS-110
Credit Points:	15
Taught:	Semester 1
Contact Hours:	20 lectures, 10 x 2 hour laboratory classes
Lecturers:	Dr. Harman
Assessment:	20% weekly laboratory tasks, 20% programming exercises, 60% written examination (in January)

Synopsis:

The aim of this module is to teach students to solve computational problems by writing simple programs in a high-level language, specifically Java. Students will understand the fundamental principles underlying imperative programming languages, and have the ability and confidence to write programs in Java to solve a variety of simple problems.

Syllabus:

What is a program?

Introduction to Programming in Java.

The Java programming environment.

What is an algorithm?

Declaring and Using Variables and Assignments.

Primitive data types, arithmetical operations, input via keyboard.

Input/Output: Input from Keyboard and Files; Output to Screen and Files.

Decisions: if statements, if-else statement, comparing numbers and strings.

Iteration: for loops, while loops and do loops, nested loops.

Collection Data Types: Arrays and Arraylists.

Common algorithms on arrays. recursion.

Basic error handling.

Program design techniques, modularisation and Methods.

Learning Outcomes:

Students will be able to develop straightforward programs to solve specific problems.

Students will be able to read and debug straightforward programs written by others.

Transferable Skills:

Computational thinking and problem solving.

Course Text:

C. Horstmann, *Java for Everyone*, Wiley, 2010.

Reading:

T. Gaddis, G. Muganda, *Starting out with Java, From Control Structures through Data Structures*, Addison Wesley, 2007.

FROM LANGUAGES TO HARDWARE

Synopsis:

This module introduces the fundamentals of digital design as well as basic structures of computers in a clear and accessible manner to engineering and computer science students. This module builds a bridge between high-level interaction and machine instructions.

Syllabus:

Basic digital logic: This topic provides a clear view about how simple logical building blocks can be used to build computers (bottom up approach).

Structure of computers - Von Neumann organisation, control flow, data and memory and time. In this subject a broader view of structure of computers comes to light, for instance, students will be aware how Boolean gates can become memory units, and how the fetch-decodeexecute cycle works.

Representing data in both high-level and low-level programming (ASCII, integer and floating point).

Machine and assembly language, building a bridge between high-level and low-level programming

Learning Outcomes:

Students will be familiar with the basic, abstract hardware building blocks that are used to construct computers. They will gain a broad understanding about the concept of layers of abstraction, from applications to hardware, and from high-level programming languages to machine languages. They will be aware of the current state of the art in computer hardware.

Transferable Skills:

Decomposing complex systems,

Translating between different representations of a complex system

Reading:

A. Clements, *Principles of Computer Hardware*, 4th Ed, Oxford University Press, 2006.

M. M. Mano, M. D. Ciletti, *Digital Design*, 4th Ed, Prentice Hall, 2007.

F. Vahid, *Digital Design*, 1st Ed, Wiley, 2007.

Code:	CS-113
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. Eslambolchilar
Assessment:	30% coursework, 70% written examination (in May/June)

PROGRAMMING II

Code:	CS-115
Credit Points:	15
Taught:	Semester 2
Contact Hours:	20 lectures, 10 x 1 hour problem sessions, 10 x 1 hour laboratory classes.
Lecturer:	Dr. Xie
Co-requisite:	CS-110
Assessment:	20% fortnightly laboratory tasks, 20% programming exercises, 60% written examination (in May/June)

Synopsis:

This module is a continuation of the module CS-110 Programming 1. In it, students will continue to enhance their skills in programming, as well as gain a basic understanding of algorithms and data structures.

Syllabus:

Objects and Classes.

Instance Methods and Fields.

Static Methods and Fields.

Object oriented programming and design techniques.

Encapsulation: Public and Private Methods, Public and Private Fields.

Basic inheritance: Sub-classes and Overriding.

Introduction to algorithms - searching and sorting.

Simple complexity analysis, introduction to data structures.

Graphical user interfaces.

Learning Outcomes:

Students will be able to develop substantial programs to solve specific problems based on algorithms using standard data structures.

Students will have an awareness of efficiency considerations for different algorithms.

Students will be able to read and debug substantial programs written by others.

Transferable Skills:

Computational thinking and problem solving.

Course Text:

C. Horstmann, *Java for Everyone*, Wiley, 2010.

Reading:

T. Gaddis, G. Muganda, *Starting out with Java, From Control Structures through Data Structures*, Addison Wesley, 2007.

C. Horstmann, *Big Java*, 4th edition, Wiley, 2010.

Further suitable texts on object oriented programming, algorithms and data structures.

PROFESSIONAL ISSUES I: COMPUTERS AND SOCIETY

Synopsis:

Students will be introduced to the main professional issues associated with software development and computer science research. Students will also examine the main impacts of computing on society and social issues on the practice of computing, including legal and ethical concerns such as copyright and the data protection act. Issues of network security will be discussed. The course also includes communication skills, including the writing of a technical report and giving a formal presentation.

Syllabus:

Impact and reach of Computer Science in society: e.g., domains of use and influence, ethical frameworks, codes of conduct, legal constraints, freedom of speech and censorship, privacy and surveillance.

Information security and the importance of encryption.

The impact of these issues on the development and testing of software will be examined. The practice of computing will also be viewed through other disciplines (e.g. Economics) to highlight the practical value of the main course material.

Learning Outcomes:

Students will be aware of major societal, ethical and profession-level issues associated with Computer Science; they will have had experience of writing a technical report and giving a formal presentation on technical material.

Transferable Skills:

Independent discovery of literature, reflection on and critiquing of perspectives, report writing, presentation skills.

Reading:

A.A. Adams and R. McCrindle, *Pandora's Box, Social and Professional Issues of the Information Age*, John Wiley & Sons, 2008.

D.G. Johnson, *Computer Ethics*, 4th Edition, Pearson, 2009.

B. Schneier, *Secrets & Lies: Digital Security in a Networked World*, Wiley, 2004.

Code:	CS-130
Credit Points:	15
Taught:	Semester 1
Contact Hours:	20 lectures, 10x 1 hour case study sessions, 10 x 1 hour tutorial sessions.
Lecturer:	Mr. Whyley
Assessment:	50% synopsis and Report writing and presentation 50% written examination (in January)

PROFESSIONAL ISSUES II: SOFTWARE DEVELOPMENT

Code:	CS-135
Credit Points:	15
Taught:	Semester 2.
Co-requisite:	CS-130.
Contact Hours:	20 lectures, 10 x 2 hour laboratory sessions.
Lecturer:	Dr. Roggenbach
Assessment:	20% weekly lab classes, 20% continuous assessment, 60% written examination (in May/June)

Synopsis:

The aim of this course is to give an understanding of the fundamental software tools, testing and design methods that are used to create reliable software. A number of state-of-the-art development environments will be shown, with hands-on experimentation and use of test systems. Students will also be given a sound grasp of the use of these systems in the different professional software development processes used in the software industry. Innovative software development methods such as Extreme Programming will be introduced and learnt in hands-on laboratory work.

Syllabus:

Introduction to Integrated Development Environments (IDEs).

The Software Development Process.

Software metrics.

Agile Programming/Extreme Programming.

Specifying a program test set.

Program debugging tools and debugging strategies.

Basic unit testing and tools for unit testing.

Learning Outcomes:

An understanding of the methods for developing reliable software. A sound knowledge of current tools and methods for developing and testing software to ensure its reliability and to pinpoint known errors. Students will be able to explain the operation and testing of a simple computer program.

Transferable Skills:

Problem solving. Demonstration skills.

Reading:

I. Sommerville, *Software Engineering*, 9th Edition, Addison-Wesley, 2011.

R. Pressman, *Software Engineering: A Practitioner's Approach* 7th Edition, McGraw Hill, 2009.

P.C. Jurgensen, *Software Testing: A Craftsman's Approach* 3rd Edition, Auerbach, 2008.

CONCEPTS OF COMPUTER SCIENCE I

Synopsis:

This module gives an overview of some of the main principles underlying computers and computing from both a theoretical and an applied point of view. It is accessible and relevant to students of all disciplines who wish to learn about, or reinforce their understanding of, computers and computer science.

Syllabus:

Overview.

The Information Layer.

Binary values and number systems.

Data representation.

The Hardware Layer.

Gates and circuits.

Computing components.

The Programming Layer.

Low-level programming languages and pseudocode.

Learning Outcomes:

Students will gain an appreciation of the scope and limitations of computer science and its applications.

They will be familiar with the principles involved in a number of areas of modern computing.

Transferable Skills:

Computational thinking and problem solving.

Course Text:

N. Dale and J. Lewis, *Computer Science Illuminated*, 4th Edition, Jones and Bartlett, 2011.

Code:	CS-150
Credit Points:	15
Taught:	Semester 1
Contact Hours:	30 lectures, 11 x 1 hour laboratory classes
Lecturer:	Dr. Eslambolchilar
Assessment:	30% continuous assessment, 70% written examination (in January)

CONCEPTS OF COMPUTER SCIENCE II

Code:	CS-155
Credit Points:	15
Taught:	Semester 2
Contact Hours:	30 lectures, 10 x 1 hour problem sessions
Lecturers:	Dr. Sharp
Co-requisite:	CS-150 or CS-113
Assessment:	30% continuous assessment, 70% written examination (in May/June)

Synopsis:

This module follows on from CS-150 and gives an overview of some of the main principles underlying computers and computing from both a theoretical and an applied point of view. It is accessible and relevant to students of all disciplines who wish to learn about, or reinforce their understanding of, computers and computer science.

Syllabus:

The Programming Layer:
Problem Solving and Pseudocode (Ch 7, 3 weeks).
The Operating Systems Layer.
Operating systems (Ch 10, 2 week).
File systems and directories (Ch 11, 1 week).

The Communications Layer.
Networks (Ch 15, 2 week).
World wide web (Ch 16, 2 week).

Learning Outcomes:

Students will gain an appreciation of the scope and limitations of computer science and its applications.

They will be familiar with the principles involved in a number of areas of modern computing.

Transferable Skills:

Computational thinking and problem solving.

Course Text:

N. Dale and J. Lewis, *Computer Science Illuminated*, 4th Edition, Jones and Bartlett, 2011.

MODELLING COMPUTER SYSTEMS I

Synopsis:

This module introduces students to mathematical tools and techniques for modelling computing systems.

Syllabus:

Introduction.
Propositional logic.
Sets.
Boolean algebras and circuits.
Predicate logic.
Proof strategies.
Functions.
Relations.

Code:	CS-170
Credit Points:	15
Taught:	Semester 1
Contact Hours:	30 lectures, 10 x 1 hour problem sessions
Lecturer:	Dr. Beckmann
Assessment:	30% continuous assessment, 70% written examination (i n January)

Learning Outcomes:

Students will become familiar with the fundamental mathematical techniques for modelling hardware and software systems and will develop skills in scientific modelling such as abstraction, the precise formulation of informal notions, rigorous reasoning and analysis.

Transferable Skills:

General mathematical discipline, abstract modelling and formal reasoning, computational thinking.

Reading:

F. Moller, *Modelling Computing Systems*, Swansea University, 2011, to be published by Springer in due course.

K.H. Rosen, *Discrete Mathematics and its Applications*, 6th Edition, McGraw-Hill, 2007.

MODELLING COMPUTER SYSTEMS II

Code:	CS-175
Credit Points:	15
Taught:	Semester 2
Contact Hours:	30 lectures, 10 x 1 hour problem sessions
Co-requisite:	CS-170
Lecturer:	Professor Moller
Assessment:	30% continuous assessment, 70% written examination (in May/June)

Synopsis:

This module will follow on from CS-170 and introduces students to mathematical tools and techniques for modelling computing systems.

Syllabus:

Recursion and inductive definitions.

Proofs by induction.

Games and strategies.

Modelling processes.

The bisimulation games.

Logical properties of processes.

Concurrent processes.

Learning Outcomes:

Students will become familiar with the fundamental mathematical techniques for modelling hardware and software systems and will develop skills in scientific modelling such as abstraction, the precise formulation of informal notions, rigorous reasoning and analysis.

Transferable Skills:

General mathematical discipline, abstract modelling and formal reasoning, computational thinking.

Reading:

F. Moller, *Modelling Computing Systems*, Swansea University, 2011, to be published by Springer in due course.

K.H. Rosen, *Discrete Mathematics and its Applications*, 6th Edition, McGraw-Hill, 2007.

LEVEL ONE AWAYDAY

Synopsis:

Since 2007, level one students and staff of the Computer Science Department have attended an annual half day “Awayday”, at an off-campus venue.

The aim of the Awayday is to provide:

- a reflection of the course so far;
- an opportunity for every student to give feedback to the Department;
- an open discussion between staff and students on education in Computer Science;
- informal conversations and interaction between staff and students.

Syllabus:

The Awayday normally consists of

- a talk by the Head of Department;
- a talk by the Head of Level One;
- a talk by representative current Level Two students;
- a team-exercise;
- a staff/student meeting, at which all aspects of the course experienced so far can be discussed.

Code:	CS-130
Credit Points:	see CS-130
Taught:	Semester 1
Contact Hours:	Half day
Coordinator:	Mr. Whyley
Assessment:	Attendance at the Awayday is considered as part of the coursework for CS-130 and a mark will be awarded accordingly

LEVEL TWO MODULES

DECLARATIVE PROGRAMMING

Synopsis:

This module provides an introduction to the issues raised in developing and using concurrent and distributed systems. Consideration of practical and formal solutions to example problems from operating systems and networking.

Syllabus:

Functional Programming in Haskell.

The functional programming paradigm and its relation to other programming paradigms.

Functions, definitions and types.

Solving simple algorithmic problems using iteration and recursion.

Polymorphism and higher-order functions.

Programming with lists.

Logic Programming in Prolog.

The essence of logic programming.

Pattern matching, recursion, backtracking and resolution.

Database programming.

Extralogical aspects of Prolog.

Data structure terms and lists.

Learning Outcomes:

Students will be able to specify and write programs in functional and logic programming languages. They will be able to develop solutions to simple algorithmic problems using declarative rather than procedural concepts.

Transferable Skills:

Problem solving and abstract modelling.

Reading:

G. Hutton, *Functional Programming in Haskell*, Cambridge University Press, 2006.

P. Blackburn, J. Bos and K. Striegnitz, *Learn Prolog Now!*, College Publications, 2006.

Code:	CS-205
Credit Points:	15
Taught:	Semester 2
Contact Hours:	20 lectures, 10 x 2 hour laboratory sessions
Pre-requisites:	CS-110, CS-115
Lecturer:	tbc
Assessment:	20% coursework, 20% laboratory work, 60% written examination (in May/June) (This module will not run in 2011/2012)

CONCURRENCY

Code:	CS-210
Credit Points:	15
Taught:	Semester 1
Contact Hours:	20 lectures, 10 x 1 hour problem sessions 10 x 1 hour laboratory sessions,
Lecturer:	Dr. Blanck
Pre-requisites:	CS-110, CS-115, CS-170, CS-175
Assessment:	20% lab work, 20% assignment, 60% written examination (in January)

Synopsis

This module provides an introduction to the issues raised in developing and using concurrent and distributed systems. Consideration of practical and formal solutions to example problems from operating systems and networking.

Syllabus:

Background and motivation: concurrent programs and modelling.

Processes and Threads: modelling and implementing processes.

Concurrent execution: modelling concurrency and multi-threaded programs.

Shared objects and mutual exclusion: interference, mutual exclusion in Java.

Monitors and condition synchronization:

semaphores, bounded buffers.

Deadlock and livelock: analysis, dining philosophers.

Safety and liveness properties.

Model-based design: from requirements to models to implementations.

Dynamic systems: master-slave model.

Message passing: synchronous and asynchronous message passing.

Concurrent architectures.

Timed systems.

Learning Outcomes:

Students will have an appreciation of the subtle and complex problems in concurrent systems. They will be aware of strategies to reliably solve these problems. They will be aware of the core algorithms used in concurrent/distributed systems in practice (operating systems, networks – including web applications – and hardware), and will appreciate the link between models of concurrency and their practical application.

Transferable Skills:

Problem solving. Ability to analyse and reason about small but complex program fragments. Ability to relate abstract concepts to concrete examples.

Reading:

J. Magee and J. Kramer, *Concurrency: State Models and Java Programming*, 2nd Edition, Wiley, 2006.

B. Goetz, *Java Concurrency in Practice*, Addison-Wesley, 2006.

LOGIC PROGRAMMING AND ARTIFICIAL INTELLIGENCE

Synopsis

First order logic as a programming language.
Introduction to the fundamentals of Prolog.
Introduction to Artificial Intelligence (AI) and the programming for AI using Prolog.

Syllabus:

Logic Programming: The essence of logic programming and the Horn clause subset.
Pattern matching, recursion, backtracking and resolution. Emphasis will be placed on using Prolog as a practical language for AI.
Database programming. Extralogical aspects of Prolog. Data structure terms and lists.
Artificial Intelligence: The aims and scope of AI.
Basic problem solving.

Heuristics and associated searching techniques. Natural language processing.

Simple examples of language recognition and parsing using definite clause grammars (DCGs).

Learning Outcomes:

- gain an understanding and practical knowledge of the programming language Prolog;
- be able to apply tree techniques for solving puzzles and problems;
- have a knowledge of Definite Clause Grammars (DCG) and be able apply them to parse and recognise simple fragments of English.

Transferable Skills:

Problem solving. Programming in Prolog and constructing simple DCGs.

Course Text:

P. Blackburn, J. Bos and K. Striegnitz, *Learn Prolog Now!*, College Publications, 2006.

Reading:

I. Bratko, *Prolog-Programming for Artificial Intelligence*, 3rd Ed, Pearson, 2000.

Code:	CS-215
Credit Points:	20
Taught:	Semester 2
Contact Hours:	20 lectures, 10 x 1 hour laboratory sessions,
Lecturer:	Dr. Grant, Professor Mosses
Pre-requisites:	CS-110, CS-115, CS-170, CS-175
Assessment:	20% coursework, 80% written examination (in May/June)

COMPUTER GRAPHICS I: IMAGE PROCESSING AND SYNTHESIS

Code: CS-217
Credit Points: 10
Taught: Semester 2
Contact Hours: 20 lectures
Lecturer: Dr. Jones
Pre-requisites: CS-141, or
CS-161 and CS-171

Assessment: 20% coursework,
80% written
examination
(in May/June)

(This module is only
available to students
studying for the BSc
in Computer Science
and Physics)

(This module may not
run in 2011/2012)

Synopsis:

This module gives an introduction to the use of computer graphics and its applications particularly for the production of realistic representations.

Syllabus:

Representing images: Image sampling and quantization for digital images, the human vision system, light intensities, gamma correction and video lookup tables.

Processing images: Colour models, Convolution, Spatial linear filtering, removing noise, template matching (an introduction to computer vision), dithering, colour models, histograms, histogram equalization, anti-aliasing.

Synthesising images: Object representation (3D), ray tracing, ray/sphere intersection, recursive ray tracing, bounding volumes, recursive bounding volumes, kd-tree, octrees, illumination

(ambient, diffuse, specular and the rendering equation).

Visualization and Animation: Marching squares and marching cubes, particle systems, numerical simulation, flocking, genetic algorithms and genetic programming.

Learning Outcomes:

Students will be aware of different forms of computer imagery; methods for synthesising images from data, and the post-processing of images. Students will have experienced programming a graphical application and carrying out operations on a digital image.

Transferable Skills:

Problem solving. Abilities to learn and use computer systems and software packages effectively, and to evaluate and deploy new technologies.

Reading:

D. Hearn and M. P. Baker, *Computer Graphics with OpenGL*, 3rd Ed, Prentice-Hall, 2003.
A. Low, *Introductory Computer Vision and Image Processing*, McGraw-Hill, 1991.

DATABASE SYSTEMS

Synopsis:

This module gives an appreciation of the complexity of real-world databases. It considers some of the problems that can occur in multi-user, multi-transaction situations. It introduces relational databases and covers the design and implementation. It gives students experience in applying these techniques.

Syllabus:

What is a database? What is data? Database software and benefits. ANSI/SPARC model, database structure.

Relational databases — properties, designing, problems. Normalisation — normal forms, functional dependence, primary keys, integrity constraints and rules, validation.

Real world example — SQL and practical sessions using a relational database. Client/server technology, web servers, PHP, ASP, including examples and applications

ER Model — entities, relationships, modelling, attributes, converting to relational model.

Relational calculus and its application to databases, relational algebra — select, project, join, union, intersection, difference, cartesian product, query optimisation.

Recovery and concurrency — transaction processing, locking, detecting deadlocks. Multi-user databases — client/server, distributed, commit protocols.

Learning Outcomes:

Students will be aware of relational databases and the need for the normalisation of data.

Students will have been exposed to transaction processing and how to detect and avoid problems that can arise in a multi-user and/or distributed environment. Students will have designed a database using the ER model, and have practical experience of a relational database.

Transferable Skills:

Problem identification, problem analysis and abstract modelling. Abilities to learn and use computer systems and software packages effectively, and to evaluate and deploy new technologies.

Reading:

C. J. Date, *An Introduction to Database Systems*, 8th Ed, Addison-Wesley, 2004.

J. S. Bowman, S. L. Emerson and M. Darnovsky, *The Practical SQL Handbook*, 4th Ed, Addison-Wesley, 2004.

Code:	CS-219
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Stein
Pre-requisites:	CS-141 or CS-161 and CS-171
Assessment:	20% coursework, 80% written examination (in January) (This module is only available to students studying for the BSc in Computing with Finance) (This module may not run in 2011/2012)

SOFTWARE ENGINEERING I

Code:	CS-230
Credit Points:	15
Taught:	Semester 1
Contact Hours:	20 lectures, 10 x 2 hour laboratory sessions
Lecturer:	Mr. Whyley, Dr. Wilson
Pre-requisites:	CS-110, CS-115, CS-130, CS-135
Co-requisites:	CS-235
Assessment:	20% group work: system evaluation, 20% individual work: system redesign, 60% written examination (in January)

Synopsis:

This module exposes the student to the major components of a practical software lifecycle through team-based practical software engineering. This module introduces students to the concepts of human computer interaction (HCI), and the methodologies for designing and evaluating interactive systems. Students are introduced to the issues and techniques of team working.

Syllabus:

Part A: General Software Engineering and Project Management:

Conventional Software Process and Software Life-cycle models.

Software project management and team organization.

Risk assessment and management.

Cost estimation, commercial and economic

factors.

Software design and evaluation (taught in conjunction with Part B).

Requirements analysis, system design, software development, and testing, re-engineering.

Part B: Human-Computer Interaction (HCI) and Interactive System Design and Evaluation:

Interaction requirements vs. software requirements.

Hierarchical task analysis (HTA), scenarios and personas.

Graphical user interface (GUI) and WIMP interaction styles.

Visual perception and colours.

Memory and Attention.

GUI designing and prototyping.

Evaluation methodologies and techniques.

User studies.

GUI implementation.

Learning Outcomes:

Students will gain:

- An understanding of the principles of software engineering.
- An understanding of the key HCI concepts in the context of system evaluation and design.
- The ability to design and evaluate GUIs.

Transferable Skills:

Problem solving through analysis and abstract reasoning.

The ability to read critically, to précis and judge information.

Exposure to team work, time management, project management, and risk assessment.

Skills in written communication and documentation.

Reading:

H.V. Vliet, *Software Engineering, Principles and Practice*, 3rd Edition, Wiley, 2008.

A. Dix, J. Finlay, G. D. Abowd and R. Beale, *Human Computer Interaction*, Prentice Hall, 2003.

SOFTWARE ENGINEERING II

Code:	CS-235
Credit Points:	15
Taught:	Semester 2
Contact Hours:	15 lectures, 25 team surgery laboratory sessions
Lecturer:	Dr. Laramee
Co-requisites:	CS-230
Pre-requisites:	CS-110, CS-115, CS-130, CS-135
Assessment:	20% group work: OO software construction: phase 1 (then handover), 20% group work: OO software construction: phase 2 (takeover), 20% Individual work: software re-engineering, 40% written examination (in May/June)

Synopsis:

Building on the programming knowledge and experience acquired by the students in the co - and pre-requisites, this module enables students to apply advanced software development methodologies, with a focus on object-oriented software design and engineering. The module provides students with valuable opportunities to acquire knowledge and experience in important aspects of software engineering with a strong emphasis on teamwork and practical assignments. A significant element of the module is the construction of a substantial software product, where students are assessed on their effectiveness at working together as a team.

Syllabus:

Part A: Object-Oriented Software Engineering:

Requirement analysis and specification.

Object oriented software architecture design.

GUI design and development.

Event-driven programming in an object-oriented programming environment.

Object orientation: inheritance, abstraction, encapsulation, polymorphism.

Modelling with classes.

Using design patterns.

Object-oriented design principles.

Object-oriented software development and reusability in software engineering.

The use of object orientation in software re-engineering.

Part B: Tools and Techniques for Software Engineering.

Collaborative tools - Version Control Systems, Team-based IDEs.

Advanced Software Engineering Methodologies - RAD, Scrum, Extreme Programming.

Software maintenance.

Learning Outcomes:

Students will gain:

- An understanding of Object-oriented programming concepts, and knowledge of their applications in software design and engineering processes,
- The ability to build GUIs and skills of event-driven programming,
- Experience and appreciation of group work, skills of project management, risk assessment and cost estimation, and awareness of commercial and economic factors.

Transferable Skills:

Problem solving through analysis and abstract reasoning.

Experience and appreciation of team work, time management, project management, and risk assessment.

Skills in written communication and documentation.

The ability to learn and use computer systems and software packages effectively.

Reading:

H.V. Vliet, *Software Engineering, Principles and Practice* (3rd Edition), Wiley, 2008.

A. Dix, J. Finlay, G. D. Abowd and R. Beale, *Human Computer Interaction*, Prentice Hall, 2003.

LANGUAGE AND COMPUTATION

Code:	CS-236
Credit Points:	20
Taught:	Semester 2
Contact Hours:	30 lectures, 10 hours tutorial, 10 hours lab sessions
Lecturers:	Dr. Setzer, Professor Tucker, Mr. Whyley
Assessment:	30% coursework, 70% written examination (in May/June)

Synopsis:

This module introduces the principles for defining the syntax and semantics of programming languages. Program labs will provide practical experience in writing compilers. The students will learn about the limits of computation.

Syllabus:

Introduction to the syntax and semantics of programming languages. Grammars for defining syntax. BNF, the Chomsky hierarchy and the language recognition problem. Finite automata, pushdown automata, context free grammars. Parse trees, concrete and abstract. Abstract data types, portability and information hiding. Interfaces and

signatures. Implementations and algebras. Specifying data types. Comparing implementations using homomorphisms. Mathematical models of imperative programs-states, traces. Comparing implementations using homomorphisms. Turing analysis of computation. Register machines, assembly languages, virtual machines. The Turing halting problem. Tools for parsing syntax. Token recognition, error checking and recovery, symbol tables, tree generation and transformation.

Learning Outcomes:

Students will understand basic techniques for defining the syntax and semantics of programming languages. They will have a practical understanding of the compilation process. They will be aware of the notions of portability, specification and verification and how these notions are supported by programming language techniques. They will have learned how to specify, define and implement abstract data types. They will know techniques for compiling simple languages. They will be aware of the limits of computation.

Transferable Skills:

Ability to construct abstract models. Ability to use tools for software construction. Ability to reason abstractly. Ability to understand systems in an appropriate way within the limits of computation.

Reading:

N. J. Cutland, *Computability: An introduction to recursive function theory*, Cambridge University Press, 1980.

K. C. Loudon, *Compiler Construction: Principles and Practice*, Addison-Wesley, 1997.

T. A. Sudkamp, *Languages and Machines*, 3rd Ed, 2005.

J. V. Tucker and K. Stephenson, *Data, Syntax and Semantics*, Course Notes, Dept. of Computer Science, Swansea University, 2006.

ALGORITHMS

Synopsis:

This module introduces various techniques for the design and analysis of algorithms through practical problem solving.

Syllabus:

Algorithm design techniques: divide-and-conquer (min-max, mergesort, closest points); greedy algorithms (making change, minimum spanning trees, data compression); dynamic programming (making change, sequence alignment, shortest paths).

Algorithm analysis techniques: pseudocode for describing algorithms; big-o notation; solving recurrences; NP-completeness.

Data structures: graph representations and algorithms (breadth-first and depth-first search; topological sorting and strongly-connected components); binary tree representations and algorithms (insertion and deletion; heapsort and priority queues).

Example application domains: eg, network flow; string matching; computational geometry; disjoint-set data structures.

Learning Outcomes:

The ability to formalise and analyse problems; to present algorithmic solutions to such problems based on standard techniques; and to analyse the correctness and efficiency of such solutions.

Transferable Skills:

Problem solving through analysis and abstract reasoning.

Reading:

T. Cormen, C. Leiserson, R. Rivest and C. Stein, *Introduction to Algorithms*, 2nd Ed, 2001.

Code:	CS-242
Credit Points:	20
Taught:	Semester 1
Contact Hours:	30 lectures, 10 problem classes, 10 laboratory sessions
Lecturer:	Dr. Beckmann, Dr. Kullmann
Pre-requisites:	CS-141, CS-142
Assessment:	30% coursework, 70% written examination (in January)
	<i>(This module is only available to students studying for the BSc in Computer Science and Physics)</i>
	<i>(This module may not run in 2011/2012)</i>

DATABASE SYSTEMS

Code:	CS-250
Credit Points:	15
Taught:	Semester 1
Contact Hours:	20 lectures, 10 x 1 hour problem/tutorial sessions
Lecturer:	Dr. Stein
Pre-requisites:	CS-141 or CS-161 and CS-171
Assessment:	30% coursework, 70% written examination (in January)

Synopsis:

This module will discuss the theory, design and implementation of databases.

Syllabus:

What is a database? What is data? Database software and benefits. ANSI/SPARC model, database structure.

Relational databases — properties, designing, problems. Normalisation — normal forms, functional dependence, primary keys, integrity constraints and rules, validation.

Real world examples — SQL and practical sessions using a relational database. Client/server technology, web technology e.g. PHP, ASP, Ruby on Rails, including examples and applications.

ER Model — entities, relationships, modelling, attributes, converting to relational model.

Relational calculus and its application to databases, relational algebra — select, project, join, union, intersection, difference, cartesian product, query optimisation.

Recovery and concurrency — transaction processing, locking, detecting deadlocks. Multi-user databases — client/server, distributed, commit protocols.

Security — managing users and passwords, encryption, securing a database (e.g. MySQL).

Learning Outcomes:

Students will be aware of relational databases and the need for the normalisation of data. Students will have been exposed to transaction processing and how to detect and avoid problems that can arise in a multi-user and/or distributed environment. Students will have designed a database using the ER model, and have practical experience of a relational database.

Transferable Skills:

Problem identification, problem analysis and abstract modelling. Abilities to learn and use computer systems and software packages effectively, and to evaluate and deploy new technologies.

Reading:

C. J. Date, *An Introduction to Database Systems*, 8th Edition, Addison-Wesley, 2003.

A. Beaulieu, *Learning SQL*, 2nd Edition, O'Reilly, 2009.

COMPUTER GRAPHICS

Synopsis:

This module will provide an introduction to the use of computer graphics and its applications particularly for image processing and the production of realistic representations.

Syllabus:

Fundamentals: Image sampling and quantization. Digital images. Storage and pixels. Perception, human visual system, Gamma correction. Mathematical background.

Image Processing: Representation — sizing, re-scaling, rotation, colour components, brightness and colour models, histograms, histogram equalization, nearest neighbour, bilinear and tricubic interpolation. Processing techniques — JPEG compression, quantization, antialiasing, filtering, convolution, dithering, edge detection and denoising.

Image Synthesis: Ray tracing — modelling scenes, accelerating ray tracing using bounding volumes and octrees, fundamental primitives, lighting and illumination, shadows, reflections and transparency.

Applications: Volume data and rendering — isosurfacing, volume rendering, Maximum Intensity Projection.

Animation Techniques: Key-frame animation, particle systems, flocking, genetic algorithms, and a summary of other techniques.

Learning Outcomes:

Students will be aware of different forms of computer imagery; methods for synthesizing images from data; and the post-processing of images. Students will have experienced programming a graphical application and carrying out operations on a digital image.

Transferable Skills:

Problem solving. Abilities to learn and use computer systems and software packages effectively, and to evaluate and deploy new technologies.

Reading:

D. D. Hearn, M. P. Baker and W. Carithers, *Computer Graphics with OpenGL*, 4th Edition, Prentice-Hall, 2010.

A. Low, *Introductory Computer Vision and Image Processing*, McGraw-Hill, 1991.

Code:	CS-255
Credit Points:	15
Taught:	Semester 2
Contact Hours:	30 lectures, 10 x 1 hour problem sessions
Pre-requisites:	CS-110, CS-115
Lecturers:	Dr. Jones
Assessment:	30% continuous assessment, 70% written examination (in May/June)

ALGORITHMS

Code:	CS-270
Credit Points:	15
Taught:	Semester 1
Contact Hours:	30 lectures, 10 x 1 hour problem sessions
Lecturers:	Dr. Kullmann
Assessment:	30% continuous assessment, 70% written examination (in January)

Synopsis:

This module introduces students to the formal concepts of algorithms and data structures and will enable them to understand how the selection of different algorithms and data types affects the performance and efficiency of a program. Particular attention will be paid to the fundamental problems of searching, sorting, and graph traversal.

Syllabus:

Introduction to the concept of algorithm and program efficiency.

Mathematical foundations: asymptotic notation,

summations, recurrence relations.

Introduction to various abstract data types: stacks, queues, lists, heaps and tables.

Searching algorithms: binary search trees, balanced search trees, hash tables.

Sorting algorithms: merge sort, quick sort heap sort.

Graph representations and algorithms: adjacency lists and matrices, depth-first and breadth-first search.

Learning Outcomes:

Students will appreciate the idea of analysing an algorithm to determine its efficiency.

Students will be familiar with, and be able to manipulate, basic abstract specifications of some standard data types.

Students will know and understand various standard sorting and searching algorithms and be able to comment on their relative performance.

Students will be familiar with directed and undirected graphs, in particular their various representations, and be able to solve algorithmic problems based on depth- and/or breadth-first search.

Transferable Skills:

General mathematical discipline, abstract modelling and formal reasoning, computational thinking.

Reading:

T. Cormen, C. Leiserson, R. Rivest and C. Stein, *Introduction to Algorithms* 3rd Edition, MIT Press, 2009.

AUTOMATA AND FORMAL LANGUAGE THEORY

Synopsis:

This module introduces the principles for defining the syntax of programming languages. The students will learn about the major abstract language classes underpinning computation as well as about the limits of computation.

Syllabus:

Introduction to the syntax of programming languages: grammars for defining syntax. BNF, the Chomsky hierarchy and the language recognition (parsing) problem.

Finite-state automata, regular languages and regular expression: equivalences between formalisms, methods for determining when a language is or is not regular.

Push-down automata, Context-free languages and context-free grammars: equivalences between formalisms, methods for determining when a language is or is not context-free.

Turing machines, recursive and recursively-enumerable languages: equivalences between formalisms, methods for determining when a language is or is not recursive.

Learning Outcomes:

Students will understand basic techniques for defining the syntax of programming languages and have an understanding of the compilation process. They will be aware of the limits of computation and be familiar with the standard hierarchy of formal language classes as well as the classes of machines which they model.

Transferable Skills:

Ability to construct abstract models and to understand systems within the limits of computation; abstract modelling and formal reasoning; computational thinking.

Reading:

P. Linz, *An Introduction to Formal Languages and Automata*, 4th Edition, Cambridge University Press, 2006.

H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation* 2nd Edition, Prentice Hall, 1998.

Code:	CS-275
Credit Points:	15
Taught:	Semester 2
Contact Hours:	30 lectures, 10 x 1 hour problem sessions
Lecturer:	tbc
Assessment:	30% continuous assessment, 70% written examination (in May/June) (This module will not run in 2011/2012)

LEVEL TWO AWAYDAY

Taught:	Semester 2
Contact Hours:	Whole day
Coordinator:	Mr. Whyley
Assessment:	<i>There is no assessment for this event, but attendance is compulsory for all students who will be studying CS-344 and CS-354 (project) in their third year.</i>

Synopsis:

Since 2009, level two students and staff of the Computer Science Department have attended an annual “Awayday” at an off-campus venue.

The aim of the Awayday is to provide:

- a reflection of the course so far;
- an opportunity for every student to give feedback to the Department;
- an open discussion between staff and students on education in Computer Science;
- informal conversations and interaction between staff and students.

- information about project work and studying in level 3.
- the opportunity for students to make their project selection after discussion with potential supervisors.
- the opportunity for members of lecturing staff who have not lectured at levels one and two to introduce themselves and their research interests.
- the opportunity for potential project supervisors to explain their project ideas.

Syllabus:

The Awayday normally consists of

- a talk by the Head of Department;
- a talk by the Head of Level Two;
- a talk by representative current Level Three students;
- a team-exercise;
- a staff/student meeting, at which all aspects of the course experienced so far can be discussed.
- talks by “new” members of lecturing and research staff.
- a project selection event.

LEVEL THREE MODULES

COMPUTER GRAPHICS II: MODELLING AND RENDERING

Synopsis:

This module builds upon the material presented in module CS-217, and focuses on the techniques and algorithms for modelling and rendering three dimensional graphical objects. There is also a look at computer animation techniques.

Syllabus:

3D Geometrical Transformations: Translation, scaling, rotation, composition, reflection, shear, coordinate system transformation.

3D Viewing: View specification and viewing transformation. Parallel and perspective projections. View volume.

Illumination and Shading: Type of light sources; ambient, diffuse, specular reflection; constant, Gouraud and Phong shading.

Geometric Modelling: Boundary representations, parametric curves and surfaces, sweep representations, spatial-partitioning representations, constructive solid geometry.

Hidden Surface Removal Algorithms: Depth-sort, z-buffer, back face culling and BSP trees.

Further Study on Ray-Tracing: The basic recursive algorithm, simple intersection algorithms, efficiency considerations and quality considerations.

Other Rendering Methods: Photon ray tracing, radiosity, shadow, texture mapping.

Volumetric Modelling: Surface extraction, implicit surfaces, volume data types, quadtrees and octrees, direct volume rendering, applications.

Learning Outcomes:

Students will acquire an understanding of the main concepts of graphical modelling and rendering, and knowledge of a range of advanced techniques.

Transferable Skills:

Problem solving. Ability to evaluate and deploy new technologies. Information retrieval, and ability to manage learning processes.

Course Text:

D. Hearn and M. P. Baker, *Computer Graphics with OpenGL*, 3rd Ed, Prentice Hall, 2003.

Reading:

A. Watt, *3D Computer Graphics*, 3rd Ed, Addison-Wesley, 2000.

H. W. Jensen, *Realistic Image Synthesis Using Photon Mapping*, AK Peters, 2001.

P. Shirley and R. K. Morley, *Realistic Ray Tracing*, AL Peters, 2003.

E. Angle, *Interactive Computer Graphics*, 3rd Ed, Addison-Wesley, 2003.

J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes, *Computer Graphics: Principles and Practice*, 2nd Ed, Addison-Wesley, 1990.

POV-Ray Documentation, <http://www.povray.org/documentation/>

Code:	CS-307
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Jones
Assessment:	20% coursework, 80% written examination (in January)

CONCEPTS OF PROGRAMMING LANGUAGES

Code:	CS-311
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Incompatible:	CS-M61
Lecturer:	Prof. Mosses
Assessment:	20% coursework, 80% written examination (in May/June) Familiarity with a functional programming language such as Haskell or ML, as well as with an object-oriented one such as C# or Java, is assumed

Synopsis:

This module introduces the student to the concepts underlying the concrete constructs of programming languages supporting a certain style of programming (paradigm), like imperative, object-oriented, concurrent, functional, logic and visual programming.

Syllabus:

This course introduces principles of programming languages from a practical viewpoint. To this end, we first study and describe general concepts of programming languages, including data types, expressions, commands, declarations, and abstractions. We then analyse how these concepts are related to the concrete design of a programming language that supports a certain style of programming (paradigm) as e.g. imperative, object-oriented, concurrent, functional, logic and visual programming.

Throughout the course, Ada, C, C++, Java, and Haskell are used as reference languages. Simple exercises help understanding of the subject in the way of learning by doing.

Learning Outcomes:

Students will understand the concepts underlying the concrete constructs of a programming language and what characterises the different programming paradigms.

This knowledge will lead them to an understanding of the programming languages they are working with, making it easier to learn new programming languages, and allow them to judge the design of a programming language as well as to select an appropriate language to solve a certain problem.

Transferable Skills:

Ability to learn and use computer systems and packages effectively.

Course Text:

P. D. Mosses, *Course Notes on Concepts of Programming Languages*, Dept of Computer Science, Swansea University, 2011.

Reading:

D. A. Watt, *Programming Language Design Concepts*, Wiley, 2004.

HIGH INTEGRITY SYSTEMS

Synopsis:

The module enables students to develop an appreciation of the problems of developing high-integrity systems, together with practical experience of applying modern, formal techniques to the production and verification of such software.

Syllabus:

Introduction and Motivation: What are high integrity and critical systems? Legal and ethical issues. Examples of major failures of high integrity systems. Successes and how/why they worked. Standards for safety-critical software and their shortcomings.

Analysis: The hazard analysis process. Safety analysis and the safety case. Safety issues related to, but outside software. Human factors: the role of poor interfaces in software failures.

Specification and Verification: Languages and tools for formal specification. Tools for automated and machine-assisted theorem proving.

Software Production: Issues in program language selection to minimise failure. The software engineering process in the production of high-integrity software.

Correctness: Validation and verification: the advantages and disadvantages of testing and formal verification.

Learning Outcomes:

Students will be familiar with issues surrounding high-integrity systems, including legal and ethical issues and hazard analysis.

They will understand techniques for specifying and verifying high-integrity software.

They will have experience with using tools for developing critical systems.

Transferable Skills:

Problem solving, and especially abstract modelling and formal reasoning. Information retrieval, ability to read critically, to précis and judge information, and ability to manage learning processes.

Course Texts:

N. Storey, *Safety-Critical Computer Systems*, Addison-Wesley, 2nd Ed, 1996.

Reading:

J. G. P. Barnes, *High Integrity Software: The SPARK approach to safety and security*, Addison-Wesley, 2003.

N. Leveson, *Safeware: Systems Safety and Computers*, Addison-Wesley, 1995.

C. Geffroy and G. Motet, *Design of Dependable Computing Systems*, Springer, 2011.

Code:	CS-313
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Setzer
Incompatibles:	CS-M13
Assessment:	20% coursework, 80% written examination (in January)

CRYPTOGRAPHY AND IT SECURITY

Code:	CS-318
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. T. Chen (Engineering)
Prerequisites:	CS-238
Incompatible:	CS-M18
Assessment:	20% coursework, 80% written examination (in May/June)

Synopsis:

The aim of this course is to examine theoretical and practical aspects of computer and network security.

Syllabus:

Introduction: threats and their causes.

Security engineering: security criteria; security models.

Cryptography: basic encryption & decryption; cryptanalysis; symmetric cryptosystems (e.g. DES, AES, RC4); asymmetric cryptosystems (e.g. RSA, Diffie-Hellman key exchange, ElGamal);

cryptographic hash functions & digital signatures; key management; authentication concepts; access control.

Tools & technologies: IPsec; tunneling & VPNs; TLS, SSL, SSH and related tools; PGP and GPG; security in OpenBSD.

Vulnerabilities and attacks: port scanning; packet sniffing; buffer overflows; SQL injection.

Security issues in wireless networks.

Learning Outcomes:

Students will be aware of the main security issues in today's IT infrastructures. They will have good knowledge of the current techniques for increasing IT security, and awareness of their limits. They will understand the models and methods used to systematically construct secure systems or enhance the security of existing systems.

Transferable Skills:

Problem analysis and solving, abstract modelling, formal reasoning. Ability to manage learning process. General mathematical discipline. Ability to learn and use computer systems effectively.

Course Texts:

W. Stallings, *Cryptography and Network Security*, 3rd Ed, Prentice Hall, 2003.

B. Schneier, *Applied Cryptography*, 2nd Ed, Wiley, 1996.

Reading:

M. Bishop, *Introduction to Computer Security*, Addison Wesley, 2005.

N. Ferguson and B. Schneier, *Practical Cryptography*, Wiley, 2003.

E. Skoudis and T. Liston, *Counter Hack Reloaded*, 2nd Ed, Prentice Hall, 2006.

J. Edney and W. Arbaugh, *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*, Addison-Wesley, 2004.

DATA VISUALISATION

Synopsis:

Data Visualization is concerned with the automatic or semi-automatic generation of digital images that depict data in a meaningful way(s). It is a relatively new field of computer science that is rapidly evolving and expanding. It is also very application oriented, i.e., real tools are built in order to help scientists from other disciplines.

Syllabus:

We will start off by introducing the fundamentals of visualization. Introductory topics include purposes and goals of visualization, applications, challenges, the visualization pipeline, sources of data, data dimensionality, data types, and grid types.

The next sub-topic examines information visualization, that is, visual representations of abstract data. Information visualization topics include hierarchical data, tree maps, cone trees, focus and context techniques, graphs and graph layouts, multi-dimensional data, scatter plots, scatter plot matrices, icons, parallel coordinates, interaction techniques, linking and brushing.

The second major sub-topic is the study of volumetric data. Volume visualization topics include slicing, surface vs. volume rendering, transfer functions, interpolation schemes, direct volume visualization, ray casting, image order vs. object order algorithms, gradients filtering, interpolation, and isosurfacing.

The third major sub-topic is vector field visualization. Topics include simulation, measured, and analytical data, steady and time-dependent (unsteady) flow, direct and indirect flow visualization, applications, hedge hog plots, vector glyphs, numerical integration schemes, streamlines, streamline placement, geometric flow visualization techniques, texture-based techniques, and feature-based flow visualization.

Learning Outcomes:

Students will gain competence in the field of data visualization. They will understand the basic methods available for the computer-aided depiction of data from several inter-disciplinary and application oriented sources. They will also gain and understanding of the visualization problems that have been solved as well as the challenges that remain. Students will also obtain a heightened awareness of implementation challenges associated with data visualization.

Transferable Skills:

The ability to identify and generate advanced visualizations of data, comparative analysis, the ability to identify sources of data and the challenges when visualizing data as well as the challenges that scientists and practitioners from other disciplines face.

Reading:

M. Ward, G. Grinstein, and D. Keim, *Interactive Data Visualization: Foundations, Techniques, and Applications*, A.K. Peters, 2010.

A. Telea, *Data Visualization: Principles and Practice*, A.K. Peters, 2008.

Additional reading materials will be distributed during lectures.

Code:	CS-337
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Laramee
Corequisites:	CS-307
Prerequisites:	CS-217
Incompatible:	CS-M07
Assessment:	20% Coursework, 80% written examination (in January)

INTERNET COMPUTING

Code:	CS-338
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Harman
Incompatible:	CS-M58
Assessment:	30% coursework, 70% written examination (in January)

Synopsis:

The module will introduce a range of mainly Java-based technologies in current use for developing distributed systems over the internet. Practical experience of building internet-based systems will be gained via coursework.

Syllabus:

Socket programming in Java.

Distributed programming - the object oriented approach using remote objects.

Java RMI: synchronous and asynchronous systems - polling and callbacks

Web services: XML based distributed programming models, SOAP, WSDL, ReST

Learning Outcomes:

Students will become familiar with current and near-future internet-based distributed computing.

They will understand the technical and other advantages and disadvantages of such systems.

They will be thoroughly familiar with methodological good practice in the development of internet-based distributed systems.

They will have gained practical competence in a subset of such systems, via coursework.

Transferable Skills:

Students will practice and develop their abilities to solve technical problems, in a methodologically-structured framework.

Students will develop their self-study skills, while learning a number of technical skills.

Reading:

M. Kalin, *Java Web Services: Up and Running*, O'Reilly, 2009

W. Grosso, *Java RMI*, O'Reilly, 2002.

The following web site contains substantial tutorial information:

<http://www.oracle.com/us/java/index.html>

The course notes (supplied) contain numerous links to sources of additional information.

PROJECT IMPLEMENTATION AND DISSERTATION

Synopsis:

The aims of this module are:

- to provide students with the opportunity of exploring a particular topic in computer science in some considerable depth;
- to provide the opportunity of specifying, designing and implementing a complete system and experiencing the major phases of the life-cycle of an IT project;
- to enhance students' competence in system design, algorithm analysis and mathematical reasoning, and their fluency in using programming languages and tools;
- to give students an intellectual challenge to their abilities to learn new subjects without instruction, and to further develop their abilities in literature searching, report writing, verbal presentation, project planning and time management.

Code:	CS-344
Credit Points:	20
Taught:	Whole Session
Contact Hours:	Regular contact with supervisor
Co-ordinators:	Mr. Whyley, Dr. Roggenbach
Co-requisite:	CS-334
Assessment:	Continuous assessment <i>This module is only available to students who are majoring in Computer Science</i>

Description:

Dissertation.

Deadline: Semester 2, Wednesday May ???th 10am.

The dissertation (40-100 pages) is a comprehensive and self-contained report on the work done on the project (see CS-354). For internal students the module CS-354 must have been completed before a dissertation will be accepted. External students must viva / demonstrate their project before it will be considered by the Department. The document should address the following topics:

- discussion of the subject area and its history;
- a study and survey of relevant literature and similar work;
- formulation of scientific questions and the answers to them;
- theoretical background and mathematical prerequisites;
- technical problems considered and methods used to solve them;
- discussion of issues arising in specifying, designing, implementing and testing the system (e.g. requirements analysis, user interface, system architecture, algorithms, major data structures, etc.);
- evaluation of results (e.g. complexity, efficiency, user-friendliness, reliability, etc.);
- user and system manual;
- progress and achievements of the project;
- suggestions for further work.

This list should be understood as a suggestion rather than a checklist.

Submission:

Students should submit two copies of the dissertation to Room 206 and submit an electronic copy in either **.pdf** or **Microsoft Word** format to the Blackboard turnitin site. One of the hard copies **must** include a cd containing full source code of any software written for the project.

Learning Outcomes:

Students will have explored a particular topic in computer science in some considerable depth. They will have specified, designed and implemented a complete system and experienced the major phases of the life-cycle of an IT project. Their competence in system design, algorithm analysis and mathematical reasoning will have been enhanced, and their fluency in using programming languages and tools improved. They will have had the intellectual challenge to their abilities of learning new subjects without instruction, and further developed their abilities in literature searching, report writing, oral presentation, project planning and time management.

Transferable Skills:

Written communication and documentation, and oral presentation. Time management and project management. Information retrieval, ability to read critically, to précis and judge information, and ability to manage learning processes. Problem solving.

Note: The British Computer Society (BCS) also stipulates additional constraints on projects for students who wish their degree to gain the exemption from the BCS professional Examinations. See Page ??? for details. If you are unsure as to whether your project meets the requirements please discuss it with your supervisor.

This module is not available to students doing a joint honour scheme or whose minor subject is Computer Science.

ARTIFICIAL INTELLIGENCE APPLICATIONS

Synopsis:

This module introduces the student to the fundamentals of knowledge based systems, presents the basics of expert system shells such as AILog, and presents the fundamentals of fuzzy expert systems. It also introduces the student to aspects of soft computing such as genetic algorithms and genetic programming.

Syllabus:

Expert systems. Early development including MYCIN. The basic architecture — working memory, inference engine, user interface. Rule-based systems. Frame based systems. Methods of inference. Forward and backward chaining. Resolution proof method. Inexact reasoning and uncertainty. Certainty factors and probability. Non-monotonic proof techniques. Default and temporal reasoning. Explanation — how and why. Knowledge acquisition.

Fuzzy logic. Fuzzy expert systems.

Case based reasoning. Expert system shells (AILog).

Evolutionary computing — genetic algorithms, genetic programming.

Learning Outcomes:

Students will acquire the basics of the expert system shell AILog and be able to produce an expert system using this tool, a deep understanding of the concepts of evolutionary programming and be able to solve problems using this paradigm, the thorough knowledge of fuzzy logic and the ability to produce rules for a fuzzy controller.

Transferable Skills:

Problem solving. Ability to evaluate and deploy new technologies.

Reading:

D. Poole and A. Mackworth, *Artificial Intelligence, Foundations of Computational Agents*, CUP, 2010.

M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1996.

J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

J. R. Koza, *Genetic Programming I and II, The Video*, UVC videos.

B. Kosko, *Fuzzy Thinking*, Harper Collins, 1984.

J. L. Kolodner, *Case Based Reasoning*, Morgan Kauffman, 1993.

L. N. de Castro, *Fundamentals of Natural Computing - Basic Concepts, Algorithms and Applications*, Chapman and Hall, 2006.

Code:	CS-345
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Grant
Incompatible:	CS-M65
Assessment:	10% coursework, 90% written examination (in January)

BUILDING RELIABLE WEB APPLICATIONS

Code:	CS-348
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Mr. Whyley
Incompatible:	CS-M68
Assessment:	30% continuous assessment, 70% written examination (in May/June)

Synopsis:

The module will introduce the principles and technologies used for building web-based systems using the .NET Framework. Practical experience of building web systems will be gained via coursework.

Syllabus:

Early Client and Server-side Web Technologies: CGI, ISAPI, ASP and scripting languages.

Introduction to the .NET Framework: CLI, CLR, ASP.NET.

The C# Programming Language.

Principles: security, scalability, performance, data integrity.

.NET Visual Development Tools

Practical Issues: Security in ASP.NET, scalability (application state), performance ASP.NET.

XML, SOAP, WSDL and UDDI.

Building and Using Web Services.

Hybrid client/server based technologies, AJAX

Learning Outcomes:

Students will become familiar with state-of-the-art web programming principles and technologies.

They will understand the technical and other advantages and disadvantages of such technologies.

They will be thoroughly familiar with methodological good practice in the development of web systems.

They will have gained practical competence in a subset of such systems, via coursework.

Transferable Skills:

Problem solving; using advanced software packages for programming; self-study.

Reading:

D. Esposito, *Programming Microsoft ASP.NET 3.5*, Microsoft Press, 2008.

S. Walther, *ASP.NET 3.5 Unleashed*, SAMS, 2008.

MOBILE INTERACTION DESIGN

Synopsis:

This module presents key human computer interaction design issues, methods, tool and techniques in a mobile and ubiquitous systems context. Students will learn how to improve the user interfaces they design and be equipped to develop efficient, effective and satisfying applications for an important, emerging class of computing device. The module will involve students in prototyping and evaluating mobile applications and introduce embedded application development environments.

Code:	CS-349
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Prof. Jones
Incompatible:	CS-M29
Assessment:	30% coursework, 70% written examination (in May/June)

Syllabus:

User interface technologies and components;

Characteristics of effective user interfaces;

Human-factors design and development methodologies;

User-centred requirements gathering & analysis;

Prototyping & evaluation;

Mobile & Embedded programming;

Design issues and strategies for (example list): accessing complex functions (menus, modes etc); mobile web browsing and searching; rich media access; and, mobile communities.

Learning Outcomes:

Students will gain a thorough understanding of the human-factor issues relating to mobile and ubiquitous computing systems.

Students will have a deep knowledge of user-centred software design tools, models and methods.

Transferable Skills:

Course encourages "design" based thinking; the high degree of analytical, reflective and critical skills needed for user-centred interactive system development are widely applicable in many other contexts.

Course Text:

M. Jones and G. Marsden, *Mobile Interaction Design*, John Wiley & Sons, 2006.

PROJECT SPECIFICATION AND DEVELOPMENT

Code:	CS-354
Credit Points:	20
Taught:	Whole Session
Contact Hours:	Regular contact with supervisor
Co-ordinators:	Mr. Whyley, Dr. Roggenbach
Co-requisite:	CS-344
Assessment:	Continuous assessment <i>This module is only available to students who are majoring in Computer Science</i>

Synopsis:

The aims of this module are:

- to provide students with the opportunity of exploring a particular topic in computer science in some considerable depth;
- to provide the opportunity of specifying, designing and implementing a complete system and experiencing the major phases of the life-cycle of a computing project;
- to enhance students' competence in system design, algorithm analysis and mathematical reasoning, and their fluency in using programming languages and tools;
- to give students an intellectual challenge to their abilities to learn new subjects without instruction, and to further develop their abilities in literature searching, report writing, verbal presentation, project planning and time management.

Description:

Final year projects may range from the production of a substantial, high-quality piece of software with little "experimental" content, to entirely theoretical studies of some aspect of computer science. In practice, both extremes are unusual as most projects involve a substantial amount of software, perhaps of an "exploratory" nature, together with some theoretical aspects. The Department produces an annual list of proposed projects, and students should approach members of staff for detailed information on those projects that interest them or suggestions for alternatives. Each student will be supervised by a member of staff. This is a project preliminaries module that involves a number of milestones including the production of three documents, a short public presentation, a formal review meeting and a demo/viva meeting. **Students should submit two copies of each document to room 206, and submit an electronic copy in either .pdf or Microsoft Word format to the Blackboard turnitin site.** The supervisor is to be consulted on the precise contents of each document.

Milestones:

Initial Project Document.

Length: 15-20 pages. Proportion of mark: 35%. Deadline: Semester1, Week 4, Monday 10am.

This document should give the title and introduction to the project area. It should detail the scientific and technical background of the project, and present a rigorous discussion of the project. For example, it should contain a design specification of the entire system to be developed or discussion of theoretical problems to be addressed. The document should cover the following topics: *(This list should be understood as a suggestion rather than a checklist.)*

- a study and survey of relevant literature and similar work;
- a detailed project plan;
- a complete discussion of the background and the relation of the project to this;
- the main methods and tools to be employed or evaluated;
- the main scientific questions to be considered;
- the main technical problems to be solved;
- the software and hardware constraints if appropriate;
- anticipated problems and further areas of study or influence.

Public Presentation.

Duration: 10 minutes. Proportion of mark: 20%. Tuesday 22nd November 2011.

The presentation of the aims and background of the project, and the progress to date, will be given to an audience of about 20 students and staff. It will be held at the Department's Annual Undergraduate Computer Science Colloquium at Gregynog (21st - 23rd November 2011).

Project Review.

During the last week of Michaelmas Term.

An unassessed, formal meeting with the student's supervisor to review the project and the progress in detail. In particular, it is intended to identify problems with the project work and to correct possible errors of judgement before the Christmas vacation.

Interim Document.

Length: 10-15 pages. Proportion of mark 25%. Deadline: Semester 2, Week 3, Monday 10am.

This document should summarise progress to date, report preliminary results and state changes to the initial plan if any. The document may include the following topics:

- a progress review;
- a further literature survey;
- preliminary results (e.g. problems mathematically formulated, solutions found, algorithms designed and parts of the system formally specified or implemented);
- a revised project plan and timetable.

This list should be understood as a suggestion rather than a checklist.

Demonstration.

Proportion of mark: 20%. Semester 2, ??? May ???th 2012.

The Department will organise a Project Demonstration Fair, to which will be invited contacts from industry and students from other levels. All lecturing staff will attend at various times. Students will produce a poster and will be expected to explain and demonstrate their project whenever a large enough audience is gathered. Students will be assessed on the quality of their poster and demonstration.

Learning Outcomes:

Students will have explored a particular topic in computer science in some considerable depth. They will have specified, designed and implemented a complete system and experienced the major phases of the life-cycle of a computing project. Their competence in system design, algorithm analysis and mathematical reasoning will have been enhanced, and their fluency in using programming languages and tools improved. They will have had the intellectual challenge to their abilities of learning new subjects without instruction, and further developed their abilities in literature searching, report writing, verbal presentation, project planning and time management.

Transferable Skills:

Written communication and documentation, oral presentation, and interactive discussions. Time management, and project management. Problem solving. Information retrieval, ability to read critically, to précis and judge information, and ability to manage learning processes.

Note: The British Computer Society (BCS) also stipulates additional constraints on projects for students who wish their degree to gain the exemption from the BCS professional Examinations. See Page ??? for details. Students who are unsure as to whether their project meets the requirements should discuss it with their supervisor.

HIGH-PERFORMANCE COMPUTING IN C/C++

Code:	CS-358
Credit Points:	10
Taught:	Semester 2 (tbc)
Contact Hours:	20 lectures
Lecturer:	Dr. Borgo
Assessment:	100% coursework

Synopsis:

The module will provide C/C++ fundamentals to students. An overview of High Performance Computing through the C/C++ languages will also be given.

Syllabus:

Introduction to C/C++ ,Fundamentals/close to hardware, UNIX Systems, Differences from Java

highlighted, h and .cpp, Compilers and compiling.

More on the grammar: Structures, sizeof, #define, if-else, case, and other weird operators, loops, comma operator, booleans, enum, Scope of a variable, static variables

Basic Classes in C++: private, protected, public, Use of Constructors and Destructors, Operator overloading (Complex class), Pointers (C programming style) and arrays, with emphasize on software engineering and memory models, Memory Allocation, malloc, free, new[], delete[], Memory Leaks, Classes as a clean solution to memory leak, Multidimensional arrays, linear formula: Strings (safe/unsafe), Argument passing, Pointer to functions

C/C++ differences: Malloc vs new, String vs char *, printf,read vs cin, classes vs struct. When using object technology?

Inheritance: Multiple inheritance, Virtual functions, dynamic binding

Templates: Function and class templates, Issues, Example: sort function, STL, Iterators

High Performance Computing (or how your program will run 20x faster): floating point representations, Types of parallelisms, What C allows, Compilers and mental model, Processor pipeline, branching instructions, memory access, math instructions, simple types, size and alignments, SSE instruction programming, Multi-threaded programming/ OpenMP, Memory communications, Barriers/MPI/pragma

System designs: More on the software engineering side, Pitfall, More on Real-time environments, normalized compilers

Learning Outcomes:

The module will provide two main learning outcomes:

-The basic concepts behind C/C++ programming languages, two very much used programming languages by the IT industry, will be learnt.

-The know-how of applying C/C++ programming to High-Performance Computing (HPC), including knowledge on both HPC architectures and parallel algorithms.

Transferable Skills:

.Problem solving. Programming on parallel systems. Ability to evaluate and deploy new technologies.

Reading:

B. Stroustrup, *The C++ Programming Language: Special Edition*.

S. Prata, *C++ Primer Plus*, 5th Ed.

SOFTWARE TESTING

Synopsis:

Testing is the process of systematically experimenting with an object (the SUT = System Under Test) in order to establish its quality, where quality means the degree of accordance to the intention or specification. This module will cover various test scenarios; practical exercises will allow the students to gain hands-on experience with various testing tools.

Syllabus:

The module provides a profound overview on industrially relevant methods in software testing and points out current research directions.

- Functional Testing: *Boundary Value Testing, Equivalence Class Testing, Decision Table-Based Testing.*
- Structural Testing: *Path Testing, Data Flow Testing.*
- Integration and System Testing: *Levels of Testing, Approaches to Integration Testing, Threads for System Testing, Interaction Testing.*
- Object-Oriented Testing: *Issues, Class Testing, Object-Oriented Integration Testing.*
- Selected Research Topics: *e.g. Testing Hybrid Systems.*

Learning Outcomes:

Thorough understanding of testing as a method to validate software systems; critically evaluate and select software test scenarios; problem analysis.

Transferable Skills:

Analysis of systems; systematic design of experiments.

Reading:

P. C. Jorgensen, *Software Testing*, CRC Press.

Code:	CS-364
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. Roggenbach
Assessment:	30% coursework, 70% written examination (in May/June)

EMBEDDED SYSTEMS

Code:	CS-368
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 laboratory classes, 10 lectures
Lecturer:	Dr. Beckmann
Assessment:	10% course work, 15% team laboratory exercises, 35% individual lab book, 40% written examination (in January)

Synopsis:

Embedded systems are information processing systems embedded into enclosing products such as cars, telecommunication or fabrication equipment. They are essential for providing ubiquitous information, one of the key goals of modern information technology.

The aim of this course is to discuss selected topics of embedded system design.

The lab provides hands-on experience in the design of embedded systems, based on the Lego-Mindstorms kit.

Awareness of logical concepts (propositional logic, first order logic) will help the understanding of this module.

Syllabus:

The lectures discuss selected techniques in their specialisation to the design of embedded systems such as:

- Requirements, Specification and Modeling
- Programming-language-level description techniques.
- Hardware (Sensors, actuators, processors)
- Operating systems, middleware, scheduling
- Hardware/software partitioning and codesign
- Simulation, testing and verification techniques

The labs consist of a series of experiments using the Lego Mindstorms Kit that give the students hands-on experience in typical design challenges in embedded systems. Possible topics include examples from

- control theory
- real time systems
- discrete control
- fault tolerance
- distributed algorithms.

Learning Outcomes:

Students will understand and be able to apply engineering principles for system design and their specific application in embedded systems. They will be able to cope with various methods for specification/modelling, analysis, design, implementation and verification.

Transferable Skills:

Understanding of computer programs as part of socio-technical systems; problem solving; demonstration, presentation and writing skills; team-working.

Reading:

P. Marwedel, *Embedded System Design*, Springer, 2006.

DESIGN PATTERNS AND GENERIC PROGRAMMING

Synopsis:

A study of generic programming and selected design patterns and idioms, using Java as the teaching language. Students will learn the techniques for creating highly efficient and highly reusable libraries.

Syllabus:

Generic programming has the goal of expressing algorithms and data structures in the most general way, while maintaining the same efficiency as hand-crafted specialised code. Object-orientation in its traditional form (Java for example) realises polymorphism only via so-called sub-class polymorphism, while generic programming extends these techniques by the use of parametric polymorphism, realised in C++ by Generics (together with other means for polymorphism like overloading of classes and global functions). This module introduces the basic mechanisms of generic programming (using Java as the teaching language) with emphasis on design patterns and idioms.

Introduction : basics; erasure; reification; upgrading of libraries.

Selected design patterns, especially policy-based design and visitors.

Learning Outcomes:

Students will have an understanding of the ideas and techniques of generic programming, especially a thorough awareness of how to use the ideas and techniques to write efficient and useful libraries. They will acquire a sound knowledge of the main design patterns and idioms.

Transferable Skills:

Abstract modelling and implementation of software via generic techniques. Problem solving. Ability to evaluate and deploy new technologies.

Reading:

M. Naftalin and P. Wadler, *Java Generics and Collections*, O'Reilly, 2007.

E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns. Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.

E. Freeman and E. Freeman, *Head First Design Patterns*, O'Reilly, 2004.

Code:	CS-371
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures/seminars
Incompatible:	CS-M71
Lecturer:	Mr. Whyley
Assessment:	20% continuous assessment, 80% written examination (in May/June)

LOGIC FOR COMPUTER SCIENCE

Code: CS-375	Synopsis: This module provides an introduction to logic and its applications to computer science, in particular to the formal specification and verification of computer programs. Syllabus: <ul style="list-style-type: none">• Propositional Logic; syntax, semantics, proof system.• Predicate Logic; syntax, semantics, proof system.
Credit Points: 10	
Taught: Semester 2	
Contact Hours: 20 lectures	
Lecturer: Dr. Berger	
Assessment: 30% coursework, 70% written examination (in May/June)	

- Applications of logic to program specification and verification.
- Specialised Logics e.g. for security protocols, reactive systems and credit card systems.

Learning Outcomes:

At the end of this module students will understand the syntax, semantics and proof rules of first-order predicate logic; be aware of other logics that serve special purpose in computer science, (e.g. modal logic, process logic), understand the importance of logic for computer science, be able to express informal statements as formal predicate logic and carry out simple formal proofs.

Transferable Skills:

Problem solving, logical thinking, assessing the validity and invalidity of logical arguments.

Reading:

D. van Dalen, *Logic and Structure*, Springer, 1994

D. B. Plummer, J. Barwise and J. Etchemendy, *Tarski's World*, revised and expanded edition, CSLI Lecture Notes, 2008

COMPUTER VISION AND PATTERN RECOGNITION

Synopsis:

This module introduces students to the important and modern topics and concepts of computer vision and pattern recognition, including image processing, segmentation, feature extraction, camera calibration, stereo vision, motion analysis, object tracking, recognition, data clustering, and dimensionality reduction. It teaches techniques that are used to understand and interpret the contents of images and videos and dissects state-of-the-art vision systems, such as Microsoft Kinect.

Code:	CS-377
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Xie
Incompatible:	CS-M77
Assessment:	10% coursework, 90% written examination (in January)

Syllabus:

This course is composed of four parts: Introduction, Image Processing, Video Analysis and Pattern Recognition & Applications.

Introduction: The first part of the lectures gives an overview of Computer Vision and Pattern Recognition (CVPR) and a road show of this course. It also provides a brief revision of basic and important mathematical techniques frequently used in CVPR.

Image processing: filtering, Hough transform, registration, object extraction, shape recognition, segmentation, texture analysis.

Video analysis: camera models and calibration, stereo vision, depth estimation, motion estimation and tracking, local features for tracking.

Pattern Recognition: data clustering and K-means, Gaussian Mixture Modelling, dimensionality reduction, and applications.

Learning Outcomes:

Students will become familiar with the important concepts of computer vision and pattern recognition, acquire a knowledge of how the analysis of digital images and videos may be performed, and develop the skills necessary to program a basic computer vision system.

Transferable Skills:

Basic problem solving. Ability to learn and use computer systems and software package. Ability to evaluate new technologies. General mathematical analysis.

Reading:

D. Forsyth and J. Ponce, *Computer Vision: a Modern Approach*, Prentice Hall 2003

C. Bishop, *Pattern Recognition and Machine Learning*, Springer 2006.

NEW SCREEN TECHNOLOGIES

Code:	MS-306
Credit Points:	20
Taught:	Semester 2
Contact Hours:	10 lectures
Lecturer:	Dr. Berry (Media & Communication)
Assessment:	100% 3250 word essay

Synopsis:

This course introduces and develops students critical understandings of new media technologies. It seeks to extend important theoretical approaches and concepts for analysing new media and digital technology, providing a critical framework for research and practice in new media. This course draws from a wide body of theoretical literature including philosophy, sociology, political science, media studies and medium theory.

Syllabus:

Introduction

Understanding technology

The information society

Code: the poetics of cyberspace

Digital texts: the internet

Digital friends: the social turn

Digital play: games and technology

Digital space: mobility and ubiquity

Digital touch: haptic technology

Essay Surgery

Learning Outcomes:

An understanding of key theoretical directions in new media..

Knowledge and appreciation of key texts in the field of digital media.

An awareness of the problems and different theoretical approaches to studying new media and media generally

Transferable Skills:

Students should have developed rigorous organisational skills through the management of personal study.

They should have developed theoretical and methodological skills in the field of new media theory

Reading:

Y. Benkler, *The Wealth of Networks*, Yale University Press, 2006.

D. M. Berry, *Copy, Rip, Burn: The Politics of Copyleft and Open Source*, Pluto Press, 2008.

M. Castells, *The Information Society: The Rise of the Network Society*, Blackwells, 2000.

F. Kittler, *Gramophone, Film, Typewriter*, Stanford University Press, 1999.

H. Jenkins, *Convergence Culture*, New York University Press, 2006.

L. Manovich, *The Language of New Media*, The MIT Press, 2002.

DIGITAL PHILOSOPHY: THE ROOTS OF THE VIRTUAL

Synopsis:

This module critically explores the intellectual and philosophical history of contemporary digital technology, digital media and digital culture.

Syllabus:

The module will critically explore the intellectual and philosophical history of contemporary digital technologies, digital media and digital culture. Its syllabus will be drawn from:

Antecedents of virtuality in western philosophy: idealist and materialist epistemologies (Plato, Gnosticism, Descartes, Hume)

Historical ideas about the relationship between the organic and mechanical and man and machine; the roots of cyborg theory

20th century philosophies of technology

Cybernetics and information theory

Human-computer interaction and ideas of personal tool use

Electronic media, unity and community

The counterculture and computers; the video activist movement

Games, Game worlds, fantasy and Dungeons and Dragons

New media theory; key texts of the internet

Science fiction and the virtual

Ideas about linked or collective knowledge and intelligence

Transhuman futures

Digital critics

Learning Outcomes:

On completion of this module students should be able to:

- Demonstrate a critical understanding of the historical and intellectual roots of digital technology
- Demonstrate a critical awareness of the issues raised by digital technology
- Demonstrate an understanding of the different theoretical approaches towards technology and the political, social and cultural implications of their positions
- Critically analyse philosophical arguments concerning digital technologies and their effects

Transferable Skills:

At the end of this module, students should:

Have developed organizational skills through the management of personal research and working towards essay deadlines

Have developed communication skills through essay preparation and writing.

Have the ability to apply imaginative and critical thought in connection with the study of digital technologies and their history.

Code:	MS-353
Credit Points:	20
Taught:	Semester 1
Contact Hours:	10 lectures
Lecturer:	Mr. Merrin (Media & Communication)
Assessment:	100% 3500 word essay

Have the ability to form independent views and to express, examine and defend them effectively in written form.

Have the ability to appraise and evaluate philosophical theories, ideas and arguments.

Reading:

S. Perkowitz, *Digital People. From Bionic Humans to Androids*, Joseph Henry Press, 2004.

E. Davis, *Techgnosis*, Serpent's Tail, 1998.

W. Gibson, *Neuromancer*, Harper Collins [1984], 1995.

F. Turner, *From Counterculture to Cyberculture*, University of Chicago Press, 2006

D. Channell, *The Vital Machine. A Study of Technology and Organic Life*, Oxford University Press, 1991.

U. Apollonio, (ed.) *Futurist Manifestos*, Tate Publishing, 2009.

N. Weiner, *Cybernetics*, MIT Press [1948], 1961.

J. C. Licklider, *Man-Computer Symbiosis*, 1960, available at:

<http://groups.csail.mit.edu/medg/people/psz/Licklider.html>.

J. C. R. Licklider and R. Taylor, *The Computer as Communication Device*, 1968 available at:

<http://www.kurzweilai.net/meme/frame.html?main=/articles/art0353.html>.

J. Illich, *Tools For Conviviality*, Harper and Row, 1973.

M. McLuhan, *Understanding Media*, Routledge [1964], 1994.

M. Shamberg, *Guerilla Television*, Holt Reinhart and Winston, 1971.

C. Leadbetter, *We Think: The Power of Mass Creativity*, Profile Books, 2008.

C. Shirky, *Here Comes Everybody*, Allen Lane, 2008.

P. Levy, *Collective Intelligence*, Perseus Books, 1997.

R. Kurzweil, *The Singularity is Near*, Gerald Duckworth and Co. Ltd, 2005.

N. K. Hayles, *How We Became Posthuman*, The University of Chicago Press, 1999.

'KC' (1995) *Industrial Society and Its Future ('the Unabomber Manifesto')*, available at:

<http://cyber.eserver.org/unabom.txt>.

THE CLASSICAL TRADITION IN THE SCIENCES

Synopsis:

While developments in science and scientific medicine have played a key part in the shaping of the modern world, the contrast between twenty-first century knowledge and the knowledge of our ancestors can make it easy to overlook continuities in the study of nature over the centuries and the rationality of other forms of the scientific enterprise. So too can the image of science as, in some sense, an apolitical activity be divorced from its social and cultural settings. This module will study scientific institutions, theories, and methods in Antiquity and demonstrate how these - along with the reasons for studying nature - both shaped and were shaped by society and culture.

Code:	HIP300
Credit Points:	10
Taught:	Semester 1
Contact Hours:	10 lectures plus 2 seminars
Lecturer:	Dr. Rihll (History and Classics)
Assessment:	Online test 50%, 1500 word essay 50%

Syllabus:

1. Introducing ancient science
2. Pythagoras and Pythagoreanism
3. The first atomic theory and its critics
4. Hippocratic and other medicines
5. Training and education in classical societies
6. Mathematics for work and play in Euclid & Archimedes
7. Ordering the Heavens: Ptolemy's Astronomy and Astrology
8. Mapping the Earth: Ptolemy's Geography
9. Public knowledge: Roman encyclopaedias and other handbooks
10. Slender and broken threads: the survival and loss of ancient scientific ideas

Learning Outcomes:

At the end of this module, students should be able to:

- demonstrate sound factual knowledge of the institutions, theories, methods and objectives of ancient science
- critically evaluate the status of science as an enterprise shaped by, and shaping, ancient society and culture

Transferable Skills:

During the course of this module, students should have developed their ability to:

- interpret and analyse textual and non-textual material
- formulate and present arguments both orally and in writing
- plan and pursue independent reading and research

Reading:

T. E. Rihll, *Greek Science: New Surveys in the Classics*, OUP for the Classical Association, 1999.

FROM NATURAL PHILOSOPHY TO SCIENCE

Code:	HIP301
Credit Points:	10
Taught:	Semester 1
Contact Hours:	10 lectures plus 2 seminars
Lecturer:	Dr. Mosely (<i>History and Classics</i>)
Assessment:	100% written examination (January)

Synopsis:

While developments in science and scientific medicine have played a key part in the shaping the modern world, the contrast between twenty-first century knowledge and the knowledge of our ancestors can make it easy to overlook continuities in the study of nature over the centuries. So too can the image of science as, in some sense, an apolitical enterprise divorced from its social and cultural settings. This module will consider the development of the scientific enterprise from the fifteenth century to the twentieth and, by studying

the evolution of scientific institutions, theories, and methods show how these - as well as the reasons for studying nature - have changed over time and have both shaped and been shaped by society and culture. As part of the attempt to understand the significance of the changing scientific enterprise to the history of the modern world, it will address the question of what constitutes 'modern science' and consider debates about when it came into being.

Syllabus:

1. Classical traditions transformed: a new world and a new universe
2. The anatomical Renaissance: Vesalius, Harvey, and the body
3. The occult sciences: alchemy, astrology, natural magic
4. Experimental and mechanical philosophies: Bacon, Descartes, and the Royal Society
5. Isaac Newton and the spectrum of Newtonianism
6. The End of Natural Philosophy: the French Revolution and the sciences
7. British Science in the Nineteenth Century: Babbage, the politics of science and professionalisation
8. Twentieth-century transformations: Relativity, Quantum Mechanics and Evolutionary Biology
9. Science and war: the Manhattan Project, radar research and Bletchley Park
10. Post-war trends: big science, biology and computing

Learning Outcomes:

At the end of this module, students should be able to:

- demonstrate sound factual knowledge of the changing institutions, theories, methods and objectives of early modern and modern science
- critically evaluate competing claims about the origins of 'modern' science
- critically evaluate the changing status of science as an enterprise shaped by, and shaping, society and culture

Transferable Skills:

During the course of this module, students should have developed their ability to:

- interpret and analyse textual and non-textual material
- formulate and present arguments both orally and in writing
- plan and pursue independent reading and research

Reading:

P. J. Bowler and I. R. Morus, *Making Modern Science: A Historical Survey*, University of Chicago Press, 2005

UNDERGRADUATE COMPUTER SCIENCE COLLOQUIUM

Synopsis:

Since 1985, students and staff of the Computer Science Department have attended an annual 2-3 day Undergraduate Computer Science Colloquium at Gregynog, the University of Wales Conference Centre near Newtown, Powys.

The aim of the colloquium is to provide:

- an overview of Computer Science, its research and applications, its history and future developments;
- an opportunity for every student to give a presentation about their project;
- an open discussion between staff and students on the education in Computer Science;
- a reflection on career prospects for computer science graduates;
- informal conversations between staff and students.

Syllabus:

The Colloquium normally consists of:

- presentation by all students of their work on their projects (assessed under CS-354);
- invited lectures on different aspects of Computer Science, usually including research, historical, commercial and social topics;
- a review and discussion of the current state and future development of Computer Science education at Swansea;
- advanced education, employment and career development.

Learning Outcomes:

Students will have given formal presentations to a medium-sized professional audience. They will have reflected on (i) the current state and future of their subject, (ii) their education and training, and (iii) on their personal development and professional prospects.

Transferable Skills:

Oral presentation, and interactive discussions. Ability to read critically, to précis and judge information.

Taught:	<i>Semester 1</i>
Credit Points:	<i>see CS-354</i>
Contact Hours:	<i>2-3 days of seminars, presentations, meetings and social events</i>
Coordinator:	<i>Dr. Roggenbach</i>

PROJECT DEMONSTRATION FAIR

Taught:	Semester 2
Credit Points:	see CS-354
Contact Hours:	Half a day for the fair plus a lecture on poster design
Organisers:	Dr. Roggenbach
Assessment:	During the fair the student will be visited by his/her project supervisor and second marker who will mark the demonstration/display as part of the assessment for CS-354

Synopsis:

Since 2008, final year students present the results of their project to the departmental public as well as to the local IT industry. This event is organised in cooperation with IT Wales. In the past it has received generous funding from the Welsh Assembly. The aim of the fair is to provide the students with a professional fair environment, where they:

- learn how to present and how to defend their projects,
- see how their own project fits into the context of the whole year's results,
- get into contact with IT companies.

Syllabus:

The fair lasts about 4 hours. During this time, the students have to be available at their assigned exhibition space for discussions with the visitors. Depending on the interest of the visitor, such a discussion can last between 1 and 10 minutes. The students are also encouraged to visit the exhibition space of their fellows.

Learning Outcomes:

The students will learn how to present their results in a concise but exciting way. The fair format will force them to concentrate on the essentials of their work.

Transferable Skills:

Oral presentation and interactive discussion. Ability to present a focused project in a larger context.

LEVEL M MODULES

RESEARCH METHODOLOGY

Synopsis:

This module consists in lectures and seminars about fundamental research methodologies and good practice in research, formulation of research questions and hypotheses, logical reasoning, literature research, proper acknowledgement of sources, principles in carrying out experimental research.

Syllabus:

Lectures about fundamental research methodologies and good practice in research, formulation of research questions and hypotheses, logical reasoning, literature research, proper acknowledgement of sources, principles in carrying out experimental research.

Seminars about selected scientific texts.

Learning Outcomes:

Students will have gained an understanding of fundamental research methodologies and good practice in research. These include the formulation of research questions and hypotheses, techniques of valid and convincing argumentation, literature research, proper acknowledgement of sources, extraction of information from literature, project planning.

They will be conversant with the principles of carrying out experimental research.

Transferable Skills:

Valid argumentation, literature research, project planning.

Course Texts:

Recommended texts may vary

Code:	CS-M00
Credit Points:	10
Taught:	Semester 1
Contact Hours:	Lectures and seminars
Co-ordinators:	tbc
Assessment:	Continuous assessment

(This module is only available to students doing a Computer Science MSc and is compulsory for all such schemes)

GROUP PROJECT

Code:	CS-M04
Credit Points:	20
Taught:	Whole session
Contact Hours:	Time in tutorials
Co-ordinators:	Dr. Eslambolichar
Assessment:	Continuous assessment

Synopsis:

Students on this module will specify, develop, test and document a substantial software system in groups of (usually) three or four.

Syllabus:

Students will specify, develop, test and document a substantial software system under the supervision of an academic staff member.

Significant emphasis will be placed on delivery - that is, meeting stated project goals. Somewhat less emphasis will be placed on ambitious technological solutions.

Milestone 1: End of term 1. Deliverables: Team Structure and Methodology Document describing the roles of the team members, the methodology they will use and a risk analysis, including steps taken to 'design out' risks with a relatively high likelihood and serious consequences (approx 10 pages; 10%); Requirements Document describing the requirements of the system in a precise and structured form (typically 10-20 pages but depends on project; 10%); Specification Document describing the specification of the behaviour (and if relevant other aspects of the project - e.g. performance) in a precise and structured form (typically 10-20 pages but depends on project; 10%).

Milestone 2: End of term 2. Deliverables: Interim Report summarising in a narrative form the progress made on the project at this point (typically 10-20 pages but depends on the project; 10%).

Milestone 3: Semester 2, Week 10. Deliverables: Poster session describing the project and its status, and a demonstration of the software (10%); User Manual (length variable; 10%); Design Document describing the design of the system, preferably using a (semi-)formal notation such as UML, and including a rationale for design decisions made (length variable; 10%); Testing Document describing in a structured way the tests applied and their outcome (length variable; 10%); and Narrative and Reflective Account describing the team's experiences, problems encountered and solved (or not) and the team's reflections on the successful and less successful aspects of the project, including how well the team was able to work together (typically 10-20 pages; 20%).

It is normally expected that all students participating will receive the same mark. However, if the students or co-ordinator feel this will be unfair, a system based on each students' ranking of their own and the other team members' work will be used. It is generally expected that students will work in teams of three or four.

Learning Outcomes:

Students will gain experience of: extended team working; project planning; resource estimation; risk analysis; software development; document writing; and system and document versioning.

Transferable Skills:

Team working, planning, management, document writing.

Course Texts:

Texts specific to particular projects will be suggested as required.

ADVANCED TOPICS IN LOGIC AND COMPUTATION

Synopsis:

This module provides a broad overview of current research. Students will write reports and give talks on general research topics in theoretical computer science.

Syllabus:

The course provides a broad overview of current research. A series of seminars will be given by leading researchers from within and outside the department. Under the guidance of the course co-ordinator students will write reports and give talks on general research topics in theoretical computer science. Reports and talks are assessed by at least two lecturers.

Learning Outcomes:

Students will have a clear picture of current research in Theoretical Computer Science and will have in-depth knowledge about selected topics in the field of logic and computation.

Students will have learned to do independent literature research and to write down their findings in a report.

Transferable Skills:

Problem structuring and solving, literature research, report writing.

Course Texts:

Recommended texts may vary. Suggested reading lists will be given for the various topics covered in the seminars given by academic staff, visitors and students.

Code:	CS-M05
Credit Points:	10
Taught:	Semester 1
Contact Hours:	Seminars
Co-ordinators:	Dr. Kullmann
Assessment:	Seminar presentation and report
	<i>This module is only available to students doing the MRes in Logic and Computation</i>

DATA VISUALISATION

Code:	CS-M07
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Laramée
Assessment:	50% continuous assessment, 50% written examination (in January)

Synopsis:

Data Visualization is concerned with the automatic or semi-automatic generation of digital images that depict data in a meaningful way(s). It is a relatively new field of computer science that is rapidly evolving and expanding. It is also very application oriented, i.e., real tools are built in order to help scientists from other disciplines.

Syllabus:

We will start off by introducing the fundamentals of visualization. Introductory topics include purposes and goals of visualization, applications, challenges, the visualization pipeline, sources of data, data dimensionality, data types, and grid types. The next sub-topic examines information visualization, that is, visual representations of abstract data. Information visualization topics include hierarchical data, tree maps, cone trees, focus and context techniques, graphs and graph layouts, multi-dimensional data, scatter plots, scatter plot matrices, icons, parallel coordinates, interaction techniques, linking and brushing.

The second major sub-topic is the study of volumetric data. Volume visualization topics include slicing, surface vs. volume rendering, transfer functions, interpolation schemes, direct volume visualization, ray casting, image order vs. object order algorithms, gradients filtering, interpolation, and isosurfacing.

The third major sub-topic is vector field visualization. Topics include simulation, measured, and analytical data, steady and time-dependent (unsteady) flow, direct and indirect flow visualization, applications, hedge hog plots, vector glyphs, numerical integration schemes, streamlines, streamline placement, geometric flow visualization techniques, texture-based techniques, and feature-based flow visualization.

Learning Outcomes:

Students will gain competence in the field of data visualization. They will understand the basic methods available for the computer-aided depiction of data from several inter-disciplinary and application oriented sources. They will also gain and understanding of the visualization problems that have been solved as well as the challenges that remain. Students will also obtain a heightened awareness of implementation challenges associated with data visualization.

Transferable Skills:

The ability to identify and generate advanced visualizations of data, comparative analysis, the ability to identify sources of data and the challenges when visualizing data as well as the challenges that scientists and practitioners from other disciplines face.

Reading:

M. Ward, G. Grinstein, and D. Keim, *Interactive Data Visualization: Foundations, Techniques, and Applications*, A.K. Peters, 2010.

A. Telea, *Data Visualization: Principles and Practice*, A.K. Peters, 2008.

Additional reading materials will be distributed during lectures.

FUTURE INTERACTION TECHNOLOGIES: MSc PROJECT

Synopsis:

The FIT MSc project provides students with the opportunity to explore a particular topic in interaction technologies in depth. The project will typically involve the development of an interactive system in which case the student will experience the major phases of the life-cycle of practical IT-project: Specification, design, implementation, testing, verification and evaluation. The project will enhance the students' competence in all aspects of design, including algorithm analysis, informal and formal mathematical reasoning, experiments and evaluation. It will give students an intellectual challenge to their abilities to learn new subjects without instruction and to further develop their abilities in literature researching, report writing, verbal presentation, project planning and time management.

Code:	CS-M08
Credit Points:	60
Taught:	Semester 2
Contact Hours:	Regular contact with supervisor
Coordinator:	Prof. M Jones
Assessment:	100% continuous assessment
	Only available as part of MSc in Computing & Future Interaction Technologies

Syllabus:

Students will be expected to undertake an in-depth project into a topic in interaction technologies under the supervision of an academic member of staff. Students write a dissertation on their project. A log book will be kept by the student, which may also be assessed. The dissertation will be assessed according to the University of Wales Standing Orders.

Learning Outcomes:

Students will have carried out a substantial, usually practical, project in the area. They will have documented their work in a dissertation that describes background, aims, methods and results of their work and critically compared it in the context of current developments in the field. The project will demonstrate students have a high-level ability to construct working systems in new application areas and the ability to evaluate them critically, using both empirical and theoretical methods as appropriate.

Transferable Skills:

Problem solving, literature searching, report writing, planning, comparative analysis.

Course Texts:

Recommended texts may vary, and will typically be set by individual members of staff.

FUTURE INTERACTION TECHNOLOGIES: MRes PROJECT

Code:	CS-M09
Credit Points:	110
Taught:	Semester 2
Contact Hours:	Regular contact with supervisor
Coordinator:	Prof. M Jones
Assessment:	100% Continuous assessment
	<i>This module is only available as part of MRes in Computing & Future Interaction Technologies</i>

Synopsis:

The FIT MRes research project provides students with the opportunity of exploring a particular topic in interaction technologies in considerable depth. The project will typically involve the development of an interactive system in which case the student will experience the major phases of the life-cycle of practical IT-project: Specification, design, implementation, testing, verification and evaluation. The project will enhance the students' competence in all aspects of design, including algorithm analysis, informal and formal mathematical reasoning, experiments and evaluation. It will give students an intellectual challenge to their abilities to learn new subjects without instruction and to further develop their

abilities in literature researching, report writing, verbal presentation, project planning and time management.

Syllabus:

Students will be expected to undertake in-depth research into a topic in interaction technologies under the supervision of an academic member of staff. Students will write a thesis on their project and present the project by an interim report (10 credits) and in a viva. A log book will be kept by the student, which may also be assessed. The dissertation and viva (100 credits) will be assessed according to the University of Wales Standing Orders.

Learning Outcomes:

Students will have carried out substantial research in the area. They will have documented their work in a dissertation that describes the background, aims, methods and results of their research and critically compares it in the context of current developments in the field. The project will demonstrate students have a high-level ability to construct new theories, principles and/or working systems in new application areas, as well as the ability to evaluate the system or theory critically, using both empirical and theoretical methods as appropriate. Students will learn how to work to a standard comparable to research conference papers in the field.

Transferable Skills:

Problem solving, literature searching, report writing, planning, comparative analysis.

Course Texts:

Recommended texts may vary, and will typically be set by individual members of staff.

COMPUTER SCIENCE PROJECT DEVELOPMENT

Synopsis:

In this module students will be presented with an overview of the research area of computer science. They are introduced into the topic, the background and the aims of their project. They write a detailed specification which will be the basis of their research project. Guidance as to appropriate research methodologies is provided.

Syllabus:

Overview of methodologies used for the development of computer science projects.

Students will be introduced by their supervisor into a specific research area related to their project. They will learn how to present their research in the form of written work.

Students are introduced into their project by their supervisor. They write a detailed specification of their project explaining the background and the aims of the project.

A project plan is also included.

Project Proposals

- identifying a research topic
- finding and reading related work
- report writing, citations and references
- using (digital) library services and search tools
- planning and managing a research project.

Learning Outcomes:

Students will demonstrate an understanding of how scientific research is conducted, reported, reasoned about and evaluated.

Students will be able to show their understanding of the requirements of a masters level project by writing a formal project proposal and specification which contains an outline solution to the problem and which clearly defines the scope of the MSc project, its goals, the methodology to be undertaken and the criteria of its evaluation

Students will have gained an in-depth knowledge in specific areas related to their project. They will have critically assessed different methods to be used in their project and will have developed a detailed plan for carrying out their project.

Transferable Skills:

Problem solving, literature searching, project planning and time management, and report writing.

Reading:

C. Dawson, *Projects in Computing and Information Systems: A Student's Guide*, 2nd Ed, Addison-Wesley, 2009.

J. Weyers, K. McMillan, *How to write Dissertations and Project Reports*, Prentice Hall, 2009.

Code:	CS-M10
Credit Points:	10
Taught:	Semester 2
Contact Hours:	10 seminars, 10 hours one-to-one supervision
Coordinator:	tbc
Co-requisite:	CS-M00
Incompatibles:	Other project specification modules
Assessment:	40% literature search and outline project proposal, 60% project specification document

SOFTWARE CONCEPTS AND EFFICIENCY

Code:	CS-M12
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Co-requisite:	CS-M41
Coordinator:	tbc
Assessment:	2 x 15% programming assignments, 70% written examination (in May/June)

Synopsis:

This module introduces students to the formal concepts of algorithms and data structures and will enable them to understand how the selection of different algorithms and datatypes affect the performance and efficiency of a program. The object-oriented language Java will be used for practical implementations.

Syllabus:

Introduction to the concept of algorithm and program efficiency.

Sorting and Searching Algorithms.

The concept of data abstraction with particular reference to the object-oriented paradigm; abstract specification of various standard data types; implementation of various data types; deriving storage requirements from software specifications; complexity and efficiency considerations for storage management approaches; data storage issues in distributed environments, including consistency and transparency. The object-oriented language Java will be used for practical implementations.

Learning Outcomes:

Students will be able to:

- Determine the appropriate data structures and data management techniques in a variety of software applications, including distributed applications, and critically evaluate the relative merits of alternatives.
- Formally specify a range of abstract data types for use with the object-oriented and procedural paradigms.
- Implement a range of data structures and data management techniques in different ways, utilising the object-oriented paradigm.
- Determine the efficiency and complexity implications of the different approaches.
- Appreciate the idea of analysing an algorithm to determine its efficiency.
- Implement and analyse standard sorting and searching algorithms.

Students will be aware of how memory is managed and how this can affect program performance.

Transferable Skills:

Problem solving; ability to learn and use computer systems and software packages effectively. Simple mathematical reasoning.

Course Texts:

R. Sedgewick, Algorithms in Java, 3rd Ed, Addison-Wesley, 2004.

M. T. Goodrich, R. Tamassia, Data Structures and Algorithms in Java, 4th Ed, John Wiley & Sons, 2005.

B. R. Preiss, Data Structures and Algorithms with Object-Oriented Design Patterns in Java, John Wiley & Sons, 2001.

CRITICAL SYSTEMS

Synopsis:

The module enables students to develop an appreciation of the problems of developing safety-critical system software, together with practical experience of applying modern, formal techniques to the production and verification of such software.

Syllabus:

Introduction and Motivation: What are safety critical systems. Legal and ethical issues. Examples of major failures of safety critical computer systems. Successes and how/why they worked. Standards for safety-critical software and their shortcomings.

Analysis: The hazard analysis process. Safety analysis and the safety case. Safety issues related to, but outside software. Human factors: the role of poor interfaces in software failures.

Specification and Verification: Languages and tools for formal specification. Verification by model-checking. Tools for automated and machine-assisted theorem proving.

Software Production: Issues in program language selection to minimise failure. The Software Engineering process in the production of high-integrity software.

Correctness: Validation and verification: the advantages and disadvantages of testing and formal verification.

Learning Outcomes:

Students will be familiar with issues surrounding critical systems, including legal and ethical issues, hazard analysis and techniques for the specification and production of high-integrity software. They will have had experience in applying formal specification techniques to critical systems. They will be familiar with and have had experience in applying programming languages suitable for developing high-integrity software for critical systems (e.g. SPARK ADA).

Transferable Skills:

Skills in abstract modelling and problem solving. Ability to read critically and to précis and judge information. Ability and confidence to learn unaided.

Course Text:

N. Storey, *Safety-Critical Computer Systems*, Addison-Wesley, 2nd Ed, 1996.

Reading:

J. Barnes, *High Integrity Software. The SPARK Approach to Safety and Security*, Addison-Wesley, 2003.

N. Leveson, *Safeware: Systems Safety and Computers*, Addison-Wesley, 1995.

J. C. Geffroy and G. Motet, *Design of dependable computing systems*, Springer, 2011.

Code:	CS-M13
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Setzer
Incompatible:	CS-313
Assessment:	30% Coursework, 70% written examination (in January)

INDUSTRIAL PROJECT

Code:	CS-M14
Credit Points:	40
Taught:	Whole Session
Contact Hours:	Regular meetings with project supervisor
Co-ordinators:	Dr. Harman, Mr. Whyley
Assessment:	Continuous assessment

Synopsis:

Students will build a software application, or develop an advanced specification of a software application using industrial tools and methodologies. Where appropriate, the project will be undertaken with an industrial partner.

Syllabus:

Milestone 1: End of term 1. Deliverables: Methodology Requirements Document describing the methodology to be used, a risk analysis - including steps taken to 'design out' risks with a relatively high likelihood and serious consequences - as well as the requirements of the system in a precise and structured form (typically 10-20 pages but depends on project; 15%); Specification Document describing the specification of the behaviour (and if relevant other aspects of the project - e.g. performance) in a precise and structured form (typically 10-20 pages but depends on project; 10%).

Milestone 2: end of term 2. Deliverables: Interim Report summarising in a narrative form the progress made on the project at this point (typically 10-20 pages but depends on the project; 10%).

Milestone 3: Semester 2, Week 10. Deliverables: Poster session describing the project and its status, and a demonstration of the software (10%); User Manual (length variable; 10%); Design Document describing the design of the system, preferably using a (semi-)formal notation such as UML, and including a rationale for design decisions made (length variable; 15%); Testing Document describing in a structured way the tests applied and their outcome (length variable; 10%); and Narrative and Reflective Account describing the student's experiences, problems encountered and solved (or not) and their reflections on the successful and less successful aspects of the project (typically 10-20 pages; 20%).

Learning Outcomes:

Students will have be able to work within a defined, industry-standard framework; to employ standard specification, and software production methodologies, and software engineering tools; practical experience of realistic resource modelling and quality assurance; to produce a series of deliverables, in the form or reports and finished software, to a well-defined timetable; and to adapt to circumstances, and manage the project process.

Transferable Skills:

Written communication and documentation, oral presentation, and interactive discussions. External collaboration, time management and project management. Problem solving. Ability to evaluate and deploy new technologies. Information retrieval, ability to read critically, to précis and judge information, and ability to manage learning processes.

DIRECTED STUDIES IN LOGIC AND COMPUTATION

Synopsis:

Students will be assigned specific study topics associated with their intended research topic. They will be expected to undertake specialised individual study under the direction of their tutor. Both students and lecturers will give talks on project related topics. The student talks are assessed.

Syllabus:

Students will be assigned specific study topics associated with their intended research topic. They will be expected to undertake specialised individual study under the direction of their tutor. Both students and lecturers will give talks on project related topics. The student talks are assessed.

Learning Outcomes:

Students will have in depth knowledge in a specialised research topic in the field of logic and computation. They will have learned acquiring knowledge through independent study of research literature. They will be able to critically assess and compare current research literature.

Transferable Skills:

Problem solving, literature searching, comparative analysis.

Reading:

Recommended texts may vary according to research topic.

Code:	CS-M15
Credit Points:	10
Taught:	Semester 2
Contact Hours:	Seminars
Co-ordinators:	Dr. Kullmann
Assessment:	Presentations on specialised topics
	<i>This module is only available to students doing the MRes in Logic and Computation</i>

IT SECURITY: THEORY AND PRACTICE

Code:	CS-M18
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. T. Chen (Engineering)
Incompatible:	CS-318
Assessment:	30% coursework, 70% written examination (in May/June)

Synopsis:

The aim of this course is to examine theoretical and practical aspects of computer and network security.

Syllabus:

Introduction: threats and their causes.

Security engineering: security criteria; security models.

Cryptography: basic encryption & decryption; cryptanalysis; symmetric cryptosystems (eg DES, AES, RC4); asymmetric cryptosystems (eg RSA, Diffie-Hellman key exchange, ElGamal);

cryptographic hash functions & digital signatures; key management; authentication concepts; access control.

Tools & technologies: IPSec; tunneling & VPNs; TLS, SSL, SSH and related tools; PGP and GPG; security in OpenBSD.

Vulnerabilities and attacks: port scanning; packet sniffing; buffer overflows; SQL injection. Security issues in wireless networks.

Learning Outcomes:

Students will be fully aware of the main security issues in today's IT infrastructures. They will have a detailed understanding of the current techniques for increasing IT security, and be fully aware of the limitations. They will understand and be familiar with the models and methods used to systematically construct secure systems or enhance the security of existing systems.

Transferable Skills:

Problem analysis and solving, abstract modelling, formal reasoning. Ability to manage learning process. General mathematical discipline. Ability to learn and use computer systems effectively.

Course Texts:

W. Stallings, *Cryptography and Network Security*, 3rd Ed, Prentice Hall, 2003.

B. Schneier, *Applied Cryptography*, 2nd Ed, Wiley, 1996.

Reading:

M. Bishop, *Introduction to Computer Security*, Addison-Wesley, 2005.

N. Ferguson and B. Schneier, *Practical Cryptography*, Wiley, 2003.

E. Skoudis and T. Liston, *Counter Hack Reloaded*, 2nd Ed, Prentice Hall, 2006.

J. Edney and W. Arbaugh, *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*, Addison-Wesley, 2004.

INTERACTIVE SYSTEMS DESIGN

Synopsis:

Interactive systems are ubiquitous - from handheld devices, even medical implants, to large systems such as the world wide web. Some systems are safety-critical (such as aircraft flight decks); some are mission-critical (such as ticket machines);

some are utilitarian; some are fun. Almost all are badly designed and badly documented - and they cause users problems. Surprisingly, the theory and practice of interactive systems design is not well-developed, and what is known is not widely known. This module reviews the problems and obvious solutions, and shows how information theory, graph theory, finite state machines, and other elementary computer science techniques, when applied well, can make a huge difference. The module has an underlying theme of the social and ethical imperatives why one should make better systems.

Code:	CS-M19
Credit Points:	10
Taught:	Whole session
Lecturer:	Professor Thimbleby
Assessment:	100% coursework

Syllabus:

Part 1: Problems

Usability, evaluation, error.

Review of HCI, classic issues, cognitive psychology.

Part 2: Theory

Information theory, Graph theory, FSAs, Markov Models, user manual generation.

Defining and programming interactive systems (statecharts etc).

Overview of Human Factors theory and issues: ergonomics, affordance, human error, user models.

Part 3: Solutions

Larger systems, ethics and design principles.

Part 4: Research topics

Classic literature and personalities.

Learning Outcomes:

Students will be able to recognise, critique and (know how to) solve many usability problems in interactive systems; they will be able to design and make reliable interactive systems.

Students will have a significant appreciation for 'user centred design' and will know practical means to achieve it, from requirements analysis through to evaluation, as well as technical approaches. They will have a thorough appreciation of the social and ethical framework.

Transferable Skills:

The ability to analyse, design and constructively criticise any complex system. To see computer science as a wider subject, able to address non-computing problems.

Reading:

H. Thimbleby, *Press On*, MIT Press, 2007.

R. J. Wieringa, *Design Methods for Reactive Systems: Yourdon, Statemate, and the UML*, Morgan Kaufmann, 2003.

MSC PROJECT

Code:	CS-M20
Credit Points:	60
Taught:	Summer vacation
Contact Hours:	Individual project supervision
Coordinator:	Dr. Stein
Co-requisite:	CS-M00
Assessment:	100% dissertation and demonstration/viva (This module is only available to students pursuing an MSc degree in Computer Science, for which it is compulsory)

Synopsis:

This module will provide students with the opportunity of exploring a particular topic in computer science in some considerable depth

Syllabus:

The student will carry out independent project under the guidance of their supervisor.

The dissertation may include the

following topics:

- Discussion of the subject area and its history;
- A literature survey;
- Formulation of scientific questions and the answers to them;
- Theoretical background;
- Description of the approach taken;
- Discussion of issues arising in the undertaking of the project;
- Evaluation of results;
- Progress and achievements of the project;
- Suggestions for further work.

Learning Outcomes:

Students will be able to undertake independent research into appropriate areas of Computer Science;

plan and undertake a significant independent piece of project work;

critically evaluate their work in the context of current work in related areas.

Transferable Skills:

Problem solving; literature searching; report writing; comparative analysis.

Course Texts:

Reading:

C. Dawson, *Projects in Computing and Information Systems: A Student's Guide*, 2nd Ed, Addison-Wesley, 2009.

J. Weyers, K. McMillan, *How to write Dissertations and Project Reports*, Prentice Hall, 2009.

Other reading dependent on project

SOFTWARE TEAM PROJECT

Synopsis:

The aim of this module is to provide students with the opportunity to apply their specialised knowledge to a realistic problem, and gain practical experience of the processes involved in the team-based production of software.

Syllabus:

Project planning, tools and techniques for planning.

Project conduct, time management, risk management, and team working.

Review of legal, ethical, social and professional issues applicable to the computer industry.

Starting from an outline description of a realistic problem, each team is required to develop a fully implemented software solution using appropriate engineering and project management techniques. Wherever possible, teams are organised on the basis of shared interest, and the problem is designed to exercise their understanding of their area of specialised study.

Assessment is based on the documentation produced at each stage of the process, a presentation and demonstration of the final product, the effectiveness of the team's management of the project, and the understanding and contribution of each individual.

After completing this module each team will be expected to have:

1. Held regular group meetings and maintained a diary of their progress.
2. Defined and allocated tasks to individual members according to their skills.
3. Used available project management tools to organise their activities.
4. Produced a fully tested working implementation of their software.
5. Delivered appropriate documentation of a professional standard.
6. Given a presentation in which they described the way the project was managed, explained their design and demonstrated their final product.

Each individual student will be expected to have contributed fully to his or her team's activities, and will be expected to be able to:

1. Describe the processes involved in the team-based production of software.
2. Explain and justify the design of their team's finished product.
3. Give an account of their individual contribution to the team's effort.

Learning Outcomes:

Students will have demonstrated their ability to apply their specialised knowledge to a realistic problem, and have gained practical experience of the processes involved in the team-based production of software.

Students will understand what is required for a team to:

1. Hold regular group meetings and maintain a diary of their progress.
2. Define and allocate tasks to individual members according to their skills.
3. Use available project management tools to organise their activities.
4. Produce a fully tested working implementation of their software.
5. Deliver appropriate documentation of a professional standard.
6. Give a presentation in which they described the way the project was managed, explained

Code:	CS-M24
Credit Points:	10
Taught:	Semester 2
Contact Hours:	10 lectures/seminars
Lecturer:	Dr. Laramee, Dr. Mora
Assessment:	20% initial team/group exercise . 50% team project report (split over two phases) . 10% presentation . 20% individual report .

their design and demonstrated their final product.

Each individual student will be expected to have contributed fully to his or her team's activities, and will be able to:

1. Describe the processes involved in the team-based production of software.
2. Explain and justify the design of their team's finished product.
3. Give an account of their individual contribution to the team's effort.

Transferable Skills:

Problem solving, project planning, time management and risk assessment, team work, oral and written communication.

Reading:

- I. Sommerville, *Software Engineering: International Version*, 9th Ed, Pearson Education,

MOBILE INTERACTION DESIGN

Synopsis:

This module presents key human computer interaction design issues, methods, tools and techniques in a mobile and ubiquitous systems context. Students will learn how to improve the user interfaces they design and be equipped to develop efficient, effective and satisfying applications for an important, emerging class of computing device. The module will involve students prototyping and evaluating mobile applications and introduce embedded application development environments.

Code:	CS-M29
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Prof. Jones
Incompatibles:	CS-349
Assessment:	30% coursework, 70% written examination (in May/June)

Syllabus:

User interface technologies and components;

Characteristics of effective user interfaces;

Human-factors design and development methodologies;

User-centred requirements gathering & analysis;

Prototyping & evaluation;

Mobile & Embedded programming;

Design issues and strategies for: accessing complex functions (menus, modes etc); mobile web browsing and searching; rich media access; and, mobile communities.

Learning Outcomes:

Understanding of the human-factor issues relating to mobile and ubiquitous computing systems.

Knowledge of user-centred software design tools, models and methods.

Transferable Skills:

Course encourages "design" based thinking; the high degree of analytical, reflective and critical skills needed for user-centred interactive system development are widely applicable in many other contexts.

Reading:

M. Jones and G. Marsden, *Mobile Interaction Design*, John Wiley & Sons, 2006.

LOGIC AND COMPUTATION MRes PROJECT

Code:	CS-M35
Credit Points:	110
Taught:	Whole session
Contact Hours:	Regular meetings with project Supervisor
Co-ordinator:	Dr. Berger
Assessment:	Thesis and viva Only available to those students studying for the MRes in Logic and Computation

Synopsis:

The research project will provide students with the opportunity of exploring a particular topic in computer science in some considerable depth. The project may involve the development of a software system in which case the student will experience the major phases of the life-cycle of practical IT-project: Specification, design, implementation, testing, verification and validation.

The project will enhance the students' competence in algorithm analysis, informal and formal mathematical reasoning. It will give students an intellectual challenge to their abilities to learn new subjects without instruction and to further develop

their abilities in literature researching, report writing, verbal presentation, project planning and time management.

Syllabus:

Students will be expected to undertake in-depth research into a topic in Logic and Computation under the supervision of an academic member of staff. Students write a thesis on their project and present the project in a viva. The dissertation and viva will be assessed according to the University of Wales Standing Orders.

Learning Outcomes:

Students will have carried out substantial research in the area of logic and computation.

They will have documented their work in a dissertation that describes background, aims, methods and results of their research and critically compares it in the context of current developments in the field.

Transferable Skills:

Problem solving, literature searching, report writing, planning, comparative analysis.

INTERACTION TECHNOLOGIES: SEMINARS & READING

Synopsis:

This module encourages students to explore the advanced literature and research results underpinning the field of interaction technologies. Classic papers (and controversies) are covered, as well as recent work from the leading journals and conferences. Students achieve a clear view of the 'cutting edge' and issues in the field.

Code:	CS-M39
Credit Points:	10
Taught:	Whole session
Contact Hours:	10 lectures
Co-ordinator:	Professor Thimbleby
Assessment:	100% continuous assessment

Syllabus:

Papers are selected from the recent research literature, presented and discussed in seminars. Classic papers and some books are covered, as well as reviewing the work of the leading researchers and laboratories in the field.

Learning Outcomes:

The ability to demonstrate detailed understanding of a set of topics in interaction.

The ability to review and critically assess the literature on specific topics in that area at the current limits of theoretical or research understanding.

The ability to analyse and present the results of a literature review both as a scientific report and as an oral presentation.

Transferable Skills:

The ability to engage and present their own academic work e.g., to professionals in their own field.

Autonomous study, and full awareness of resources to aid the study of research topics in interaction technologies.

Effective research management (in terms of time, direction of study, sources of information and relevance to the topic).

Course Texts:

R. M. Baeker, Morgan Kaufmann *series of Readings in Human-Computer Interaction*, ACM Digital Library, 1995.

J. M. Carroll, *Human-Computer Interaction in the New Millennium*, ACM Press, 2001.

SOFTWARE TECHNOLOGY PROJECT DEVELOPMENT

Code:	CS-M40
Credit Points:	10
Taught:	Semester 2
Contact Hours:	10 hourse seminars, 10 hours individual supervision
Co-requisite:	CS-M00
Incompatible:	Other project specification modules
Co-ordinator:	Dr. Stein
Assessment:	40% literature search and outline project proposal, 60% project specification document

Synopsis:

In this module students will be presented with an overview of the research area of software technology. They are introduced into the topic, the background and the aims of their project. They write a detailed specification which will be the basis of their research project. Guidance as to appropriate research methodologies is provided.

Syllabus:

Overview of methodologies used for the development of software technology projects.

Students will be introduced by their supervisor into a specific research area related to their project. They will learn how to present their research in the form of written work.

Students are introduced into their project by their

supervisor. They write a detailed specification of their project explaining the background and the aims of the project.

A project plan is also included.

Project Proposals

- identifying a research topic
- finding and reading related work
- report writing, citations and references
- using (digital) library services and search tools
- planning and managing a research project.

Learning Outcomes:

Students will demonstrate an understanding of how scientific research is conducted, reported, reasoned about and evaluated.

Students will be able to show their understanding of the requirements of a masters level project by writing a formal project proposal and specification which contains an outline solution to the problem and which clearly defines the scope of the MSc project, its goals, the methodology to be undertaken and the criteria of its evaluation

Students will have gained an in-depth knowledge in specific areas related to their project. They will have critically assessed different methods to be used in their project and will have developed a detailed plan for carrying out their project.

Transferable Skills:

Problem solving, literature searching, project planning and time management, and report writing.

Reading:

C. Dawson, *Projects in Computing and Information Systems: A Student's Guide*, 2nd Ed, Addison-Wesley, 2009.

J. Weyers, K. McMillan, *How to write Dissertations and Project Reports*, Prentice Hall, 2009.

PROGRAMMING IN JAVA

Synopsis:

This intensive course provides a solid introduction to the Java programming language and development process.

Syllabus:

Introduction to Java. The Object-Oriented paradigm - objects, classes and methods. Basic graphical interfaces. Exception handling in Java. I/O and files in Java.

Learning Outcomes:

Students will be able to design, implement and document working Java programs in a modular maintainable style to the standard required in a Masters degree. They will be able to read pieces of code written by others, identify errors and bugs using standard debugging tools and techniques to adapt and/or correct such code. They will be able to critically evaluate Java programs.

Transferable Skills:

Effective use of general IT facilities; Problem solving skills; Ability to learn and use computer systems and software packages effectively.

Code: CS-M41
Credit Points: 20
Taught: Semester 1
Contact Hours: 20 lectures plus 10 x 2 hour laboratory classes
Lecturer: Dr. Kullmann
Incompatibles: CS-M81
Assessment: 50% continuous assessment,
50% written examination
(in January)

To be awarded credits for this module, students must pass both the examination and coursework components to a tolerable level

This module is only available to Masters students who are not majoring in Computer Science

Course Texts:

- T. Gaddis and G. Muganda, *Starting out with Java. From Control Structures through Data Structures*, Addison-Wesley, 2008.
- T. Gaddis, *Starting out with Java. From Control Structures through Objects*, Addison-Wesley, 2008.

INTERACTION TECHNOLOGIES: LAB AND FIELD WORK

Code:	CS-M49
Credit Points:	10
Taught:	Whole session
Contact Hours:	18 lectures and in-class activities
Lecturer:	Dr. Wilson
Assessment:	100% continuous assessment

Synopsis:

This is a compulsory module for the Computer Science FIT Masters programmes, and provides laboratory skills and experience.

Syllabus:

Practical and laboratory skills; evaluation methods; managing experiments with users, including ethical considerations.

Learning Outcomes:

The ability to demonstrate thorough practical understanding of laboratory methods.

The ability to build and evaluate interactive systems.

The ability to write up and document experimental work.

Transferable Skills:

Practical skills in organising laboratory work, including planning and recording experiments, building and evaluating interactive systems.

Course Text:

J. Lazer, J. H. Feng and H. Hochheiser, *Research Methods in Human-Computer Interaction*, Wiley, 2010.

Reading:

P. Cairns and A. L. Cox, *Research Methods for Human-Computer Interaction*, Cambridge University Press, 2008.

SAFE AND SECURE SYSTEMS PROJECT DEVELOPMENT

Synopsis:

In this module students will be presented with an overview of the research area of safe and secure systems. They are introduced into the topic, the background and the aims of their project. They write a detailed specification which will be the basis of their research project. Guidance as to appropriate research methodologies is provided.

Syllabus:

Students will learn an overview of methodologies used for the development of safe and secure systems. They will be introduced by their supervisor into a specific research area related to their project. They will learn how to present their research in the form of oral and written work.

Students are introduced into their project by their supervisor. They write a detailed specification of their project explaining the background and the aims of the project.

The specification contains as well a detailed project plan.

Learning Outcomes:

Students will have an appreciation of the research area of safe and secure systems. Students will have a detailed appreciation of specific areas related to their project. They will be able to critically assess different methods to be used in their project and will have developed a detailed plan for carrying out their project. They will have gained an understanding of appropriate research methodologies.

Transferable Skills:

Problem solving, literature research, planning.

Reading:

Recommended texts will vary.

Code:	CS-M50
Credit Points:	10
Taught:	Semester 2
Contact Hours:	10 hours seminars, 10 hour one-to-one supervision
Coordinator:	Dr. Setzer
Co-requisite:	CS-M00
Incompatible:	Other project specification modules
Assessment:	20% talk on a selected topic in safe and secure system, 20% report on selected topic, 60% project specification document

COMPUTER SCIENCE CONCEPTS

Code:	CS-M53
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Sharp
Assessment:	2 x 10% problem sheets, 80% written examination (in January) (This module is only available as part of the MSc Computer Science degree programme)

Synopsis:

This module aims to provide students with a complete picture of the operation of a computer system, both hardware and software. The course concentrates on ideas and principles with examples taken from a wide variety of current computer systems.

Syllabus:

The origins of the computer: the von Neumann computer. Commercial Computers: UNIVAC I to Intel 80x86.

Computer Arithmetic. Boolean logic, integer and floating-point representation and arithmetic.

Computer systems components and function. The

CPU, memory and input/output.

Interconnection structures. Computer modules and the bus.

Memory: characteristics, hierarchy and storage capacity. Internal semiconductor main memory: memory types and Cache. External: magnetic disk, optical memory and magnetic tape.

Input/output: module function, direct, interrupted and DMA I/O.

CPU: registers, the instruction cycle, pipelining and interrupted execution. Control unit: micro-operations, hardwired and microprogrammed.

RISC & CISC: characteristics and the controversy.

Operating systems: scheduling and memory management.

Assembly language, high level data types and languages.

Introduction to Formal Methods;

Mathematical foundations: sets, functions, and relations; modelling data with algebras.

Formal logic: truth tables; axioms; quantifiers.

Learning Outcomes:

Students will:-

- have a clear understanding of how software and hardware interact in a computer system;
- be fully aware of the principles behind modern computer architecture and operating systems;
- be comfortable with the fundamental mathematical tools of Computer Science and be able to apply them to the writing of formal specifications and documentation;
- be fully aware of how such tools can be used to produce provably correct software.

Transferable Skills:

Problem solving.

Reading:

A Clements, *Principles of Computer Hardware*, 4th Edition, Oxford University Press, 2006.

W Stallings, *Operating Systems: Internals and Design Principles*, Pearson 2008.

D J Velleman, *How to Prove It: A Structural Approach*, Cambridge University Press, 1994.

DISTRIBUTED O-O PROGRAMMING

Synopsis:

Building distributed applications using the object oriented programming paradigm.

Syllabus:

Sockets, Ports and TCP/IP. Distributed programming using sockets in Java. Java RMI; RMI callbacks.
XML, SOAP and Web Services.

Learning Outcomes:

Students will gain in depth experience of building distributed applications using sockets and Java RMI (including polling and callback). They will gain an understanding of other technologies, including XML-based distributed programming technologies (SOAP and XML Web Services).

Transferable Skills:

Students will develop skills in solving advanced technical problems in a methodologically well-structured framework. They will develop self-study skills and learn a range of technical skills.

Reading:

M. Kalin, *Java Web Services: Up and Running*, O'Reilly, 2009.

W. Grosso, *Java RMI*, O'Reilly, 2002.

The following web site contains substantial tutorial information:

<http://www.oracle.com/us/java/index.html>

The course notes (supplied) contain numerous links to sources of additional information.

Code:	CS-M58
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Harman
Incompatible:	CS-338
Assessment:	40% coursework, 60% written examination (in January)

RELATIONAL AND OBJECT-ORIENTED DATABASE SYSTEMS

Code:	CS-M59
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Mr. Whyley
Assessment:	20% coursework, 80% written examination (in January)

Synopsis:

This module gives an appreciation of the complexity of real-world databases. It considers some of the problems that can occur in multi-user, multi-transaction situations. It discusses relational and object-oriented databases and covers their design and implementation. Distributed databases and databases linked to the web will also be discussed, as will data warehousing and data mining. Students will gain practical experience in designing and implementing a database.

Syllabus:

A review of the nature of data and databases and an overview of database management and database system architecture.

Data models: relational databases, object databases.

Relational databases: the structure of the relational model, integrity constraints, relational algebra and calculus, normalisation.

Transaction management, data security and recovery, optimisation, distributed databases, concurrency control.

Object-oriented databases, type inheritance, active databases, temporal databases, logic-based databases.

Data warehouses and data mining, data visualisation.

Web technology and databases.

Learning Outcomes:

Students will have a comprehensive understanding of the principles of relational databases and object-oriented databases. They will be able to design and implement databases and be aware of potential problems and how to avoid them. They will be able to normalise a database and understand why it is necessary.

Students will be familiar with the extra problems associated with distributed databases and the need to ensure integrity and how to control concurrency. Students will have a thorough understanding of how data warehouses operate and the principles of data mining.

They will know how databases can be linked to web applications. They will have gained practical experience of designing and implementing a database.

Transferable Skills:

Problem identification, problem analysis and abstract modelling. Abilities to learn and use computer systems and software packages effectively, and to evaluate and deploy new technologies.

Course Text:

C. J. Date, *An Introduction to Database Systems*, 8th Ed, Addison-Wesley, 2004.

T. Connolly and C. Begg, *Database Systems*, 5th Ed, Addison-Wesley, 2010.

R. Elmasri and S. B. Navathe, *Database Systems*, 6th Ed, Pearson, 2011.

CONCEPTS OF PROGRAMMING LANGUAGES

Synopsis:

This module introduces the student to the concepts underlying the concrete constructs of programming languages supporting a certain style of programming (paradigm), like imperative, object-oriented, concurrent, functional, logic and visual programming.

Syllabus:

This course introduces principles of programming languages from a practical viewpoint. To this end, we first study and describe general concepts of programming languages, including data types, expressions, commands, declarations, and abstractions. We then analyse how these concepts are related to the concrete design of a programming language that supports a certain style of programming (paradigm) as e.g. imperative, object-oriented, concurrent, functional, logic and visual programming.

Throughout the course, Ada, C, C++, Java, and Haskell are used as reference languages. Simple exercises help understanding of the subject in the way of learning by doing.

Learning Outcomes:

Students will have a thorough understanding of the concepts underlying the concrete constructs of a programming language and what characterises the different programming paradigms.

This knowledge will lead them to a deeper understanding of the programming languages they are working with, making it easier to learn new programming languages, and allow them to critically judge the design of a programming language as well as to select an appropriate language to solve a certain problem.

Transferable Skills:

Ability to learn and use computer systems and packages effectively.

Course Text:

P. D. Mosses, *Course Notes on Concepts of Programming Languages*, Dept of Computer Science, Swansea University, 2011.

Reading:

D. A. Watt, *Programming Language Design Concepts*, Wiley, 2004.

Code:	CS-M61
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Prof. Mosses
Incompatible:	CS-311
Assessment:	30% coursework , 70% written examination (in May/June)
	<i>Familiarity with a functional programming language such as Haskell or ML, as well as with an imperative or object- oriented one such as C# or Java, is assumed</i>

ARTIFICIAL INTELLIGENCE APPLICATIONS

Code:	CS-M65
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Incompatible:	CS-345
Lecturer:	Dr. Grant
Assessment:	20% coursework, 80% written examination (in January)

Synopsis:

This module introduces the student to the fundamentals of knowledge based systems, presents the basics of expert system shells such as AILog, and presents the fundamentals of fuzzy expert systems. It also introduces the student to aspects of soft computing such as genetic algorithms and genetic programming.

Syllabus:

Expert systems. Early development including MYCIN. The basic architecture — working

memory, inference engine, user interface. Rule-based systems. Frame based systems. Methods of inference. Forward and backward chaining. Resolution proof method. Inexact reasoning and uncertainty. Certainty factors and probability.

Non-monotonic proof techniques. Default and temporal reasoning. Explanation — how and why. Knowledge acquisition.

Fuzzy logic. Fuzzy expert systems.

Case based reasoning. Expert system shells (AILog).

Evolutionary computing — genetic algorithms, genetic programming.

Learning Outcomes:

Students will acquire the basics of the expert system shell AILog and be able to produce an expert system using this tool, a deep understanding of the concepts of evolutionary programming and be able to solve problems using this paradigm, the thorough knowledge of fuzzy logic and the ability to produce rules for a fuzzy controller.

Transferable Skills:

Problem solving. Ability to evaluate and deploy new technologies.

Reading:

D. Poole and A. Mackworth, *Artificial Intelligence, Foundations of Computational Agents*, CUP, 2010.

M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1996.

J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

J. R. Koza, *Genetic Programming I and II, The Video*, UVC videos.

B. Kosko, *Fuzzy Thinking*, Harper Collins, 1984.

J. L. Kolodner, *Case Based Reasoning*, Morgan Kauffman, 1993.

L. N. de Castro, *Fundamentals of Natural Computing - Basic Concepts, Algorithms and Applications*, Chapman and Hall, 2006.

GRAPHICS PROCESSOR PROGRAMMING

Synopsis:

GPU technology has emerged in recent years as a powerful way to do graphics and computing in general. This module will explain the specificities of current graphics hardware technology and how it is used in real time gaming and fast general computing applications.

Syllabus:

3D evolution and Standard Graphics Hardware Pipeline. The fundamentals of OpenGL and DirectX.

Graphics Hardware. Specific 3D architectures including NVidia, ATI and the new Intel Larrabee architecture will be detailed.

Graphics extensions and Shading languages. Vertex, Geometry and Fragments shaders as extensions in OpenGL and DirectX.

Advanced Graphics techniques. Ambient Occlusions, Multi-pass rendering and Global Illumination on the GPU.

GP-GPU. The GPU as a co-processor. New technologies like OpenCL, NVidia CUDA for general purpose computing.

Future of Graphics technology. Possible evolution in the near future of Graphics technology.

Learning Outcomes:

Students will get in touch with the latest Graphics technology. Programming exercises will improve their skills in the area of Graphics. GP-GPU classes will prepare students for the next revolution in computing: Highly parallel co-processors.

Transferable Skills:

Context awareness. Problem solving. Programming on parallel systems. Ability to evaluate and deploy new technologies.

Course Text:

H. Nguyen, *GPU Gems 3: Programming Techniques for High-Performance Graphics and General-Purpose Computation.*

W. Engel, *Shader X5: Advanced Rendering Techniques.*

Code:	CS-M67
Credit Points:	10
Taught:	Whole session
Contact Hours:	10 lectures
Lecturer:	Dr. Mora
Assessment:	100% continuous assessment
	(This module is suspended for 2010 - 2011)

WRITING WEB AND WEB SERVICE APPLICATIONS

Code:	CS-M68
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Incompatible:	CS-348
Lecturer:	Mr. Whyley
Assessment:	40% continuous assessment, 60% written examination (in May/June)

Synopsis:

Building web and web service applications, typically to implement dynamic web pages.

Syllabus:

Early client and server side web programming technology. The .NET Framework and C# programming language. Principles: security, scalability, performance and data integrity. .NET Visual Development tools. Practical Issues: Security, state and performance. Web Services: XML, WSDL, SOAP and UDDI. Alternative technologies - AJAX and Rails.

Learning Outcomes:

Students will gain experience of building web and web-service based applications. They will understand the technical and other advantages and disadvantages of such systems. They will have experience of using state-of-the art programming tools, and will be exposed to methodological good practice.

Transferable Skills:

Problem solving; using advanced programming tools; self study.

Reading:

D. Esposito, *Programming Microsoft ASP.NET 3.5*, Microsoft Press, 2008.

S. Walther, *ASP.NET 3.5 Unleashed*, SAMS, 2008.

INTERACTION TECHNOLOGIES: INFORMATION RETRIEVAL

Synopsis:

This module provides a thorough understanding of how search engines work and how users interact with them. The course covers the retrieval of documents from the web and from text databases, and also searching for images and audio recordings.

Syllabus:

Fundamentals of information seeking and information retrieval: log rule and text indexation; compression of text indexes; indexation of images and audio materials; user factors in information seeking (search tactics and searcher scenarios).

Code:	CS-M69
Credit Points:	10
Taught:	Semester 1
Contact Hours:	16 lectures
Lecturer:	Dr. Wilson
Assessment:	30% practical project 70% written examination (in January)

Learning Outcomes:

Thorough comprehension of multimedia retrieval systems and the mathematical basis of modern information retrieval algorithms. Deep understanding of and ability analyse the human-centred aspects of these systems. Ability to specify, design, build implement and test such a system.

Transferable Skills:

Ability to model complex human artifacts using mathematical formulae; ability to perform empirical tests of computer systems; ethnographic data-gathering

Course texts:

I. Ruthven and D. Kelly, *Interaction Information Seeking, Behaviour and Retrieval*, Facet Publishing.

Reading:

R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.

G. Marchionini, *Information Seeking in Electronic Environments*, Cambridge University Press, 1995.

I. Witten, A. Moffat and T. Bell, *Managing Gigabytes*, 2nd Ed, Morgan Kaufmann, 1999.

VISUAL COMPUTING PROJECT DEVELOPMENT

Code:	CS-M70
Credit Points:	10
Taught:	Semester 2
Contact Hours:	10 hours seminar, 10 hours one-to-one supervision
Coordinator:	Dr. Xie
Incompatible:	Other project development modules
Assessment:	40% outline project proposal (including presentation), 60% Project specification document

Synopsis:

In this module students will be presented with an overview of the research area of Visual Computing. They are introduced into the topic, the background and the aims of their project. They write a detailed specification which will be the basis of their research project. Guidance as to appropriate research methodologies is provided.

Syllabus:

Students will gain an overview of methodologies used for the development of Visual Computing related applications. They will be introduced by their supervisor into a specific research area related to their project. They will learn how to present their research in the form of oral and written work.

They write a detailed specification of their project explaining the background and the aims of the project. The specification contains a detailed project plan, technology overview and approaches, algorithms and techniques, measures of success, risks.

Learning Outcomes:

Students will have an appreciation of the research area of Visual Computing. Students will have a detailed appreciation of specific areas related to their project. They will be able to critically assess different methods to be used in their project and will have developed a detailed plan for carrying out their project. They will have gained an understanding of appropriate research methodologies.

Transferable Skills:

Problem solving, literature searching, project planning and time management, and report writing.

Reading:

To be advised by supervisor, but typically SIGGRAPH proceedings, IEEE TVCG, IEEE PAMI, IEEE CG&A, ACM ToG, coding manuals (e.g. nVidia CG, OpenCL, DirectX)

DESIGN PATTERNS AND GENERIC PROGRAMMING

Synopsis:

A study of generic programming and selected design patterns and idioms, using Java as the teaching language. Students will learn the techniques for creating highly efficient and highly reusable libraries.

Syllabus:

Generic programming has the goal of expressing algorithms and data structures in the most general way, while maintaining the same efficiency as hand-crafted specialised code. Object-orientation in its traditional form (Java for example) realises polymorphism only via so-called sub-class polymorphism, while generic programming extends these techniques by the use of parametric polymorphism, realised in C++ by Generics (together with other means for polymorphism like overloading of classes and global functions). This module introduces the basic mechanisms of generic programming (using Java as the teaching language) with emphasis on design patterns and idioms.

Introduction : basics; erasure; reification; upgrading of libraries.

Selected design patterns, especially policy-based design and visitors.

Learning Outcomes:

Students will have an understanding of the ideas and techniques of generic programming, especially a thorough awareness of how to use the ideas and techniques to write efficient and useful libraries. They will acquire a sound knowledge of the main design patterns and idioms.

Transferable Skills:

Abstract modelling and implementation of software via generic techniques. Problem solving. Ability to evaluate and deploy new technologies.

Reading:

M. Naftalin and P. Wadler, *Java Generics and Collections*, O'Reilly, 2007.

E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns. Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.

E. Freeman and E. Freeman, *Head First Design Patterns*, O'Reilly, 2004.

Code:	CS-M71
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures/seminars
Incompatible:	CS-M71
Lecturer:	Mr. Whyley
Assessment:	20% continuous assessment 80% written examination (in May/June)

LOGIC FOR COMPUTER SCIENCE

Code:	CS-M75
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturers:	Dr. Berger
Incompatible:	CS-375
Assessment:	30% coursework and computer laboratories, 70% written examination (in January)

Synopsis:

This module provides an introduction to logic and its applications to computer science, in particular to the formal specification and verification of computer programs.

Syllabus:

- Propositional logic (syntax, semantics, proof systems of natural deduction and resolution).
- Predicate logic (syntax, semantics, proof system).
- Applications of logic to program specification and verification.
- Specialised logics e.g. for security protocols, reactive systems and credit card systems.

Learning Outcomes:

Students will understand the syntax, semantics and proof rules of first-order predicate logic, be aware of other, specialised, logics in computer science (e.g. modal logic, process logic), understand the importance of logic for computer science, be able to express informal statements as formulas in predicate logic and carry out simple formal proofs.

Transferable Skills:

Problem solving, logical thinking, assessing the validity and invalidity of logical arguments.

Reading:

D. van Dalen, *Logic and Structure*, Springer, 1994

D. Barker-Plummer, J. Barwise and J. Etchemendy, *Tarski's World, revised and expanded edition*, CSLI Lecture Notes, 2008

U. Schöeningh, *Logic for Computer Science*, Birkhäuser, 1989.

COMPUTER VISION AND PATTERN RECOGNITION

Synopsis:

This module introduces students to the important and modern topics and concepts of computer vision and pattern recognition, including image processing, segmentation, feature extraction, camera calibration, stereo vision, motion analysis, object tracking, recognition, data clustering, and dimensionality reduction. It teaches techniques that are used to understand and interpret the contents of images and videos and dissects state-of-the-art vision systems, such as Microsoft Kinect.

Code:	CS-M77
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Dr. Xie
Incompatible:	CS-377
Assessment:	20% coursework, 80% written examination (in May/June)

Syllabus:

This course is composed of four parts: Introduction, Image Processing, Video Analysis and Pattern Recognition & Applications.

Introduction: The first part of the lectures gives an overview of Computer Vision and Pattern Recognition (CVPR) and a road show of this course. It also provides a brief revision of basic and important mathematical techniques frequently used in CVPR.

Image processing: filtering, Hough transform, registration, object extraction, shape recognition, segmentation, texture analysis.

Video analysis: camera models and calibration, stereo vision, depth estimation, motion estimation and tracking, local features for tracking.

Pattern Recognition: data clustering and K-means, Gaussian Mixture Modelling, dimensionality reduction, and applications.

Learning Outcomes:

Students will become familiar with the important concepts of computer vision and pattern recognition, acquire a knowledge of how the analysis of digital images and videos may be performed, and develop the skills necessary to program a basic computer vision system.

Transferable Skills:

Basic problem solving. Ability to learn and use computer systems and software package. Ability to evaluate new technologies. General mathematical analysis.

Reading:

D. Forsyth and J. Ponce, *Computer Vision: a Modern Approach*, Prentice Hall 2003

C. Bishop, *Pattern Recognition and Machine Learning*, Springer 2006.

HIGH-PERFORMANCE COMPUTING IN C/C++

Code:	CS-M78
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. Borgo
Assessment:	100% coursework

Synopsis:

The module will provide C/C++ fundamentals to students. An overview of High Performance Computing through the C/C++ languages will also be given.

Syllabus:

Introduction to C/C++ ,Fundamentals/close to hardware, UNIX Systems, Differences from Java

highlighted, h and .cpp, Compilers and compiling.

More on the grammar: Structures, sizeof, #define, if-else, case, and other weird operators, loops, comma operator, booleans, enum, Scope of a variable, static variables

Basic Classes in C++: private, protected, public, Use of Constructors and Destructors, Operator overloading (Complex class), Pointers (C programming style) and arrays, with emphasize on software engineering and memory models, Memory Allocation, malloc, free, new[], delete[], Memory Leaks, Classes as a clean solution to memory leak, Multidimensional arrays, linear formula: Strings (safe/unsafe), Argument passing, Pointer to functions

C/C++ differences: Malloc vs new, String vs char *, printf,read vs cin, classes vs struct. When using object technology?

Inheritance: Multiple inheritance, Virtual functions, dynamic binding

Templates: Function and class templates, Issues, Example: sort function, STL, Iterators

High Performance Computing (or how your program will run 20x faster): floating point representations, Types of parallelisms, What C allows, Compilers and mental model, Processor pipeline, branching instructions, memory access, math instructions, simple types, size and alignments, SSE instruction programming, Multi-threaded programming/ OpenMP, Memory communications, Barriers/MPI/pragma

System designs: More on the software engineering side, Pitfall, More on Real-time environments, normalized compilers

Learning Outcomes:

The module will provide two main learning outcomes:

-The basic concepts behind C/C++ programming languages, two very much used programming languages by the IT industry, will be learnt.

-The know-how of applying C/C++ programming to High-Performance Computing (HPC), including knowledge on both HPC architectures and parallel algorithms.

Transferable Skills:

.Problem solving. Programming on parallel systems. Ability to evaluate and deploy new technologies.

Reading:

B. Stroustrup, *The C++ Programming Language: Special Edition*.

S. Prata, *C++ Primer Plus*, 5th Ed.

INTERACTION TECHNOLOGIES: HARDWARE AND DEVICES

Synopsis:

Future interaction technologies rely on developments in hardware, and being able to interface the hardware and software. Students are expected to achieve substantial hands-on practical experience of the 'cutting edge' and issues in the field.

Syllabus:

Input devices, output devices. Drivers. Hardware protocols, eg, USB, phidgets etc. Software protocols, eg, MVC.

Non-standard devices, such as haptic, multiple mice, 3D displays, special purpose sensors.

Learning Outcomes:

Thorough knowledge of hardware and I/O devices. Ability to build interactive devices and program drivers. Knowledge of non-standard devices, such as haptic devices and phidgets etc.

Transferable Skills:

Practical skills in building complex systems, both hardware and software, and debugging hardware/software interfaces.

Reading:

Mobile HCI conference papers.

CHI conferences papers relevant to handheld devices, ubiquitous computing and mobile computing.

Python, Online materials to learn Python for programming on handheld devices, <http://docs.python.org/tut/>

Phidgets, Online catalogs for USB sensing and control: www.phidgets.com.

Arduino, Electronic prototyping for interaction design: <http://www.arduino.cc/>

Code:	CS-M79
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. Eslambolchilar
Assessment:	50% coursework, 50% written examination (in May/June)

WEB SCIENCE PROJECT DEVELOPMENT

Code:	CS-M80
Credit Points:	10
Taught:	Semester 2
Coordinator:	Dr. Wilson
Contact Hours:	Seminars and discussions
Co-requisite:	CS-M00
Assessment:	100% continuous assessment by by coursework, including essays and presentations

Synopsis:

In this module students will be presented with an overview of the research area of web science and undertake work to prepare for their MSc project

Syllabus:

- The evolution of the web (text, static, media, dynamic, social).
- The semantic web and the web of data.
- Phenomenon of the web (online protests, social sharing, issues of privacy).
- Web science foundations.
- Example projects in web science.
- Designing projects.

Learning Outcomes:

Students will have:

- an understanding the foundations of Web Science as a specialism.
- an understanding of current research issues in Web Science.
- the ability to select and use appropriate research methods for Web Science.
- planned a Web Science MSc Dissertation project.

Transferable Skills:

Generating Research Questions!

Analytical skills for the Web Science research

Reading:

Foundations and Trends Original Web Science Issue

SOFTWARE TESTING

Synopsis:

Testing is the process of systematically experimenting with an object (the SUT = System Under Test) in order to establish its quality, where quality means the degree of accordance to the intention or specification. This module will cover various test scenarios; practical exercises will allow the students to gain hands-on experience with various testing tools.

Awareness of propositional and predicate logic will be helpful for this module.

Syllabus:

The module provides a profound overview on industrially relevant methods in software testing and points out current research directions.

- Functional Testing (Boundary Value Testing, Equivalence Class Testing, Decision Table-Based Testing)
- Structural Testing (Path Testing, Data Flow Testing, MC/DC)
- Integration and System Testing (Levels of Testing, Approaches to Integration Testing, Threads for System Testing, Interaction Testing)
- Object Oriented Testing (Issues, Class Testing, Object-Oriented Integration Testing)
- Selected Research Topics (e.g. Testing Hybrid Systems)

Learning Outcomes:

Thorough understanding of testing as a method to validate software systems; ability to test systems; understanding the levels of testing; critically evaluate and select software test scenarios; problem analysis.

Transferable Skills:

Analysis of systems; systematic design of experiments.

Course Text:

P. C. Jorgensen, *Software testing*, CRC Press.

Code:	CS-M84
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. Roggenbach
Incompatible:	CS-364
Assessment:	30% computer based and written coursework, 10% report on testing a given system, 60% written examination (in May/June)

MODELLING AND VERIFICATION TECHNIQUES

Code:	CS-M85
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. Berger
Assessment:	30% computer-based and written coursework, 70% written examination (in January)

Synopsis:

This module will give an overview of the landscape and the state of the art of current modelling and verification techniques. One particular tool for software verification will be studied in depth. Students will gain hands-on experience in using that tool.

Syllabus:

Overview of techniques for formal verification.
Interactive theorem proving, automated theorem proving and model checking.

Introduction to one specific logic for modelling and verification.

Techniques for modelling of software using verification tools.

Practical verification of software examples.

Learning Outcomes:

The students will have gained an overview of the landscape and the state of the art of current modelling and verification techniques. They will have acquired a deep understanding of one particular verification tool and know how to translate practical and mathematical problems into its language. They will have obtained hands-on experience in practical verification.

Transferable Skills:

Problem solving, especially abstract modelling and formal reasoning. General mathematical discipline.

Reading:

Jean-Francois Monin and M. G. Hinchey, *Understanding formal methods*, Springer, 2003.

R. D. Tennent, *Specifying Software. A hands-on introduction*, Cambridge University Press, 2002.

Further recommended texts regarding the particular verification tool will be set by individual members of staff.

COMPUTER GRAPHICS, MRes VISUAL COMPUTING PROJECT

Synopsis:

Within one month of the commencement of the course, the student will decide upon a topic of research in discussion with their supervisor. The student will evaluate current research and propose their own work programme based on the contribution they will make. Students will attend regular progress meetings with their supervisor, submit all deliverables, and submit their thesis in accordance with the University Regulations no later than September 30th.

Syllabus:

This module will give students experience of working independently and in depth on a Graphics related project. Following approval of a project plan, the student will carry out the research programme including the following: attendance at progress meetings with supervisor(s), submission of interim research progress report(s), preparation of a journal style paper or poster, presentation of their research at a departmental conference and submission of their thesis in accordance to University regulations.

Learning Outcomes:

Students will:

- have a great depth of knowledge in a complex and specialised area;
- be working at the current limits of theoretical or research understanding;
- be able to make confident decisions about research direction and the tools to use for the job;
- be able to synthesise ideas and create responses to problems that expand existing knowledge and is able to develop new approaches in new situations;
- be able to independently evaluate and accurately report on their own or others work.

Transferable Skills:

Autonomous use of resources; self-directed learning; can isolate, assess and resolve problems of all degrees of predictability, can engage in a full and professional manner with other researchers in the area.

Course Texts:

ACM Transaction on Graphics

ACM Transactions on Graphical Tools

Computer Graphics Forum

IEEE Computer Graphics and Applications

Computer Vision and Image Understanding

and other journals and conference proceedings (including SIGGRAPH and IEEE Visualisation).

Code:	CS-M87
Credit Points:	120
Taught:	Whole session
Contact Hours:	Regular meetings with project supervisor
Co-ordinator:	Dr. M. W. Jones
Assessment:	100% dissertation and viva
	Only available to those students studying for the MRes in Visual Computing

EMBEDDED SYSTEMS

Code:	CS-M88
Credit Points:	10
Taught:	Semester 1
Contact Hours:	20 laboratory classes, 10 lectures
Lecturer:	Dr. Beckmann
Incompatible:	CS-368
Assessment:	10% course work, 15% team laboratory exercises, 35% individual lab book, 40% written examination (in January) (This module is restricted to up to 20 students; priority is given to MSc students on the Safe and Secure Systems pathway)

Synopsis:

Embedded systems are information processing systems embedded into enclosing products such as cars, telecommunication or fabrication equipment. They are essential for providing ubiquitous information, one of the key goals of modern information technology.

The aim of this course is to discuss selected topics of embedded system design.

The lab provides hands-on experience in the design of embedded systems, based on the Lego-Mindstorms kit.

Awareness of logical concepts (propositional logic, first order logic) will help the understanding of this module.

Syllabus:

The lectures discuss selected techniques in their specialisation to the design of embedded systems such as:

- Requirements, Specification and Modeling
- Programming-language-level description techniques.
- Hardware (Sensors, actuators, processors)
- Operating systems, middleware, scheduling
- Hardware/software partitioning and codesign
- Simulation, testing and verification techniques

The labs consist of a series of experiments using the Lego Mindstorms Kit that give the students hands-on experience in typical design challenges in embedded systems. Possible topics include examples from

- control theory
- real time systems
- discrete control
- fault tolerance
- distributed algorithms.

Learning Outcomes:

Students will understand and be able to apply engineering principles for system design and their specific application in embedded systems. They will be able to cope with various methods for specification/modelling, analysis, design, implementation and verification.

Transferable Skills:

Understanding of computer programs as part of socio-technical systems; problem solving; demonstration, presentation and writing skills; team-working.

Reading:

P. Marwedel, *Embedded System Design*, Springer, 2006.

ADVANCED DATABASE SYSTEMS

Synopsis:

The aim of this module is to extend the students' understanding of the principles and techniques used in the design and implementation of modern database systems.

Syllabus:

Enhanced E-R modelling

Advanced SQL; The SQL Standard and its extensions.

Object-oriented and Object-relational databases.

Databases for XML and XML query languages.

Data warehouses and data mining. OLAP.

Query processing and query optimisation.

Web Technology and DBMSs.

Learning Outcomes:

After completing this module students will be able to:

- describe, discuss, compare and contrast various kinds of DBMS including relational, object-oriented, object-relational, multi-dimensional, deductive, real-time and XML;
- design database schemas for different types of database;
- write database queries which use extensions to SQL.

Transferable Skills:

Problem identification, problem analysis and abstract modelling. Ability to learn and use computer systems and software packages effectively, and to evaluate and deploy new technologies.

Reading:

T. Connolly & C. Begg, *Database Systems*, 5th Edition, Pearson Education, 2010.

Code:	CS-M89
Credit Points:	10
Taught:	Semester 2
Contact Hours:	20 lectures
Lecturer:	Dr. Stein
Co-requisite:	CS-M59
Assessment:	20% coursework, 80% written examination (in May/June)

HUMAN COMPUTER INTERACTION PROJECT DEVELOPMENT

Code:	CS-M90
Credit Points:	10
Taught:	Semester 2
Contact Hours:	15 - 20 seminars and discussions
Co-requisite:	CS-M00
Coordinator:	Dr. Wilson
Incompatible:	Other project specification modules
Assessment:	40% essay on HCI issues in focus domain, 50% pilot study report 10% a single presentation (rotated each class)

Synopsis:

In this module students will be presented with an overview of the HCI research area. They are introduced into the topic, the background and the aims of their project. They write a detailed specification which will be the basis of their research project. Guidance as to appropriate research methodologies is provided.

Syllabus:

Students will gain an overview of methodologies used for the development of HCI applications. They will be introduced by their supervisor into a specific research area related to their project. They will learn how to present their research in the form of oral and written work.

Learning Outcomes:

Students will have

- an understanding the foundations of HCI as a specialism
- an understanding of current research issues in HCI
- the ability to select and use appropriate research methods for HCI
- planned a HCI MSc Dissertation project

Transferable Skills:

Problem solving, literature searching, project planning and time management, and report writing.

Reading:

Appropriate texts will be recommended by supervisors

SOFTWARE ENGINEERING PRINCIPLES

Synopsis:

Students will be introduced to the principles of software development and the main professional issues associated with its practice. They will also develop a significant piece of software in teams.

Syllabus:

Introduction to the principles of professional software development.

Software Process Models (e.g. waterfall, spiral).

Software Project Management. Project planning, tools and techniques for planning.

Project conduct, time management, risk management, and team working.

Review of legal, ethical, social and professional issues applicable to the computer industry.

Quality Assurance and Testing.

Team work.

Learning Outcomes:

Students will have:-

- appreciation of the different approaches to software development;
- become fully aware of the major professional issues associated with software development;
- worked together on a significant software project;
- become fully familiar with many aspects of managing software projects;
- knowledge of quality assurance and testing methods for software systems.

Transferable Skills:

Problem solving, project planning, time management and risk assessment, team work, report writing.

Reading:

I. Sommerville, *Software Engineering: International Version*, 9th Edition, Pearson Education, 2011.

Code:	CS-M94
Credit Points:	10
Taught:	Whole Session
Contact Hours:	10 lectures
Lecturer:	Dr. Mora
Assessment:	40% team project, 60% written examination (in May/June) (This module is normally only available to students on MSc Computer Science)

**MSc MODULES FROM
OTHER
COLLEGES/DEPARTMENTS**

THINKING ABOUT DIGITAL MEDIA

Synopsis:

Through close analysis of digital media/new media texts, this module will examine the theoretical and practical issues of new media technologies and the implications for social, political and economic spheres.

Syllabus:

1. Introduction: Digital Media Core Texts
2. Prehistory of New Media Theory: Lippman, Dewey, Walter Benjamin and Frankfurt School
3. Prehistory of New Media Theory: McLuhan and Toronto School
4. Prehistory of New Media Theory: Williams, Hall and the Cultural Turn
5. Prehistory of New Media Theory: Postmodernisms
6. Medium Theory: Kittler, Fuller
7. Cyberspace: Turtle, Rhinegold, Harraway
8. Political Economy: Marx, Hardt & Negri, Dyer-Witherford
9. Discourse: Fairclough, Foucault, Derrida
10. The Information Society. Castells, Pew Internet Research, Oxford Internet Surveys, Margetts
11. Politics and the Web: Coleman, Dutton, Giddens

The course will explore the theoretical themes outlined above, drawing on differing textual examples and referring to key theorists in each case. Case study texts and other media forms are provided in each case, and these will form the basis for group discussion.

Learning Outcomes:

At the end of this module, students should have:

1. An advanced awareness of the debates surrounding digital/new media.
2. An advanced understanding of theoretical issues and discussions.
3. Advanced knowledge of the key theorists and their historical context.
4. Advanced knowledge and appreciation of key media texts.
5. An advanced awareness of the comparisons and contrasts that emerge in the discussion of media and medium theory.

Transferable Skills:

At the end of this module, students should:

1. have developed advanced organizational skills through the management of personal study and working to deadlines.
2. have developed advanced communication skills through essay writing and seminar participation.
3. have built on the skills of synthesis developed at undergraduate level.
4. have built on theoretical and methodological skills developed at undergraduate level.
5. have developed skills of detailed analysis appropriate to level 4.

Reading:

For a complete reading list see the online module catalogue.

Code:	MSDM01
Credit Points:	20
Taught:	Semester 1
Contact Hours:	22 lectures
Lecturer:	Mr. Merrin
Assessment:	100% essay

UNDERSTANDING CRIME

Code:	ASCM17
Credit Points:	20
Taught:	Semester 1
Contact Hours:	20 lectures
Lecturer:	Prof. P. C. Raynor
Assessment:	100% assignment

Synopsis:

This module aims to introduce students to the main explanatory approaches used in criminology, with an appraisal of the aspirations, scope and limitations of each. The selection of explanatory approaches is based on their relevance to current debates and their potential uses or misuses in arguments about criminal justice, policy and practice: thus the focus is not on criminology as an

introduction to social theory, but rather on how criminologists have used different paradigms of explanation, what insights these have generated, how they have informed research and debate concerning crime and the extent to which they have received empirical support..

Syllabus:

1. Understanding crimes: facts or social constructs?
2. Strain theories and subcultural explanations of offending
3. Control theories and rational choice: why not offend?
4. Crime, social integration and informal controls
5. Individual difference, gender and criminal careers
6. Combined and eclectic explanatory approaches: the 'new realism' in criminology
7. Integrated approaches to the effects of 'criminogenic need'
8. Environmental criminology: the distribution of crime
9. Criminalisation, power and the State
10. Criminology, policy and practice in the criminal justice system

Learning Outcomes:

This module will enable students to identify the theoretical basis, if any, of policies or strategies adopted or advocated in criminal justice; explain the relevance, strengths and weaknesses of explanatory approaches in a given context; make a critical appraisal of 'common-sense' or popular beliefs about crime and punishment in the light of evidence and well-grounded theory; identify and appraise the theoretical basis of their own practical work (if they are practitioners) or that of others (if they are not practitioners).

Transferable Skills:

In this module, the student will acquire personal transferable skills in the following areas: developing arguments; assessing arguments against evidence; determining what kinds of evidence are relevant; distinguishing between alternative approaches to a problem; discussing alternative explanations and related policy options; planning the written presentation of arguments and evidence and presenting arguments and evidence in writing.

Reading:

M. Maguire, R. Morgan and R. Reiner, *The Oxford Handbook of Criminology*, 4th Edition, Oxford, 2007.

URBAN NETWORKS AND THE KNOWLEDGE ECONOMY

Synopsis:

The module is comprised of two sections, each involving five two-hour, student-centred seminars in which the issues from the directed reading assigned to students are presented and discussed. The first half of the module introduces postgraduate students to a wide range of global cities literatures; to demonstrate the importance of cities in the context of debates about globalization. The sessions address five key questions regarding global cities: (1) What exactly constitutes a global city? (2) How are cities related to each other in a globalizing world? (3) How are some cities trying to achieve global city status? (4) How does globalization affect different cities differently? (5) What are the main theoretical approaches to conceptualising cities in globalization? The second half of the module explores the concept of the knowledge economy and demonstrates how the ability to generate and utilize knowledge has become crucial to economic growth and development. The demands of the knowledge economy have reshaped the economic geography of winners and losers, and the reasons for this uneven geography are explored through the concepts of agglomeration, clusters the creative class and social capital, and in the context of city and regional case-studies. The final session explores the implications of the knowledge economy for policy-making and asks what lagging regions and less-favoured cities can do to engage with the knowledge economy.

Code:	<i>GEGM13</i>
Credit Points:	<i>20</i>
Taught:	<i>Semester 1</i>
Contact Hours:	<i>20 lectures</i>
Lecturer:	<i>Dr. K. G. Rees</i>
Assessment:	<i>Refer to the Geography Department</i>

Syllabus:

The module is comprised of two sections, each involving five two-hour, student-centred seminars. The first half of the module introduces postgraduate students to a wide range of global cities literatures; to demonstrate the importance of cities in the context of debates about globalization. The second half of the module explores the concept of the knowledge economy and demonstrates how the ability to generate and utilize knowledge has become crucial to economic growth and development.

Global Cities: Five sessions seeking answers to five key questions:

- (1) What exactly constitutes a global city?
- (2) How are cities related to each other in a globalizing world?
- (3) How are some cities trying to achieve global city status?
- (4) How does globalization affect different cities differently?
- (5) What are the main theoretical approaches to conceptualising cities in globalization?

The knowledge economy: Five sessions comprising:

- (1) Pinning down the knowledge economy? The form and function of knowledge in economic development
- (2) What makes some cities so innovative?: creative cities and the 'creative class':
- (3) Explaining and promoting high-tech clusters: urban and industrial agglomeration
- (4) Social approaches to innovation: Social Capital and Communities of Practice
- (5) Breaking into the knowledge economy: the prospects for lagging regions.

Learning Outcomes:

At the end of this module students should be able to:

1. discuss alternative theoretical approaches to conceptualising cities in globalization

2. understand how cities are increasingly related to each other in a globalizing world
3. demonstrate an understanding of how 'global city status' is a political project
4. evidence an empirical knowledge of how globalization affects different cities differently
5. identify different forms of knowledge and discuss their relevance to the geography of innovation
6. understand how agglomeration economies affect the innovative capacity of cities and regions.
7. assess the utility of the industrial cluster concept for understanding and influencing the contemporary geography of innovation.
8. evaluate the concepts of 'creative cities' and a 'creative class' in the knowledge economy
9. discuss the value of a sociological appreciation of innovation and local/region competitiveness
10. discuss policy alternatives and evaluate economic development strategies available to or adopted by lagging regions in order to engage with the knowledge economy.

Transferable Skills:

1. Critical analysis and evaluation of diverse theories and concepts
2. Literature search and review skills
3. Oral presentational and debating skills
4. Written skills: organization, discussion, developing an argument, clarity, referencing.
5. Organizational skills
6. Ability to integrate and synthesize diverse literatures.

Reading:

S. Sassen, *Global Networks-Linked Cities*, Routledge, 2002.

P. Taylor, *World City Network*, Routledge 2004.

T. Bender and I. Farias, *Urban Assemblages: How Actor-Network Theory Changes Urban Studies*, Routledge, 2009.

P. N. Cooke, *Knowledge Economics: Clusters, Learning and Cooperative Advantage*, Routledge, 2002.

B. Asheim, P. Cooke and R. Martin, *Clusters and Regional Development: Critical Reflections and Explorations*, Routledge, 2006.

R. Florida, *Cities and the Creative Class*, Routledge, 2005.

CONTRACT AND E-COMMERCE LAW

Syllabus:

1. The nature of contract: the structure of agreement. Non-disclosure agreements.
2. Application of the existing law to E-Commerce: Contractual issues, domain names, copyright and other intellectual property issues, evidential issues, security.
3. Deal structures.
4. Control of the internet and E-Commerce

Code:	<i>LALM121</i>
Credit Points:	<i>10</i>
Taught:	<i>Semester 1</i>
Contact Hours:	<i>20 lectures</i>
Lecturer:	<i>Dr. J.P. Howden-Evans</i>
Assessment:	<i>100% examination (in January)</i>

Learning Outcomes:

At the end of this module students will be able to:

1. Understand and critically evaluate the role of contract law within a business environment.
2. Analyse and critically evaluate at a high level the application of existing law in e-commerce.
3. Critically understand the regional and international regulatory context of e-commerce.
4. Critically assimilate and evaluate deal structures in the context of contract and e-commerce law.

Transferable Skills:

Students should be able to make proficient use of word-processing and make effective use of the internet and other IT resources in their studies. Students should be motivated to learn and able to act independently in planning and managing tasks with limited guidance in areas which they have studied.

RETHINKING GLOBAL POLITICS

Code: PO-M52

Credit Points: 20

**For details of this module consult the
Department of Politics.**

E-BUSINESS

Syllabus:

E-Business Environment - Economic actors and relationships, electronic markets and hierarchies, internal and external value chains for business;

E-Business - definitions of e-Business, e-Commerce and I-Commerce, forms of e-Business, contribution to internal and external value chains; E-Commerce - features and type of commerce, B2C, B2B and C2C e-Commerce, benefits and problems of e-Commerce;

Intra-Business E-Business - information systems and the value chain, intranets, benefits of intra-business e-Business;

B2C E-Commerce - the customer chain, forms of B2C e-Commerce, benefits of B2C e-Commerce, e-Marketing;

B2B E-Commerce- the supply chain, forms of B2B e-Commerce, extranets, benefits of B2C e-Commerce, e-Procurement;

E-Business Infrastructure - informatics infrastructure, social and technical infrastructure, electronic delivery of services and products;

Technical Infrastructure - information, information systems, information and communications technology;

Social Infrastructure - planning, strategy, management and development for e-Business;

Supporting E-business - the relationships between the technical and social infrastructure of e-business.

Learning Outcomes:

Students successfully completing this module should be able to:

Distinguish between e-Business and e-Commerce;

Demonstrate an understanding of the history of the Internet, and the significant events in its development that resulted in the emergence of e-commerce;

Appreciate the impact of Internet/Intranet/extranet technologies on business from both strategic and operational viewpoints;

Appreciate the strengths, weaknesses, opportunities and threats posed by the emergence of e-commerce;

Recognise the need for effective strategic and operational planning in the formulation and implementation of e-commerce systems;

Be able to provide examples of the differing ways in which e-commerce may provide businesses with competitive advantage and the lessons which can be learnt from successes and failures.

Transferable Skills:

Problem solving, Analysis, Design, Critical Evaluation, Effective communication, Improving own learning and performance.

Code:	<i>EBCM10</i>
Credit Points:	<i>10</i>
Taught:	<i>Semester 2</i>
Contact Hours:	<i>22 lectures</i>
Lecturer:	<i>Prof. M. D. Williams</i>
Assessment:	<i>(in June) Refer to the Department of Business for details</i>

ITWALES

INTRODUCTION

ITWales is the Department of Computer Science's own programme for industrial interaction. It is intended to have a wide spectrum of activities and be proactive and flexible. It plays a role in nurturing and supporting the Welsh IT industry. The programme is the principal way the Department pursues its third mission. ITWales is a remarkable success story. It began in 1993 in a small way with a student placement programme.

The ITWales team is supported strategically by a small Steering Committee of computer scientists, chaired by Prof. Tucker.

Our programme is strongly supported by companies throughout South Wales, especially local small and medium sized companies and also major international IT companies. It brings many opportunities and benefits to our students.

ITWales has developed a long track record of developing projects to link computing academics and computing students with business and industry to the mutual benefit of all parties. The projects and programmes delivered by ITWales all aim to support and bolster the ICT sector as well as raising the ICT skills level within businesses in general.

Currently ITWales has developed the following major programmes:-

- Student placements
- itwales.com
- Software Alliance Wales. www.softwarealliancewales.com
- Technocamps. www.technocamps.com

Student and Graduate Placement schemes

The ITWales placement scheme which aim to help students be aware of the advantages and challenges of working in industry and commerce, and to encourage them to take up and develop careers and opportunities in local small to medium sized enterprises (SMEs). The scheme also aims to help companies develop and enhance their IT resources, and assess, at limited cost, the benefits to be gained by recruiting a highly skilled graduate workforce.

itwales.com

itwales.com is an online technology magazine for business produced by ITWales, which delivers the latest news, technology reviews, interviews with leading figures in IT, and features on specific important topics.

Software Alliance Wales

Software Alliance Wales is developing a network for Software Developers throughout Wales which includes both ICT professionals and academics. This network allows developers across the region to collaborate and share best practice, and to promote Wales as a key driving force in Software Development.

Technocamps

An outreach programme for schools and colleges to encourage more pupils to study computing at all levels, GCSE, A level and HE.

STUDENT AND GRADUATE PLACEMENTS

Period:	<i>Part time Term Time & full time during Summer Vacation</i>
Commitment:	<i>6 - 10 week industrial placement</i>
Co-ordinators:	<i>Mrs. Nash, Mrs. Birdsall</i>
Assessment:	<i>Reports by the company and the student.</i>

Synopsis:

The objective of the placement programme is to expand the existing IT skills base within businesses in Wales. It aims to help companies enhance their IT resources, processes or products and allow them to assess, at limited cost, the benefits of employing a highly skilled graduate workforce. Students are also given a taster of the challenges awaiting them when they enter the employment market and are encouraged to consider careers and opportunities.

Description:

Graduates and undergraduates of EU citizenship who are either studying or have completed an IT related degree are employed by Welsh companies for up to ten weeks in the summer vacation period. The company produces a schedule of work and selects the placee through interview. During the placement a visit is made to the company by a member of staff to monitor progress. Evaluation forms are completed by both student and company at the end of the placement. The company is then in a position to evaluate the benefits of employing graduate staff, and may, if appropriate, be able to offer employment to the student who will already be well acquainted with the company's operation.

Under the current scheme small and medium sized companies based in Wales can qualify for a subsidy in the cost of the student wages. The subsidy is met by a grant from HEFCW under the Go Wales Programme. Larger companies also participate but they are required to pay the wage in full.

The benefits to the student include:

- “hands on” experience of work outside the University environment;
- the opportunity to use their technical skills on real world problems;
- an introduction to a company with the possibility of future employment;
- an opportunity to obtain remuneration for the application of IT skills—minimum wage of £250 per week;
- an opportunity to understand the performance requirements of full time employment.
- an opportunity to gain a City and Guilds Professional Development Award in Quality Work Experience.

The benefits to the company include:

- high level technological skills at relatively low cost which might not otherwise be available;
- the opportunity to assess, with limited cost and in the working environment, a potential graduate employee;
- the opportunity to enhance links between the company and the University leading to the dissemination of “state of the art” knowledge into the participating companies.

ITWALES.COM

Synopsis:

itwales.com is the online IT magazine for business in Wales. Produced monthly, with daily news updates, it contains ICT news, profiles of Welsh companies, interviews with key players in the field

of technology, reviews of the latest software and hardware, and useful articles on both basic and advanced ICT topics. As an expanding resource as well as a topical magazine, itwales.com is created with the specific purpose of encouraging Welsh business to expand, innovate and be part of the information age. It is aimed both at companies in Wales wanting to improve their ICT operation, and companies outside of Wales wanting to discover more about the ICT skills base in Wales.

Period: *Monthly editions with daily news*

Co-ordinator: *Mrs. Earls*

Description:

itwales.com was officially launched in March 2002 by Andrew Davies, former Minister for Enterprise, Innovation and Networks, Welsh Assembly Government. The magazine averages around 35,000 unique visitors per month, and has international readership. Example topics featured in itwales.com include:

News:

Global IT company launches national centre of excellence for public sector services in Wales (June 2009); Swansea technology company set to save automotive industry millions (April 2009); IT will be at the heart of the next Industrial Revolution (March 2009); National Library undertakes major digitisation project (October 2008); Report says mobiles should do more to be green (December 2008); One year on, IT professionals less concerned about organisations' environmental impact (July 2008); Cardiff company launch 'Good Broadband' guide in support of Ofcom code of practice (June 2008);

ITWales Interviews:

Dr Lyn Evans, LHC Project Leader, CERN (April 2009); Andy Robinson, Police Manager for e-Crime Wales (September 2008); Mobile Security, Symantec and Trend Micro (February 2008);

Richard Hollis, IT Security expert (December 2007).

Features:

Website aesthetics - what has it got to do with usability? (June 2009); How to make the most of your website in a recession (May 2009); An Expert's Guide to Open Source Software Security (February 2009); From Bar Codes to RFID Part 1 (November 2008); E-Business Support: New opportunities for Welsh Business (October 2008); Security of personal information—a guide for SMEs (June 2008); Customisable websites - the definitive guide (June 2008).

Profiles:

Quote Exchange (May 2009); Dezrez Services Ltd (August 2008); Marda Associates Ltd (July 2008); Accelerio Digital Solutions Ltd. (May 2008).

Reviews:

Creating a Web Site: The Missing Manual (May 2009); Head First - Rails (April 2009); Head First - PHP & MySQL (February 2009); Making your own portable applications (July 2007).

Software Alliance Wales

Period:	<i>All year</i>
Co-ordinator:	<i>Shamala Govindasamy</i>

The Software Alliance Wales (SAW) programme is a £13million pan Wales project. It is supported financially by an £8million grant from the European Social Fund.

The project which started on 1st January 2009 will run to 31st December 2014. Software Alliance Wales is led by ITWales at Swansea University in partnership with the Departments of Computer Science in the Universities of Bangor, Glamorgan, Aberystwyth and the University of Wales Trinity Saint David.

SAW has been developed to provide specific support to the software sector, to address skills requirements and to respond to demands to help Welsh Businesses to maximise the potential of exploiting new technologies.

Software Alliance Wales will develop a network for Software Developers throughout Wales which includes both ICT professionals and academics. This network allows developers across the region to collaborate and share best practice, and to promote Wales as a key driving force in Software Development.

Software Alliance Wales has four main strands to the programme:-

- Increase awareness and understanding of ICT within Welsh SME's through Business ICT Workshops.
- Providing a Continuous Professional Development programme for ICT Professionals
- An accreditation scheme for ICT Companies.
- It will also broker links between businesses and academia through Student ICT Development Projects

SAW DELIVERABLES

Business ICT workshops

SAW supports the need for increasing higher level ICT skills across all business and industry sectors by providing tailor-made, relevant, 'bite-size' development sessions, organised as workshops or seminars to address particular issues identified within the SME community.

The sessions, which provide regular up to date information on emerging opportunities and issues and learning opportunities on the use of ICT, are organised and delivered in locations throughout Wales.

All events provide business with opportunities to network and collaborate with academia and partner organisations such as the British Computer Society.

For further information please contact Dr Maria Moller on 01792 606659 or m.moller@softwarealliancewales.com

CPD Training for ICT Professionals

Continuing professional Development (CPD) is the organised continuation, improvement and broadening of knowledge, understanding and skills, as well as development of personal qualities necessary for an individual to maintain their competence in order to undertake their duties throughout their working life.

Because technology is developing and expanding so rapidly it is increasingly important that professionals employed in the ICT industries regularly review and identify what activities and skills are required to raise their individual levels of competence. This is an effective method of improving knowledge and skills for business and industry in general but on an individual basis offers competitive potential in the market place.

Software Alliance Wales provides access to highly specialised Level 4 CPD training, specific to the software sector. All training is highly specialised and is determined by and specific to the industry's requirements and will change through the lifetime of the project as new requirements in this rapidly changing technology field emerge.

Predominately delivered through short courses the skills developed are recognised by the industry and are highly transferable.

To find out about the CPD programme and the availability of courses, please contact Dr Matt Roach on 01792 606662 or m.roach@softwarealliancewales.com.

ICT Company Certification scheme

A major aspect of the SAW project is to establish an Accreditation Centre in Swansea.

Working with Accredited UK and the Welsh Government, the Accreditation Centre will encourage, guide and support ICT companies in Wales to gain Accredited UK certification, a quality standard recognised UK wide that has been specifically designed to prove the capabilities of ICT Solution providers. (<http://www.accredituk.com/>)

To find out about the ICT Accreditation Scheme please contact Dr Matt Roach on 01792 606662 or m.roach@softwarealliancewales.com

Student Development Projects

Student Development Projects are designed for students to complete their final year or MSc projects with direct industrial applications, providing them with the opportunity of in-depth exploration of a particular topic in computer science, and the opportunity of designing and implementing a complete system and experiencing the major phases of the life-cycle of a software project.

How does it work?

- The company contacts the SAW project with an initial idea.
- A meeting with the company and academics from the Department of Computer Science is organised when the idea is fully explored, a project specification is developed that includes the aims and objectives, the structure and timing and the required hardware and software resources.
- The project is then allocated to a student with relevant technical skills and a member of the academic staff specialising in the subject area concerned will supervise the project.

The Department will ensure that such a project will be allocated to a student with a sound academic record and adequate technical skills. The project will also be supervised by a member of academic staff specialised in the concerned subject area. In addition to the project milestones listed in the relevant project modules, weekly meetings between a student and his/her supervisor, an industrially related project will normally involve periodic meetings of the student, company and the University staff.

Because of the technical competence of our students and the high standard of their project work, such projects often form a valuable task in the strategic development of the company. These may be specification, prototyping or evaluation exercises, or may constitute tasks that would be otherwise unrealistic or impractical to conduct with the company resources.

To find out about Student Development projects please contact Dr Maria Moller on 01792 606659. or m.moller@ofwarealliancewales.com

TECHNOCAMPS

The Technocamps project has been devised and developed by staff from the Department of Computer Science at Swansea University.

Period:	<i>All year</i>
Co-ordinator:	<i>Mr S. Toomey</i>

A £6 million project led by Swansea University in partnership with the Universities of Bangor, Aberystwyth and Glamorgan, Technocamps will provide daily and weekly sessions to pupils aged 11-19 on a range of exciting topics such as robotics, games development, animation, digital forensics, software development and much more, to inspire them to consider the computing-based topics underpinning the STEM subjects (Science, Technology, Engineering and Maths).

The key objectives of Technocamps are

- To inspire and technically empower young people interested in computing science and STEM subjects
- To encourage them to consider the challenge of a career in this area
- To give them an opportunity to spend time in a learning environment within HE, meet academics and students and break down any barriers that would otherwise stop them taking their studies further

The work of Technocamps will support and enhance the ICT and Computing curricula taught in schools and colleges. The project does not take the place of teachers, but takes a creative approach to Computing and the STEM subjects.

Once the Technocamps workshop is over, teachers can access all the tools needed to establish and run a Technoclub within their school or college to further inspire students and encourage them to work together. Technoclubs will be encouraged to communicate, compete and work together via the Technocamps website, where more teaching resources will also be available.

The project will provide an opportunity for pupils to gain a real insight into the practical application of their learning, and let them understand potential career opportunities through 'virtual' work experience placements and specifically customised sessions.

Girls are under represented within the technology sector, so Technocamps will seek to address this problem by delivering sessions specifically targeted at the needs and wants of girls. The project will also work with students that have been categorised as NEET (Not in Employment, Education or Training) as well as those that are at risk of falling into that category.

Technocamps has been designed to nurture and inspire future generations of highly skilled and enthusiastic digital natives, who will not only develop and supply our ongoing science and technology needs but could also make Wales a world leader in the development of new and advanced technologies.

OTHER INFORMATION

GENERAL READING LIST

Computer Science is at the heart of revolutionary changes in science and engineering, medicine, design and manufacture, commerce and financial services, government and public services, defence, transportation, media and communication, and at home. Students immersed in the technical world of computer science do not usually find it easy to develop an overview of the subject, its applications and influence on society.

The following is an annotated list of references which are accessible to all students of Computer Science with as yet little or no training, and which contain material which is helpful to the understanding of the subject and its role in the world.

The exploration of the subject is a lifetime's work.

General

1. A. K. Dewdney, *The New Turing Omnibus: 66 Excursions in Computer Science*, W H Freeman, 1993.
66 easy-to-read "excursions" into 66 of the main points of interest in the subject.
2. V. Illingworth and I. Pyle (eds), *Dictionary of Computing*, 4th Ed, Oxford University Press, 1996.
An established reference on computing, containing entries explaining terms from across the subject.
3. J. G. Brookshear, *Computer Science: An Overview*, 9th Ed, Addison-Wesley, 2007.
This text provides a very broad overview of the subject, touching on the subject matter of much of the computer science curriculum.
4. J. Bentley, *Programming Pearls*, 2nd Ed, Addison-Wesley, 2000.
A compendium of 13 essays, written for programmers, which develop insightful solutions to various programming tasks.
5. H. Thimbleby, *Press On*, MIT Press, 2007.
An introduction to how to program computers so that they are easy to use; awarded Best computer and information sciences book award by the American Publishers Association.

Software Engineering

5. N. Leveson, *Safeware: System Safety and Computers*, Addison-Wesley, 1995.
A study of the causes, and methods for prevention, of system failures, through the analysis of various famous major accidents caused by system failure.
6. P. G. Neumann, *Computer Related Risks*, Addison-Wesley, 1995.
A compendium of computer mishaps, a study of their causes, and discussion of possible technologies for preventing similar mishaps.
7. S. Baase, *A Gift of Fire*, 2nd Ed, Prentice-Hall, 2003
Social, legal and ethical issues for computers and the Internet. It covers the challenges and implications of computer technology - and the responsibilities of professionals who design and use computer systems.

Algorithms for Problem Solving

8. D. Harel and Y. Feldman, *Algorithmics: The Spirit of Computing*, 3rd Ed, Addison-Wesley, 2004.
A readable presentation of the design of algorithms for problem solving, including their efficiency and inherent limitations.

9. D. Harel, *Computers Ltd*, Oxford University Press, 2000.
A small book describing as non-technically as possible the limitations of computers, the sorts of problems that you might think they can solve but cannot, and why they can't.

Logic and Discrete Mathematics

10. J. Barwise and J. Etchemendy, *Language, Proof and Logic*, CSLI Publications/Seven Bridges Press, 1999.
An elementary introduction to syntax (formal language) and semantics (meaning) in the framework of mathematical logic; comes with software that allows the reader to explore the material and test their intuition and understanding in an interactive mode for self-study.
11. R. Fagin, J. Y. Halpern, Y. Moses and M. Y. Vardi, *Reasoning about Knowledge*, MIT Press, 1995.
A highly readable introduction to one of the most exciting themes in (applied) logic and informatics: how formal logic can be applied to express and reason about statements not only about "facts" but also about one's knowledge of facts and other's knowledge,...
12. D. R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*, 20th-Anniversary Edition, Penguin, 2000.
An entertaining presentation of the logical underpinnings of computers and intelligent behaviour.
13. R. Smullyan, *Forever Undecided: A Puzzle Guide to Gödel*, Oxford University Press, 1987.
A playful presentation of some of the most important, and complex, 20th-century discoveries in logic which have profound impact on Computer Science and Artificial Intelligence.
14. D. J. Velleman, *How To Prove It: A Structural Approach*, Cambridge University Press, 1994.
An introduction to the basic mathematics and logic needed in Computer Science, presented in the style and rigour of "structured" programming.
15. COMAP (The Consortium for Mathematics and Its Applications), *For All Practical Purposes: Mathematical Literacy in Today's World*, 5th Ed, W. H. Freeman and Company, 2000.
A hefty text covering a large number of basic mathematical topics, most of direct relevance to computer science, presented at an accessible level, and backed up by numerous exercises and simple programming projects, as well as an extensive professionally-designed web site.

Computer Industry and Applications

16. R. X. Cringel, *Accidental Empires*, Penguin, 1996.
Informative and amusing account of the American PC industry.
17. G. W. Flake, *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*, MIT Press, 2000.
An eye-opening account of instructive and exciting applications of computer science.
18. K. Hafner and M. Lyon, *Where Wizards Stay Up Late: The Origins of the Internet*, Simon & Schuster, 1998.
The remarkable 30-year story of the internet.

19. W. J. Kaufmann III and L. L. Smarr, *Supercomputing and the Transformation of Science*, Scientific American, 1993.

A presentation in Scientific American's illustrative style of the capabilities of the world's most powerful computers.

History

20. M. Davis, *The Universal Computer: The Road from Leibniz to Turing*, Norton, 2000.

A history of the concepts underlying computers, and the scientists who developed them, written in an engaging style.

21. C. and R. Eames, *A Computer Perspective*, Harvard University Press, 1990.

Superb picture-book on the origins and development of information processing, compilers and computing.

22. A. Hodges, *Alan Turing: The Enigma*, Vintage, 1992.

Fascinating biography of an original thinker on computing and computers.

23. S. Singh, *The Code Book: The Secret History of Codes and Code-breaking*, Fourth Estate, 1999.

Entertaining history of the important problem of the encryption of information.

24. M. Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*, MIT Press, 2003

Excellent account of the development of the software industry.

TUTORIALS

Each student whose Home Department is Computer Science is assigned a Personal Tutor and an Academic Tutor. In the case of Level 1 Single-Honours Computer Science Students, the Personal and Academic tutor is normally the same person. For Joint Honours students the Home Department is usually not Computer Science, and their Personal Tutor will usually be an academic member of staff in their other Honours subject.

A student will retain their Personal Tutor for their entire stay within the Department, but their Academic tutor will normally change each year. The Academic Tutor is responsible for supporting your academic progress, and you can normally expect to attend seminars or meetings with your Academic Tutor regularly throughout each Semester. Note that attendance at such seminars and meetings is mandatory.

The Personal Tutor is the person to turn to for help and advice if problems of any kind arise. You can expect the tutor to provide pastoral assistance and guidance to you on issues that may affect your well-being, attendance and progress through University. You are assured that this student-tutor relationship will remain confidential subject to University guidelines and appropriate acts of parliament. Examination results will also be collected from your personal tutor. Your relationship with your Personal Tutor should not be one way and you have responsibilities, which must be met. Should you have, or develop a medical or other problem that has the potential to influence your academic progress you must advise your Personal Tutor immediately to ensure that you receive appropriate support during your time at Swansea. You are expected to attend all meetings as arranged with your Personal Tutor and provide satisfactory explanations for any absence.

The University's Personal Tutor Policy can be found here:

<http://www.swan.ac.uk/registry/academicguide/assessmentandprogress/personaltutoring/#.dn.39419>

In Level 1, because your Personal and Academic Tutor are the same person, you can expect to meet your Personal Tutor weekly, as part of your regular seminars. After Level 1, you can expect to meet your Personal Tutor at least once a term.

Seminars and meetings with your Academic Tutor are an important part of your progress through the degree programme. Among the objectives of these sessions are: 1. To allow the academic staff to get to know the students individually. To hear opinions on aspects of the course, Departmental policies and University and student life. To help the academic staff appreciate a student's academic difficulties and identify problems. To offer opinions, criticism, practical advice and help whenever appropriate. 2. To examine the progress of students' lecture courses and hold discussions, on a regular basis, about specific topics from the lectures. The idea is to support directly the learning of the subjects taught, by motivating topics, clarifying obscure points, and insisting students discuss topics in detail. 3. To discuss written and oral presentation skills. To guide students in their reading and writing, particularly through assessed report writing. 4. To discuss a range of subjects including the Educational Aims and the relationship with the aims of each year of study.

At Level Zero, the year head will normally hold an academic tutorial session for all Level 0 students at least once each term to discuss their progress. At Level One, seminar sessions are held weekly which have the additional role of supporting the module CS-130 Computers and Society: the report and presentation assessed as part of CS- 130 are organised as part of these

seminar sessions.

At Level Two, for students taking the CS-230 and CS-235 Software Engineering modules, fortnightly academic tutorials will be held to discuss all aspects of the course but will also have the additional role of supporting the group coursework assignment. The tutor's primary role will be to ensure that the group is working effectively, and to address organisational problems. In the case of Levels Three and Four, a student's project supervisor will provide general academic support and may hold either individual, or group-based, tutorials. Level Three students are expected to see their project supervisors each week during both semesters, to discuss progress on their project, and any other matters that may arise. For joint honours students who are not taking CS-130, CS-230 and CS-235, CS-344 or CS-354, the corresponding liaison officers normally act as the contact points for any academic problems. It is anticipated that students will meet their tutor at least once each term to discuss their progress. The student's home department will normally appoint a Personal Tutor.

For MSc students, tutorials will normally be held fortnightly during both semesters and have the additional role of supporting the Project Development modules. The tutor will set and assess one report. In addition, group coursework will be undertaken with the tutor acting in a supporting role. Towards the end of Semester 2, a student's project supervisor will take over the main role as academic tutor. Formal tutorials will not normally be held, but students are expected to be in contact with their supervisors regularly (normally once a week) in order to discuss progress on their project, and any other matters that may arise. Other masters students will have regular tutorials and meetings with their project supervisors as required.

ASSESSMENT AND PROGRESSION

BACKGROUND

The University's Assessment Rules (Progression and Award Rules) are well established and are included in the online Academic Guide:

<http://www.swan.ac.uk/registry/academicguide/UndergraduateAwardRegulations/UndergraduateAssessmentRegulations/>.

ASSESSMENT POLICY

One of the objectives of the University's Learning, Teaching and Assessment Strategy is to ensure that assessment is fair. To achieve this, the University has produced an assessment policy, which can be found in the Teaching Quality Manual, available on the Academic Registry website: <http://www.swan.ac.uk/registry/A-ZGuide/T/TeachingQualityManual/>.

ASSESSMENT REQUIREMENTS OF A MODULE

You are strongly advised to take note of the various methods, which the Department has decided to adopt on assessing students and to raise any queries, which you may have with your lecturers early in the session. You should also know in advance whether an essay/practical report would contribute to the overall mark for the module. Please also note any deadlines set by the Department for the submission of work and the consequences of failing to meet them. You are also advised that you are required to complete all elements of a module's assessment pattern.

EXAMINATIONS

The conduct of University examinations is governed by strict rules, set by the University, which include rules on absence from examinations, conduct during an examination, unfair practice and plagiarism. These rules are outlined in detail in the Examination Regulations and Procedures in the online Academic Guide:

<http://www.swan.ac.uk/registry/academicguide/AssessmentandProgress/ExaminationRegulationsandProcedures/>; students are advised to familiarize themselves with these regulations, in particular the following:

All University examinations, including supplementary examinations, shall be sat in Swansea. There shall be three official examination sessions, namely in January (end of first semester), May/June (end of session) and August (supplementary examinations), however some subjects, for example, Medicine and Nursing, may operate their examinations outside these official sessions.

Students must ensure that they are available for the entire duration of the University's examination periods, including where appropriate the supplementary examination period. It is the student's responsibility to check the official examination timetable and to note the dates, times and locations of their examinations.

It is a student's responsibility to familiarize themselves with the regulations relating to examinations and the location of the examination in good time before the exam takes place.

Students are reminded of the following examination procedures:

- Candidates must make sure that Colleges are made fully aware, in writing, of any extenuating circumstances which might have affected their studies and preparatory work leading up to assessment periods. It is imperative that candidates speak as soon as possible to their Personal Tutor or other member of staff in the College in order that the facts can be

brought to the attention of the College Examining Board well in advance of the University Award/Progression Board meetings. Failure to inform their Colleges of their difficulties and to provide evidence in good time before the College Examining Board meeting may well result in the alleged circumstances being disregarded. 'Extenuating circumstances' could include personal or academic-related problems or issues involving difficulty in accessing facilities or materials relevant to the course.

- Candidates are responsible for checking their examination timetables carefully when they are published to ensure that all the examinations that they are due to sit are scheduled. Any problems should be reported to the Examinations Officer, Academic Registry.
- Additional time for any candidate arriving after the start of the examination will not be given.
- Candidates must ensure that they take their Identity Card to the examination venue on all occasions and display it clearly on the desk during the examination. Candidates must also fully complete an attendance slip.
- Students must ensure that they do not engage in any form of unfair practice, whereby they take action which may result in them obtaining for themselves or others, an unpermitted advantage. For instance:
 - No unauthorised material should be taken into the venues
 - The passing of notes or exchange of materials is strictly prohibited
 - No talking or communication with other candidates is permitted
 - Students are only allowed to take permitted aids for their examination into the examination venue. These aids (e.g. pencils, pens) must be carried in a clear and transparent pencil case or "poly pocket" which will be examined on entry to the examination venue
 - Students shall use only the official stationery provided - all rough work shall be done on the stationery provided and handed in with the completed script. No script, rough work or official stationery may be removed from the venue. Students must not have in their possession in the examination room, nor make use of, any book, manuscript, electronic calculator or any other aid which is not specifically allowed in the rubric of the examination paper. If calculators are permitted they must not contain any user-recorded data or program and must be incapable of electronic communication.
- Students who have been observed acting in breach of examination regulations will be given a formal written warning. They will also be warned that if they continue to be disruptive and persist to act in a manner that is likely to disturb further other students they will be required to leave the venue immediately. Such students will not be readmitted for this examination and the incident will be reported to the Superintendent of Assessment.

CYFLWYNO GWAITH YN Y GYMRAEG

Bydd hawl gan bob myfyriwr i gyflwyno gwaith i'w asesu yn Gymraeg beth bynnag fo cyfrwng addysgu'r cwrs hwnnw ac eithrio yn achos modiwlau ym mhynciau iaith lle mae meistroliaeth o'r iaith honno yn un o feini prawf yr asesiad. Yn achos unrhyw waith sy'n ymwneud a chwrs neu fodiwl a addysgir drwy gyfrwng y Saesneg bydd disgwyl i'r myfyriwr roi gwybod ymlaen llaw eu bod yn dymuno cyflwyno gwaith yn Gymraeg, fel y gellir gwneud trefniadau cyfieithu yn ol yr angen.

Rhaid i fyfyrwyr sy'n dilyn rhaglenni a addysgir gyflwyno cais ysgrifenedig i'r Coleg o fewn pedair wythnos i ddechrau'r modiwl(au) perthnasol er mwyn cael cyflwyno arholiadau, traethodau hir neu draethodau ymchwil yn Gymraeg. (Lle bo modiwl yn fyrrach na phedair wythnos o hyd, rhaid cyflwyno'r cais wrth ymrestru ar y modiwl). Bydd y Coleg yn anfon ceisiadau ymlaen at y Cofrestrydd Academiaidd a fydd wedyn yn trefnu i gyfieithu cwestiynau, sgrïptiau a thraethodau hir/ymchwil lle bo angen.

Rhaid i fyfyrwyr gyflwyno eu ceisiadau i'r Coleg 'gartref' gan ddefnyddio'r ffurflen hon:

https://intranet.swan.ac.uk/documents/welsh/Cyflwyno_gwaith_yn_Gymraeg.doc

Bydd y Coleg Gartref yn anfon copiâu o'r ffurflenni ymlaen i'r Gofrestrfa Academaidd (a lle bo'n briodol, Colegau eraill).

Bydd angen i fyfyrwyr sy'n dilyn rhaglenni ymchwil ac sydd am gyflwyno eu traethodau ymchwil yn Gymraeg wneud cais i'r Coleg o fewn tri mis i ymrestru am y tro cyntaf, fel rhan o'r broses o'r gofynion cadarnhau ymgeisyddiaeth. Bydd y Coleg yn anfon copiâu i'r Gofrestrfa Academaidd ar gyfer trefnu i'w cyfieithu.

SUBMITTING WORK IN WELSH

Every student will have the right to present work for assessment in Welsh for a course leading to an initial or higher degree, or diploma or certificate, irrespective of the teaching medium of that course, except in the case of modules in language subjects in which the command of the language is one of the assessment criteria. In the case of any work involving a course or module taught through the medium of English the student will be required to give prior notice that he or she wishes to present work in Welsh so that the translation arrangements can be made as necessary.

Students pursuing taught programmes of study must submit a written request to the College within 4 weeks of the start of the relevant module(s) to be permitted to submit examinations, dissertations or theses in Welsh. (Where modules are of a shorter duration than 4 weeks, requests must be submitted upon enrolment on the module.) The College shall forward requests to the Academic Registrar who will then arrange for translation of questions, scripts, dissertation/thesis where required.

Students must submit their requests to their 'home' College using the following proforma:

https://intranet.swan.ac.uk/documents/welsh/Cyflwyno_gwaith_yn_Gymraeg.doc

The College shall forward copies of the forms to the Academic Registry (and where relevant, other Schools/Colleges).

Students pursuing research programmes who wish to submit their dissertation in Welsh must submit a request to the College within 3 months of initial enrolment, as part of the confirmation of candidature requirements. The College shall forward copies to the Academic Registry for translation arrangements.

PROCEDURES FOR SUBMITTING ASSESSED WORK:

You will be required to complete set coursework or other assessed work, depending on your degree programme, and you must submit this work using one or more of the following methods as advised by your Department:

- Electronically via Blackboard and Turnitin
- By use of a dropbox. Your Department will advise you of its location.
- In person to your Student Information Office.

(Please note that you will need to use the dropbox based in the department responsible for the teaching of that module, which may not necessarily be your 'home' department.)

It is extremely important that you are aware of the rules for submitting coursework and other assessed work:

- Each piece of work will have a deadline date and you will be informed by your Department what this is;

- The deadline time for submission of all assessed work is 11am. This applies to all pieces of work, irrespective of the method of submission.

Work submitted via the dropbox:

- Assessed work submitted to a dropbox is CCTV monitored for the purposes of dispute resolution;
- Students will be required to log on to the College Intranet at <http://science.swansea.ac.uk/intranet/> and print out a barcoded cover sheet. The cover sheet is unique to your piece of work and must be attached to your coursework before placing in the dropbox;
- The dropbox is emptied by the Student Information Officer at 11am promptly on the due date. All submissions are then scanned and logged as having been received by the specified deadline. An email is sent to the student confirming receipt of the coursework.

Work submitted Electronically:

- Some coursework may require electronic submission as well as or instead dropbox submission. Where this occurs, the date-stamp of the electronic submission will stand as the definitive moment of submission.

Work submitted to the Student Information Office:

- Some larger pieces of assessed work, e.g. projects, reports, posters or diaries etc, may need to be handed in personally to your Student Information Office rather than electronically or via the dropbox. In these instances you must remember to print off the front cover sheet with your work;
- You will be told by your Department which pieces of work should be submitted to your Student Information Office;
- It is important that you remember the Student Information Office has fixed opening hours, from 9am to 1pm Mon-Fri.

LATE SUBMISSION OF WORK

Penalties will apply if you do not meet the deadlines for submitting your assessed work. Please also see the information on Extenuating Circumstances, page ???.

Late Submission Penalties:

- once set, all deadlines will be regarded as absolute.
- The timing of deadlines will be consistent across the College: 11am.
- The College will impose late-submission penalties on work that is submitted after the deadline, as follows:
 - o minus 10 marks for each calendar day late;
 - o 0% will be awarded for the sooner of: either more than 7 calendar days late or when feedback on/solutions for the assessed work is made available to students.

Candidates who are prevented from meeting such deadlines due to extenuating circumstances may apply to the appropriate College Committee for consideration.

Late submission Penalty Waivers and Extenuating Circumstances:

If you submit work after the deadline and wish to apply to have the penalty waived, or if you believe that you have extenuating circumstances that contributed to the late submission, please note that:

- The bar-coded cover sheet for the submission of work via the dropbox includes a tick-box to indicate whether you intend to submit a Penalty Waiver form and/or Extenuating

Circumstances form. These forms and any supporting evidence must be submitted directly to the Teaching Administrator in your Student Information Office separately from the coursework submission;

- The Teaching Administrator in your Student Information Office will receive claims for waiving late-submission penalties and extenuating circumstances and any supporting evidence or documents for consideration;
- Claims to waive late-submission penalties will be considered and determined by the Programme Director for the Department where the module sits, i.e. this may not necessarily be the student's Home Department. The claim may be referred by the Programme Director to the College Extenuating Circumstances Committee if so deemed;
- Extenuating Circumstances claims will be considered by the College Extenuating Circumstances Committee and their recommendations will be passed to the relevant Progress and Award Boards;
- The College will deal consistently, effectively and sensitively with claims to waive late-submission penalties and for Extenuating Circumstances;
- Students have a right of appeal to the College Extenuating Circumstances Committee.

If you have any questions or are unsure of any of the regulations on assessed work please see your Teaching Administrator in your Student Information Office for guidance.

Room 206, 2nd Floor, Faraday Building

Teaching Administrator: Mrs Jill Edwards,

email: j.edwards@swansea.ac.uk

You are encouraged to refer to the regulations provided in the Academic Guide:

www.swan.ac.uk/registry/academicguide

END OF LEVEL PROGRESSION ISSUES

The full regulations relating to end of level progression issues are available in the Assessment Regulations, published in the Academic Guide:

<http://www.swansea.ac.uk/registry/academicguide/UndergraduateAwardRegulations/UndergraduateAssessmentRegulations/>.

Students are advised to examine these regulations as they change from year to year and the Department's lecturing staff are not guaranteed to be fully conversant with the current rules.

UNFAIR PRACTICE AND PLAGIARISM

Students' attention should be referred to the University's procedures for dealing with cases of Unfair Practice as included in the Academic Guide:

<http://www.swan.ac.uk/registry/academicguide/AssessmentandProgress/UnfairPracticeProcedure/>.

Each School/College will have its own Unfair Practice representative who is responsible for dealing with School/College-level cases.

Unfair practice is defined as committing any act whereby a person may obtain for him/herself or for another, an unpermitted advantage. This shall apply whether candidates act alone or in conjunction with another/others. An action or actions shall be deemed to fall within this definition whether occurring during, or in relation to, a formal examination, a piece of coursework, or any form of assessment undertaken in pursuit of an academic or professional qualification at Swansea University.

Examples of Unfair Practice include:

- the introduction into an examination room of any unauthorised form of materials such as a book, manuscript, data or loose papers, information obtained via an electronic device such as a mobile phone or any other source of unauthorised information;
- the introduction into an examination room of any authorised material which has been annotated or changed in such a way that a student is likely to have gained an unpermitted advantage;
- copying from or communicating with another person in an examination room, except as authorised by an invigilator;
- communicating electronically with any other person;
- impersonating an examination candidate or allowing oneself to be impersonated;
- presenting evidence of special circumstances to examination boards which is false or falsified or which in any way misleads or could mislead examination boards;
- presenting an examination script as your own work when the script includes material produced by unauthorised means. This includes plagiarism.

Candidates are only allowed to take permitted aids for their examination into the examination venue. These aids (e.g. pencils, biros) must be carried in a clear and transparent pencil case or "poly pocket" which will be examined on entry to the examination venue.

Examples of Unfair Practice in non-examination conditions e.g coursework, assignments, dissertations etc:

- Plagiarism (see below);
- Collusion (see below);
- Commissioning of work produced by another;
- Falsification of the results of laboratory, field-work or other forms of data collection and analysis.

Collusion can be defined as involving two or more students working together in order to gain an unfair advantage, without prior authorisation from the academic member of staff concerned (e.g. Programme Leader, Lecturer etc) to produce the same or similar piece of work and then attempting to present this entirely as their own. It is also considered Unfair Practice for a student/students to submit the work of another with or without the knowledge of the originator. An allegation of collusion may be amended during a hearing/investigation to allow a student to be exonerated of the offence whilst alleging plagiarism against another student(s); Plagiarism is the use, without acknowledgement, of another person's work and submitting it for assessment; as though it were your own work.

The risk of cheating is immense, because the penalties when you get caught could be enough to ruin your career. For instance a Law student could find that, if found guilty of engaging in Unfair Practice, he/she will not be accepted as a member of the Law Society. Likewise a student aspiring to become a teacher, doctor or nurse might find his/her chosen career path being closed prematurely if found guilty of cheating.

The penalties imposed on students vary but if you take the risk and are found guilty, the likelihood is that all your marks for the Session will be cancelled i.e. 0% for each module. You might even find that you will be required to withdraw and be disqualified from any further University examinations, particularly if it was your second attempt or if the act was of an

extremely serious nature e.g. purchasing an essay, impersonating another student or allowing oneself to be impersonated. Colleges suspecting students of engaging in Unfair Practice shall be authorised to use appropriate means to gain evidence on the allegations, such as using plagiarism detection software or examining the student orally.

Prominent Examples of Plagiarism:

- In 2000, Robbie Williams was found guilty of having plagiarised lyrics from songs by Woody Guthrie and Loudon Wainwright III in his song 'Jesus in a Camper Van'. The consequences were costly: 25% of all royalties from the song were awarded to the original copyright owners and an agreement was made to remove the song from all subsequent pressings of Williams' album *I've Been Expecting You*.
- In March 2011, German defence minister Karl Theodor zu Guttenberg was forced to resign all public office, effectively ending his career in politics, after it emerged that he had largely plagiarised his 2006 doctoral thesis.
- In 2008, celebrity psychiatrist and broadcaster Raj Persaud was found guilty of plagiarising several written articles in books and newspapers. Although the hearing ruled that his plagiarism was not financially motivated and did not result in patient harm, Dr. Persaud was barred from practice for three months and, more pertinently, suffered permanent damage to his reputation and withdrew from the public eye.

The University is committed to providing support and guidance to prevent cases of accidental plagiarism and you are advised to seek further advice from your College if you are in any doubt as to what is required of you. Ignorance is not accepted as an appropriate defence.

The University may use electronic detection software to identify cases of unfair practice. However, knowledge of resources, changes in writing styles and google searches may prove just as effective. You may be required to submit work to Turnitin. If this is the case, you must do this by the specified deadline.

It is also possible to conduct an oral examination on any piece of work, if a College suspects a candidate of unfair practice. Again, Colleges may require students to provide copies of notes/earlier drafts of assessments, as proof that the final submission was their own.

ILLNESS DURING THE EXAMINATIONS PERIOD

Students who fall ill during the examination period, or who are adversely affected or unable to attend examinations for other reasons must inform a member of the lecturing staff **as soon as possible**.

Students whose work is interrupted by illness for less than seven days are required to inform the School by submitting a completed student self-certification form (Appendix 1) which should be accompanied by a medical certificate.

A student who has been absent from an examination must send without delay to the College a written explanation. In the case of illness, he/she must also submit a medical certificate.

You are advised always to ask your doctor for a medical certificate for illnesses which affect your studies, and present these to the College as soon as possible. It is usually too late to tell your tutor after an Examination Board has met and retrospective medical circumstances will not normally be accepted.

STUDENTS WITH EXTENUATING CIRCUMSTANCES AND/OR SPECIAL REQUIREMENTS

The University is aware that various factors during a student's career might affect academic performance. Illness is the most common factor responsible for student under-performance but other extenuating circumstances are as valid e.g. bereavement involving a close relative.

Examiners can take such factors into account before marks are finalized and before decisions relating to progression or the conferment of an award are taken. Therefore students must draw any circumstances, which might have affected their performance, to the attention of the Examiners/Tutors before the meeting of the College Examination Board and not after the results have been released. Evidence must also be provided.

A School/College might even consider awarding a compensatory pass for students who fail a module but have presented evidence to account for the failure.

Any student who, for whatever reason, requests special allowance/consideration with regard to assessment, must make his/her needs known and be prepared to produce appropriate supporting evidence. The responsibility to inform the School, College or the Disability Office rests with the student. You must tell Colleges of extenuating circumstances as soon as possible since retrospective applications will not normally be considered. You may present evidence (e.g. medical certificates) in a sealed envelope if you wish, which will only be opened for consideration by the Special Circumstances Committee.

For further information, please refer to the Academic Guide:

<http://www.swansea.ac.uk/registry/academicguide/AssessmentIssues/ExtenuatingCircumstances/>.

APPEALS PROCEDURE

An academic appeal is defined as a request for a review of a decision of an academic body charged with decisions on student progression, assessment and awards. Students/graduates may submit an academic appeal against the decision of an Examining Board, the University or one of its sub-committees if one or more of the following situations are relevant:

- Candidates who are prevented from continuing with their studies part-way through a level of study or part of a programme.
- Candidates who fail to qualify to proceed to the next stage of their programme at the end of a level, end of a part or end of a year.
- Candidates who have completed their programme but who wish to appeal against the result or candidates who are dissatisfied with the award of an exit qualification of the University.
- Where the implications of the progression decision taken by the Examining Board, the University or one of its Sub-Committees may have a significant impact on the student's overall result (e.g. capping of marks).

All appeals are subject to an initial filtering process. Any appeals based on the grounds of extenuating circumstances not previously disclosed to an Examination Board/College will be dismissed at the filtering stage of the appeals process unless compelling reasons are shown as to why these could not have been previously disclosed. Students who have completed the level/stage of study, or have been granted supplementary examinations or other form of assessment of a module at the next available opportunity, shall not normally be permitted to submit an appeal. Such students may, however, submit a request for the verification of their results (see 'Verification Procedures' below).

All requests for an appeal against a University decision must be directed to the Assistant Registrar (Appeals), Academic Registry, Swansea University, Singleton Park, Swansea, SA2 8PP. Further information about, or clarification of, these procedures is available in the Academic Guide,

<http://www.swansea.ac.uk/registry/academicguide/AssessmentandProgress/AcademicAppeals/>, and from the Academic Registry.

VERIFICATION PROCEDURES

Students have the opportunity to request a verification of their results if they wish to query the accuracy of the marks recorded for the session. A fee is payable to cover the costs of administration, though this fee will be reimbursed if the verification request is successful.

Further details are available in the Academic Guide:

<http://www.swan.ac.uk/registry/academicguide/AssessmentandProgress/VerificationRegulationsandProcedures/>.

ANONYMITY/DOUBLE MARKING

You can expect all of your formal written examinations to be marked in their anonymous state. Only the student number will be used to identify these scripts until such time as both first and double marking has been completed. Whilst anonymity is protected for as long as possible there are instances where identity will be disclosed:

- If you submit an application for extenuating circumstances the College will identify your script to ensure these circumstances are properly taken into account
- Anonymity is not maintained for final degree classifications

Schools/Colleges may conduct examination boards without the disclosure of your name however Progression and Award Boards will be conducted by name.

The University-wide policy on double marking is published in the Academic Guide: <http://www.swansea.ac.uk/registry/academicguide/AssessmentandProgress/DoubleMarkingPolicy/>. Schools/Colleges are expected to employ one of the following forms of double marking: universal double blind marking; universal non-blind double marking; universalsecond marking as check or audit; second marking as sampling or moderation; and partial second marking.

POLICY ON PUBLISHING OF STUDENT MARKS

Progression and final award decisions shall not be published by Schools/Colleges but will be communicated to students by the Academic Registry. Subsequent to this communication, for purposes of the degree ceremonies, Schools/Colleges shall be permitted to publish the award decision by student name. In this case, students should be given the option of requesting that their name and result be omitted from any such publication.

EMPLOYABILITY

This section of the handbook provides you with an overall view of the opportunities there are to develop and enhance their employability through their degree programme. It may be possible for you to pursue some modules through the medium of Welsh and your department will advise you on this

You may find Subject Benchmark statements a useful guide for this section, available from the Quality Office, Academic Registry or from the QAA website:
qaa.ac.uk/academicinfrastructure/benchmark/

LEARNING AND PROFESSIONAL DEVELOPMENT - LEAP

During your time at university one of the most important skills you will develop is being able to take responsibility for, and manage, your own learning and development. Since the academic year 2005/06 all universities have been required to provide opportunities for their students to undertake this type of personal development activity throughout their degree programme. At Swansea we call this activity Learning and Professional Development. By taking advantage of the opportunities offered for LEAP you can make the most of your time at university in both your academic work and other activities.

In practical terms LEAP for students means being able to review how and what you are learning and then plan how you are going to use this knowledge to consolidate and improve your performance.

Resources to help you with Learning and Professional Development are provided through the Pebble Pad E-portfolio platform at
pebblepad.swansea.ac.uk

There are also online courses to support your LEAP and Employability available in the Virtual Learning Environment Blackboard. You will find these in the 'My Courses' section once you have logged into Blackboard under the course 'LEAP'. These courses link closely to PebblePad.

In the 'Interactive Career Planning' course in Blackboard is a career learning tool called Destinations that contains a number of topics each exploring aspects of a popular career learning theme such as 'Interviews' or 'Personality'. Destinations is supported by up-to-date multimedia resources, short videos to support students with their career planning and job search.

Extra-curricular activities, student jobs, and placements will also provide opportunities to use and further develop skills of interest to employers and you should use the LEAP process to review them regularly.

Through undertaking the LEAP process you will build up knowledge about yourself which as well as being useful in helping you to take responsibility for your own learning and make progress academically, will also help you to make decisions about your future after university and to be able to build an effective and informative CV.

Learning and Professional Development – Specific Departmental Resources and practice

You might wish to undertake an industrial placement or to study abroad during your time as a student at Swansea. There are also opportunities to continue to learn other languages. You are advised to speak to your Personal Tutor about these options.

Employability skills information resources can be found on the Careers and Employability Service's website at <http://www.swansea.ac.uk/careers>.

The University's Careers Education, Information and Guidance (CEIG) policy has been revised and updated and aims to provide students with the knowledge, skills and qualities to:

- secure their first job after graduation;
- use these attributes subsequently in the effective management and development of their career.

Information for students about the CEIG policy can be found in the Teaching Quality Manual, available online at

www.swansea.ac.uk/registry/a-zguide/t/teachingqualitymanual

CAREER PROSPECTS/EMPLOYABILITY

One of the key aims of the University is to prepare and promote the employability of our students. Naturally, the Careers Centre plays an important role in fulfilling this aim by working with academic staff in the delivery of Careers Education & Guidance to enable our students to develop the necessary knowledge and skills for the world of work. This fits well with the implementation of Learning & Professional Development / Personal Development Planning in which you will be taking part during your time at Swansea.

Jean Brokenshire is your careers adviser and her role is to give guidance, advice and information to students and graduates about your education, training and employment, and help you make informed decisions. The following Careers programmes are designed to help you develop ideas and strategies about career planning and manage your learning and professional development, and ultimately secure a graduate job at the end of the degree course:

For level 1 students, apart from attending the introductory talk, we strongly recommend you to take part in a self-assessment called Profiling for Success and carry out a skills audit so that you know where your skills gaps are and what you plan to do about it. You can access the programme through Blackboard which you will be introduced to when you join the library.

What is Profiling for Success? It is an online psychometric assessment programme which offers 3 aptitude tests and 2 personal development planning assessments. For level 1 students, we recommend you to make use of the personal development programmes. Both the Learning Styles and Personality Assessments are not timed and there are no right or wrong answers. The test results are sent back to you by email in the form of reports which detail how you can make the most of learning styles, your strengths and weaknesses, as well as some careers ideas for you to explore. Most importantly, the whole process can be confidential if you so choose and none of these form any part of your degree assessment.

When you reach level 2, you should have some broad direction in terms of careers. We recommend Adult Directions and Prospects Planner to help you clarify your preferences. Adult Directions is an interactive careers software package. It is designed as an interest guide exercise which takes student's likes and dislikes into consideration before generating a list of matching job suggestions. This is a licensed software; the programme is available through the

Blackboard (Interactive Career Planning). Prospects Planner does similar things to Adult Directions but the programme is part of the Prospects website (www.prospects.ac.uk) and you can also access it through Interactive Career Planning. It helps users identify what they want out of a job and what they have to offer to the job / employer. Using both can certainly help you clarify your direction.

Also during your level 2, you are strongly advised to gain some work experience in the areas that interest you because it would be difficult to fit in work placements in your final year due to study and exams. For some companies, you can pick up the phone and ask but for some, you will have to apply by submitting a CV or application. So, it will be a good idea to have a CV before you approach the companies. If you have not done a CV before and would like some information on how to write a CV, you can download the CV talk or have a look at the PowerPoint presentation on Blackboard.

When you reach level 3 or your final year, you may be able to have a 1-to-1 guidance interview to review how you got on and what further help or support you might need from us. It's worth remembering some employers impose their deadlines for the graduate trainee schemes in the autumn term, and over one-third of the large employers use psychometric tests as part of their selection; this is why we recommend our students to get an idea about their verbal / numerical reasoning to improve your performance. As mentioned earlier, Profiling for Success offers 3 aptitude tests and you can practice as many times as you need.

In addition to these programmes, the Careers Centre organises a range of employer talks and Careers fairs, provides E-guidance, careers information, reference books, videos and DVDs and of course vacancies information, job shop and placement services. You can find out more about these services from the Careers website (www.swansea.ac.uk/careers) or call in at the Careers office.

The main messages are:

Your time here goes very quickly and we would like to encourage you to think about where you would like to be when you graduate. Secondly, many people have degrees these days, a degree alone is often not sufficient to secure a graduate job. Employers like to see a core set of skills, regardless of your degree subject. So, it's never too early to start thinking about your options.

Our closing message is: preparation and participation are the keys to success. So, work hard, get a good degree, take part in work experience and actively seek learning opportunities to develop the skills that employers look for. Those who are proactive are more likely to end up where they want to be.

PERSONAL QUALITIES

It cannot be emphasised enough that your studies and future potential can be enhanced by developing your personal qualities, obtaining temporary and vacation employment and to become involved in sporting and other leisure activities. Students are encouraged to participate in any personal development that may enhance employment opportunities

STUDENT CASUAL JOBS, WORK PLACEMENTS AND WORK TASTERS

Work experience can be particularly valuable and employers expect this experience because it develops a range of skills and qualities which are not easy to acquire through study or in their leisure time. Support is available to those seeking such employment from the Jobs section of

Careers and Employability Service's website at www.swan.ac.uk/careers/jobs .

Careers and Employability also helps place students with local small and medium sized employers, through the GO Wales programme www.gowales.co.uk. This service provides a valuable opportunity for students to gain practical work based skills at undergraduate and postgraduate level.

BRITISH COMPUTER SOCIETY ACCREDITATION

The British Computer Society (BCS)

The British Computer Society (BCS) is the leading professional and learned Society in the field of computers, software and information systems. Formed in 1957, it has over 34000 members in the UK and overseas. The Society is concerned with the development of computing and its effective application. Under its Royal Charter, granted in 1984, it also has responsibilities for education and training, for public awareness, and above all for standards, quality and professionalism. The BCS is also an Engineering Institution, and it is fully licensed by the Engineering Council to nominate Chartered and Incorporated Engineers and to accredit university courses and training schemes.

Only computer scientists who are well educated and experienced are eligible to become members of the BCS. The BCS sets high standards for membership including examinations. If you are interested in BCS accreditation you should contact the BCS liaison officer in the Department for further details. Students following other schemes including Computer Science may apply for exemption from the membership examination on an individual basis. The BCS welcomes applications for student membership from students studying in the Department. For details, please contact:

The British Computer Society,
1 Sanford Street,
Swindon,
Wiltshire,
SN1 1HJ

URL: <http://www.bcs.org.uk/>, Email: bcshq@hq.bcs.org.uk

Telephone: (+44) 01793 417417,

Fax: (+44) 01793 480270

BCS has an active South Wales branch and a regular programme of lectures and site visits.

BCS Accreditation

Following the most recent visit by the British Computer Society in 2009/2009 the accreditation for the various courses offered by the Department will have changed. For full details of accreditation and exemption for any specific course students are advised to consult the course coordinator. For all courses students will need to have at least completed, at the first attempt, a practical and problem solving project.

Project Requirements

The following requirements for projects to be acceptable for BCS Accreditation is extracted from information supplied by the BCS. If you have any doubts as to whether your project meets the requirements please discuss it with your supervisor. For projects to meet the requirements for exemption, they must be passed at the first attempt to gain the award, with no condonement and no referral. All projects must involve a professional approach. The project should be “real” in the sense that the product is for users other than the author. The report should demonstrate an appropriate level of professional competence in the practical development of a suitable application, tool or similar product. The report on the project should include:

- elucidation of the problem and the objectives of the project;
- an in-depth investigation of the context/literature/other similar products;
- a clear description of the stages of the life cycle undertaken;
- a description of the use of appropriate tools to support the development process;
- a description of how verification and validation were applied at all stages;

- a critical appraisal of the project, indicating the rationale for design/implementation decisions, lessons learnt during the course of the project, and evaluation (with hindsight) of the product and the process of its production (including a review of the plan and
- in the case of group projects, a clear indication of the part played by the author in achieving the goals of the project;
- references;
- appendices-technical documentation.

The specific criteria applicable to projects acceptable for full exemption from the BCS Professional Examinations is that:

- it involves at least 150 hours of individual student effort;
- the task should be to develop an IT solution to a practical problem, which would include the production of a piece of software (which may be interpreted as some or all of a specification, design or implementation of software);
- it exhibits a structured approach to information systems practice, involving a number of stages in the life cycle;
- the product exhibits the attributes of quality, reliability, timeliness and maintainability;
- it must involve the production of a professional report as described above;
- it must lead to a description of the process and of the product;
- it must contribute significantly to the overall award classification.

GENERAL INFORMATION

Health and Safety

Owing to Fire and Health & Safety Regulations (and from the general considerations of security), the hours of entry to certain buildings on the campus are restricted. Outside these hours, entry can only be gained on the written permission of the Head of College or other authority concerned.

Emergency Procedures

In the event of a medical (or other) emergency call 333 immediately on a University landline or ask a member of staff to do so. If you are unsure if 333 has been called, call 333 again. For all emergency calls on University property dial 333 on a University 'phone. This helps the University to speed the arrival of the Emergency Services.

IN THE EVENT OF A FIRE

1. Raise the alarm at once by breaking the glass of the nearest fire alarm call point.
2. Send the first available person to telephone 333 and give the location of the fire.
3. If appropriate, call for assistance and attack the fire with the correct extinguisher.
4. If the fire should get out of control, or your escape is threatened, leave the building at once, closing doors and windows as you go.

IF YOU HEAR THE FIRE ALARM

1. Leave the building immediately, closing all doors behind you.
2. When clear of the building proceed at once to the assembly area for that building (as indicated on the blue Fire Action signs around the building).

USE NEAREST AVAILABLE EXIT.

DO NOT STOP TO COLLECT PERSONAL BELONGINGS.

DO NOT USE LIFTS.

DO NOT RE-ENTER THE BUILDING.

24 Hour Emergency Services-Ambulance / Fire / Police

On campus, Hendrefoelan & Beck Hall Single Rooms

Tel: 333

Data Protection

The University's procedures comply with the principles of the Data Protection Act 1998. The responsibilities of students in relation to the provision of personal data can be found under the Publications tab on the Academic Registry Website. Students as data subjects have a right to request from the University a copy of their own personal data. A standard form must be completed and a fee of £10 is charged for each request. Forms and further details can be found at: <http://www.swan.ac.uk/university/Administration/RecordsManagement/>

The University's registration number with the Information Commissioner is Z6102454.

Transcripts and Diploma Supplements

You can expect to receive an academic transcript at the end of your studies at Swansea which details the modules you pursued and the marks obtained. In addition to the transcript, the University produces a diploma supplement for all final year students. The diploma supplement is a document, developed by groups within the European Commission that aims to facilitate academic and professional recognition of qualifications across Europe. It provides a description of the nature, level, context, content and status of the studies that were pursued and successfully completed by the individual named. It is expected that employers and higher education institutions will increasingly request this document from you when you begin

applying for posts or further study.

Transfer of Programmes and Modules:

You are allowed to change your modules during a limited period at the start of each module by completing and returning a Module Selection Authorisation Slip i.e. before the end of the second week of a module finishing in December/January and before the end of the fourth week for modules finishing later in the Academic year, normally in May/June.

If you drop a module after six weeks then you should return a Late Transfer of Modules slip. You will normally be expected to pay a fee for the additional module. You can change your programme if the Academic department(s) involved agree but you **MUST** return a completed Programme Transfer Form before you begin the second year of your current programme. Also, before proceeding always contact Student Records/Tuition Fee Officer and your LEA for advice about financial support.

Suspending Studies:

Candidates who request to suspend studies shall be required to complete the appropriate proforma (available from the Academic Registry web pages or College). Such applications shall include the grounds for the request and evidence to support the case. Normally, students should attend for interview with a member of the College and contact a member of the Academic Registry for advice. Such applications shall be forwarded to the Dean of the relevant Faculty for approval.

Candidates requesting a suspension of studies shall be required to submit the proforma to the Academic Registry during the first two terms and **no later than the first day of the summer term**, unless it is for health or other compelling reasons.

STUDENT OBLIGATIONS – UNIVERSITY

Enrolment:

You must enrol with the University on an annual basis in accordance with the relevant regulations under the Publications tab on the Academic Registry Website: <http://www.swan.ac.uk/registry/Publications/>. Students are requested to enrol online but may enrol in person. To enrol on a programme of study:

- Students check personal and programme details are correct and agree to a declaration confirming you agree to abide by the Rules and Regulations of the University.
- Students must finalise arrangements for paying tuition fees. For example, many UK undergraduates opt for a 100% fee loan from the SLC to pay their course fee. Many postgraduates are externally sponsored (e.g. embassy, trust) in which case students must hand in a confirmation of sponsorship form or a letter from your sponsor with the equivalent details.
- A student paying part or all fees directly, must either return a direct debit mandate for a UK bank account or make the first tuition fee payment.
- Students in receipt of a University bursary contribution towards their tuition fee can confirm details are correct during enrolment. If details are not correct the student should contact the member of staff who originated the bursary in order to progress authorisation.
- International students must present a current passport with a visa (and if you have one a UK identity card) as evidence of entitlement to study at Swansea University.

After presenting valid documents, forms and finalising tuition fee arrangements the university record will be amended to show a student is enrolled on a programme of study. Debtors from a previous session will not be allowed to enrol unless the outstanding amount is paid in full.

Failure to enrol within a prescribed enrolment period will result in candidature lapsing and withdrawal of the candidate from the university. Candidates shall be given the opportunity to appeal within 7 days against the decision. For further information browse the University's web page:

http://www.swan.ac.uk/Student_Records/Enrolment/

Module Selection:

Each student on a modular degree must select modules, notifying the College in accordance with the procedures currently being applied. (See the regulations governing module selection on the Academic Registry Website). The main rules to be noted are:

- Full-time students should select modules which have a total credit weighting of 120 credit points;
- Part-time students should select modules which normally should have a total credit weighting of 60 credit points;
- If a student is repeating failed modules, only the modules being repeated will be recorded on the academic record for the current session.

For further information contact your College or the Student Records Office:

Timetable Clashes:

It is the student's responsibility to identify clashes in their timetable as early as possible. Once identified these clashes should be raised with the student's personal tutor who will discuss the options for resolving these clashes and take appropriate action. For Joint Honours students the same process should be followed, clashes should be identified and reported to the personal tutor as a matter of urgency, the personal tutor will in this case discuss the options for resolving

these clashes and coordinate with other contacts responsible for the other parts of the degree being taught outside the subject area/school, on the student's behalf.

Payment of Fees:

Regulations relating to payment of tuition fees, accommodation fees and other fees are outlined on the Academic Registry Website. Payment of Tuition Fees: The Income Section of the Finance Department, Finance Building;

Telephone: 01792 295436;

E-mail: *income@swansea.ac.uk*

Payment of Accommodation Fees:

The Accommodation Office,

Penmaen Residence:

Telephone: 01792 295101

E-mail: *Rebecca.Evans@swansea.ac.uk* or

C.R.Harper@swansea.ac.uk

General Conduct and Behaviour:

You will conduct yourself in an orderly manner.

- If you wilfully damage University property you must pay for its repair and may be subject to disciplinary action.
- If you attempt to obstruct teaching, study, research or the administration of the University you will be liable to disciplinary action.
- Unauthorised absence from the University without proper cause will be liable to disciplinary action.
- You are under an obligation to inform your College of any criminal conviction prior to and during your period as a student.

General Regulations/Academic Regulations:

You should be made aware that you must abide by the University's Academic Regulations and General Regulations, which are outlined on the Academic Registry Website.

Complaints:

If you wish to make a complaint, please first read the *complaints* section under on the Academic Registry Website, outlining the University's Complaints procedure

ATTENDANCE POLICY

The University will monitor students' attendance and will take relevant action involving those students whose attendance is deemed to be unsatisfactory. The University expects you to attend all the teaching sessions associated with each module which you have elected to pursue. You must also adhere to the College's requirements identified in the College Attendance Policy, relating to compulsory teaching sessions. The need to monitor your attendance is primarily in order to enable Personal Tutors to undertake a pastoral role, to ensure your well-being. Attendance is also monitored as a requirement of External Authorities e.g. LEAs, Sponsors, UK Border Agency (UKBA). Your College will identify at least one interaction per month during term time for student pursuing Taught programmes, with 3 instances of interaction per semester. An interaction shall be defined as a contact between a student and an

academic or administrative staff member for instance a tutorial, seminar, lecture, practical session, field trip, a meeting with a Personal Tutor, an examination sitting, completing an attendance register etc.

If you are absent for any period of time you must provide your College with an explanation for your absence. Wherever possible you should seek permission in advance for any absences, this includes any time spent away from Swansea for research or personal reasons. A form is available from your College.

If you are an overseas student a central university monitoring system shall also be applied. You will be required to sign a monthly register to confirm your attendance and continued engagement with the programme on a monthly basis. This is in addition to complying with their College's attendance policy. At least one central interaction shall be identified each month (during term-time for students on taught programmes) with 3 points of interaction per semester. You will be notified of the dates on which you will be required to register by the Academic Registry.

If your attendance is unsatisfactory over two consecutive months you will be reported to the appropriate University Board. Continued unsatisfactory attendance may result in you being required to withdraw from the University. The University shall notify relevant authorities (Sponsors, Local Education Authorities, UKBA etc) of any student who is withdrawn from the University as soon as practically possible following the decision, taking into account the students' right of appeal. Such a confirmation of withdrawal may be delayed for up to a period of one month following the original decision. Colleges shall establish appropriate procedures to monitor students' general attendance which, as a minimum requirement, shall be sufficiently robust to identify student absence on a fortnightly basis. It is for this reason that you must try to meet the deadlines set for submitting appeals. The University does not accept responsibility for the consequences of informing External Agencies of a student's withdrawal if these deadlines have not been met and if the University's attention has not been drawn to the relevant information.

The University's Attendance Policy can be found in the Academic Guide:

www.swansea.ac.uk/registry/academicguide/assessmentandprogress/policyonattendance

STUDENT OBLIGATIONS – DEPARTMENT

Regulations for Using Computing Resources:

In general, the regulations for the use of computing facilities in the University (see Swansea University Handbook) are applicable to the Department of Computer Science. In addition to the University regulations, the following rules should be observed. In the following text the term “computing resources” includes all computing hardware, software, user and system manuals, communications equipment and services, and external facilities made available by the Department.

- No person shall use any computing resources of the Department without due authorisation given by the Department.
- Every allocation of computing resources shall be made on the understanding that it is to be used only for the purpose for which it was requested and only by the person by whom or on whose behalf the request was made. Use shall not be made of computing resources allocated to another person unless such has been specifically authorised by the Department.
- No person shall by any will or deliberate act jeopardise or corrupt or attempt to jeopardise or corrupt the integrity of the computing equipment, its system programs, or any other stored information.
- No person may act in any way which leads to or could be expected to lead to the disruption of the approved work of other authorised users.
- Every person authorised to use computing resources shall be expected to treat as privileged any information not provided or generated by himself which may become available to him through his use of computing resources; shall not copy, modify, disseminate, or use any part of it without permission of the appropriate person or body.
- In the case of any information which is designated in a notice issued by the Department as proprietary or otherwise confidential, every person using the computing resources of the Department shall be required:
 - (a) to observe the instructions that may be issued by the Department specifying ways in which the information may be used.
 - (b) not to copy, modify, disseminate, or make use of it in any way not specified in these instructions, without first obtaining written permission from the Director of Support Services of the Department.
- If any person has been allocated computing resources for purposes other than Department teaching and research, he or she shall be personally responsible for reimbursing the Department for the cost of resources used.
- Users of external facilities may use them only if they agree to follow any rules laid down from time to time by the institution providing these facilities.
- The Director of Support Services of the Department may suspend any person who is believed to be in breach of these rules from use of all or specified Departmental computing resources. He may also make subsequent use subject to such conditions as he thinks fit. The Director shall report the matter to the Head of Department who may recommend further action, where necessary.
- The Director of Support Services of the Department may make such general conditions on the use of Departmental computing resources as he thinks appropriate from time to time. An Appeal against the actions of the Director of Support Services of the Department under rules 9 and 10 shall be made to the Head of Department.

Laboratory Code of Conduct:

- The Laboratories are open between 8.00am and 6.00pm each week day. The porters and night security staff are fully authorised by the University to demand that students or others leave the rooms, the laboratories or the building at any time.
- Laboratory passes are to be carried at all times, and when requested, shown to members of University staff.
- Food and drink must not be consumed in any of the laboratories.
- No smoking in any University buildings, especially not in computing laboratories.
- No equipment is to be disconnected. User manuals and system documents must not be removed from the laboratories.
- Users' own computer equipment must not be taken into the laboratories, or connected to University networks, without due authorisation by a member of support staff.
- The laboratory facilities must be kept in a neat and tidy condition at all times. Please put any waste paper in the bins or boxes provided.

The Robert Recorde Room:

The Robert Recorde Room is the Seminar Room of the Department of Computer Science, University of Wales Swansea. It was created by the Department in 2000 and named after Robert Recorde (c1510-1558) who was a distinguished writer of books on arithmetic, practical calculation and geometry.

Robert Recorde was born c1510 in Tenby, Pembrokeshire. He was educated at Oxford (BA and Fellowship of All Souls in 1531) and Cambridge (MD in 1545). He practiced medicine, and was a public servant (Controller of the Bristol Mint, Surveyor of the Mines and Monies in Ireland). He died in London in 1558, imprisoned for £1000 debt following a libel case brought by Herbert, Earl of Pembroke.

Recorde was an able teacher, an author of important books, and an outstanding scholar of the 16th Century. Among his books on computation that are of special interest are:

The Ground of Artes, (1543). This is an influential book on arithmetic in the vernacular that explains both the “new” decimal arithmetic and the “ancient” abacus arithmetic.

The Pathway to Knowledge, (1551). A version of the first four books of Euclid with an emphasis on algorithmic constructions.

The Whetstone of Witte, (1557). This work is famous for the explicit invention of the equality sign = thus making an account of algebra that was completely symbolic.

“... yet muste you and all men take heed, that ... in al mennes workes, you be not abused by their autoritye, but evermore attend to their reasons, and examine them well, ever regarding more what is saide, and how it is proved, then who saieth it: for autoritie often times deceaveth many menne.”

Robert Recorde
The Castle of Knowledge 1556

2011/2012 SESSION CALENDAR

SEPTEMBER							OCTOBER							NOVEMBER						
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
			1	2	3	4						1	2		1	2	3	4	5	6
5	6	7	8	9	10	11	3	4	5	6	7	8	9	7	8	9	10	11	12	13
12	13	14	15	16	17	18	10	11	12	13	14	15	16	14	15	16	17	18	19	20
19	20	21	22	23	24	25	17	18	19	20	21	22	23	21	22	23	24	25	26	27
26	27	28	29	30			24	25	26	27	28	29	30	28	29	30				
							31													

DECEMBER							JANUARY							FEBRUARY						
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
			1	2	3	4							1			1	2	3	4	5
5	6	7	8	9	10	11	2	3	4	5	6	7	8	6	7	8	9	10	11	12
12	13	14	15	16	17	18	9	10	11	12	13	14	15	13	14	15	16	17	18	19
19	20	21	22	23	24	25	16	17	18	19	20	21	22	20	21	22	23	24	25	26
26	27	28	29	30	31		23	24	25	26	27	28	29	27	28	29				
							30	31												

MARCH							APRIL							MAY						
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
			1	2	3	4							1		1	2	3	4	5	6
5	6	7	8	9	10	11	2	3	4	5	6	7	8	7	8	9	10	11	12	13
12	13	14	15	16	17	18	9	10	11	12	13	14	15	14	15	16	17	18	19	20
19	20	21	22	23	24	25	16	17	18	19	20	21	22	21	22	23	24	25	26	27
26	27	28	29	30	31		23	24	25	26	27	28	29	28	29	30	31			
							30													

JUNE							JULY							AUGUST						
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
				1	2	3							1			1	2	3	4	5
4	5	6	7	8	9	10	2	3	4	5	6	7	8	6	7	8	9	10	11	12
11	12	13	14	15	16	17	9	10	11	12	13	14	15	13	14	15	16	17	18	19
18	19	20	21	22	23	24	16	17	18	19	20	21	22	20	21	22	23	24	25	26
25	26	27	28	29	30		23	24	25	26	27	28	29	27	28	29	30	31		
							30	31												

Important Academic Dates

Michaelmas Term:	26th September 2011 - 16th December 2011
Lent Term:	9th January 2012 - 30th March 2012
Summer Term:	23rd April 2012 - 29th June 2012
Gregynog Colloquium:	21st November 2011 - 23rd November 2011
Project Demonstration Fair:	
Level One Awayday:	
Level Two Awayday:	
Induction/Enrolment:	26th September 2011 - 30th September 2011
Semester One:	3rd October 2011 - 27th January 2012
Mid-Sessional Exams:	9th January 2012 - 20th January 2012
Feedback and Employability (1)	23rd January 2012 - 27th January 2012
Semester Two:	30th January 2012 - 29th June 2012
Sessional Revision:	30th April 2012 - 4th May 2012
Sessional Exams:	8th May 2012 - 8th June 2012
Supplementary Exams:	

All dates are subject to confirmation.;