Computer Science
Course Handbook

2010/2011

Edited by C. J. Whyley
August 2010

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.cs.swansea.ac.uk
difficilia quae pulchra

things that are excellent are difficult
CONTENTS

Introduction to the School of Physical Sciences ............................................ 1
Introduction to the Department of Computer Science .................................... 2
Computer Science Course Structure ............................................................. 4
Staff of the Department ............................................................................... 5
Swansea University History of Computing Collection ..................................... 10
Contact Points: Department ......................................................................... 11
Contact Points: University ............................................................................. 15
University Support for Students ................................................................... 17
Departmental Committees ............................................................................. 20
Staff Directory ............................................................................................... 23
General Information ....................................................................................... 24

Degree Programmes ...................................................................................... 27
  Introduction ................................................................................................. 29
  Computer Science Modules and Educational Aims ....................................... 32
  Computer Science Modules for programmes ............................................... 36
  BSc Computer Science ............................................................................... 38
  MEng Computing ....................................................................................... 39
  BSc Computing and Communications ......................................................... 40
  BSc Mathematics for Computer Science .................................................... 41
  BSc Computing with Finance ..................................................................... 42
  BSc Computer Science with a Modern Language ....................................... 43
  BSc Computer Science and Pure Mathematics .......................................... 44
  BSc Computer Science and Geoinformatics ................................................. 45
  BSc Computer Science and Psychology ..................................................... 46
  BSc Computer Science and Physics ........................................................... 47
  BA Modern Language(s) with Computer Studies ........................................ 48
  Level Three Optional Modules ................................................................... 49
  MSc Computer Science ............................................................................... 50
  MSc Computing and Software Technology ............................................... 51
  MSc Computing and Future Interaction Technologies ................................... 52
  MRes Computing and Future Interaction Technologies ............................. 52
  MRes Logic and Computation ................................................................... 53
  MRes Visual Computing .............................................................................. 53
  UCAS and UWS Route Codes .................................................................... 54

Level Zero Modules ...................................................................................... 55
  CS-061 Introduction to Computing I ........................................................... 57
  CS-071 Introduction to Computing II ........................................................... 58
Level One Modules .................................................. 59
CS-106 Discrete Mathematics for Computer Science I .................. 61
CS-108 Computers and Computing .................................... 62
CS-113 From Languages to Hardware .................................. 63
CS-116 Modelling Computer Systems .................................. 64
CS-124 Computers and Society ........................................ 65
CS-126 Discrete Mathematics for Computer Science II ................ 66
CS-141 Programming: Principles and Practice ....................... 67
CS-142 Programming: Performance and Efficiency .................... 68
CS-144 Software Development: Tools and Techniques ............... 69
CS-161 Introduction to Computing I ................................... 70
CS-171 Introduction to Computing II ................................... 71
CS-191 Functional Programming I .................................... 72
CS-199 Computers Unplugged ......................................... 73
Level One Awayday ................................................... 74

Level Two Modules ................................................... 75
CS-213 System Specification ........................................... 77
CS-215 Logic Programming and Artificial Intelligence ............... 78
CS-217 Computer Graphics I: Image Processing and Synthesis .... 79
CS-219 Database Systems ............................................. 80
CS-236 Logic and Computation ....................................... 81
CS-242 Algorithms .................................................... 82
CS-248 Concurrent Systems ........................................... 83
CS-254 Software Engineering ......................................... 84

Level Three Modules .................................................. 87
CS-307 Computer Graphics II: Modelling and Rendering .......... 89
CS-311 Concepts of Programming Languages ....................... 90
CS-313 High Integrity Systems ...................................... 91
CS-317 Computer Graphics Laboratory ................................ 92
CS-318 Cryptography and IT Security ................................ 93
CS-337 Data Visualisation ........................................... 94
CS-338 Internet Computing .......................................... 95
CS-339 Advanced Topics in Computer Science ...................... 96
CS-344 Project Implementation and Dissertation ................... 97
CS-345 Artificial Intelligence Applications .......................... 99
CS-348 Building Reliable Web Applications ......................... 100
CS-349 Mobile Interaction Design ................................... 101
CS-354 Project Specification and Development ....................... 102
CS-358 High-Performance Computing in C/C++ ..................... 104
CS-364 Software Testing ............................................. 105
CS-371 Design Patterns and Generic Programming ................ 106
CS-375 Logic for Computer Science .................................. 107
CS-377 Fundamentals of Computer Vision ........................... 108
MS-306 New Screen Technologies .................................. 109
MS-353 Digital Philosophy: The Roots of the Virtual ............. 110
HIP300 The Classical Tradition in the Sciences ................... 112
HIP301 From Natural Philosophy to Science ......................... 113
Undergraduate Computer Science Colloquium ....................... 114
Project Demonstration Fair .......................................... 115
Level M Modules

- 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-M13 Critical Systems
- CS-M14 Industrial Project
- CS-M15 Directed Studies in Logic and Computation
- CS-M17 Volume Graphics
- CS-M18 IT Security: Theory and Practice
- CS-M19 Interactive System Design
- CS-M25 Research Methodology and Project Specification
- CS-M29 Mobile Interaction Design
- CS-M32 Algorithm Design and Analysis
- CS-M34 Software Project
- CS-M35 Logic and Computation Project
- CS-M37 Graphics Surveys and Research Methodology
- CS-M39 Interaction Technologies: Seminars and Readings
- CS-M41 Programming in Java
- CS-M49 Interaction Technologies: Lab and Field Work
- CS-M51 Data Storage and Manipulation
- CS-M58 Distributed O-O Programming
- CS-M59 Relational and Object Oriented Database Systems
- CS-M61 Concepts of Programming Languages
- CS-M64 Computer Software Systems
- 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-M71 Design Patterns and Generic Programming
- CS-M74 Software Product Development
- CS-M76 Abstract Data Types and Program Synthesis
- CS-M77 Fundamentals of Computer Vision
- CS-M78 High Performance Programming in C/C++
- CS-M79 Interaction Technologies: Hardware and Devices
- CS-M81 Distributed Programming in Java
- CS-M97 State of the Art Computer Graphics and Visualisation

ITWales

- Introduction
- Research and Development Partnerships
- ITWales Business Club
- ITWales Programme of Special Events
- Student and Graduate Placements
- itwales.com
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Reading List</td>
<td>169</td>
</tr>
<tr>
<td>Tutorials</td>
<td>172</td>
</tr>
<tr>
<td>Assessment and Progression</td>
<td>174</td>
</tr>
<tr>
<td>British Computer Society Accreditation</td>
<td>182</td>
</tr>
<tr>
<td>General Information</td>
<td>184</td>
</tr>
<tr>
<td>Student Obligations - University</td>
<td>186</td>
</tr>
<tr>
<td>Student Obligations - Department</td>
<td>187</td>
</tr>
<tr>
<td>2009/2010 Session Calendar</td>
<td>190</td>
</tr>
</tbody>
</table>
INTRODUCTION TO THE SCHOOL OF PHYSICAL SCIENCES

Welcome to the School of Physical Sciences, Swansea University.

The University's research and teaching are organised into ten Schools, each containing disciplines with common intellectual goals. The School of Physical Sciences is a partnership between Computer Science, Mathematics and Physics. The three Departments have a common academic origin, ethos and are dedicated to curiosity-driven research at the highest international levels. What we do, we try to do very well.

The three Departments enjoy a large degree of autonomy with individual research, teaching and financial priorities reflecting their distinct scientific cultures. The Departments are research intensive and internationally focused. Indeed, we have world leading and internationally excellent research achievements in all three subjects, a fact recognised publicly in the national Research Assessment Exercise 2008.

In a crowded, educated and developing world, our subjects are vitally important for the future. Swansea computer scientists, mathematicians, and physicists have made distinguished contributions to their fields and educated thousands of students to a high level. The School is dedicated to making a serious and substantial contribution to the knowledge and skills of current and future generations.

Although most of your activities, whether undergraduate or postgraduate, will take place within your Department, and the School will have little impact on your daily routines, the School provides a strong foundation for the communication and dissemination of information and ideas and, hence, for the well-being of Science at Swansea.

I very much hope that your studies here will be enjoyable and transforming.

Prof. John V. Tucker
Head of School of Physical Sciences

1 August 2009
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 4. 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:

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

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!

Dr. Phil Grant
Head of Department
Note:
1. The above diagram illustrates the current scheme for the BSc in Computer Science.
2. In Year One students choose modules totalling 10 credit points from Computer Science or other academic fields.
3. The Undergraduate Computer Science Colloquium is not an independently-assessed module, though it contains some assessed components.
STAFF OF THE DEPARTMENT

ACADEMIC STAFF

Professor of Computer Science and Head of School
J. V. Tucker, BA, MSc, PhD, CEng, FBCS

Head of Department, Deputy Head of School and Senior Lecturer
P. W. Grant, BSc, DPhil

Professors
M. Chen, BSc, PhD, FBCS

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.

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

H. Thimbleby, BSc, MSc, PhD, MACM, Hon.FRSA, FIEE, CEng
Human-computer interaction. Future interaction technologies.

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

U. Berger, Dipl-Math, Dr rer nat, PD
Logic, proof theory and applications. Domain theory. Theorem provers and program synthesis.

**Senior Lecturers**

M. W. Jones, BSc, PhD Computer graphics. Volume visualisation and rendering. Image processing.


R. S. Laramee, BSc, MSc, PhD Data visualisation, data analysis, computer graphics. Human-computer interaction.


**Lecturers**

J. E. Blanck, MSc, PhD Theoretical computer science. Logic and computability theory. Domain theory. Continuous data types.

P. Eslambolchilar, BEng, MEng, PhD Mobile Human-Computer Interaction. Gesture recognition, Multimodal interfaces, Dynamics and continuous interaction.

N. A. Harman, BSc, PhD Algebraic specification of microprocessors and other digital hardware. Algebraic specification of languages and environments. Formal specification methodologies.


J. A. Sharp, BSc, PhD Data flow computing. Program design environments. Parallel processing. Functional programming.

M. Wilson, MEng, PhD Human computer interaction. Future interaction technologies.
RCUK Academic 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

Support Staff
D. Arter, BSc
Systems Design/Development, Human Computer Interaction

J. Edwards
Senior Secretary

S. J. Fenn
Secretary

J. D. Pellard
Technician

S. Phillips
Clerical Assistant, Finance Officer

P. H. Roberts-Davies, HNC
Technician

D. Theobald, HNC, BTEC
Technician

ITWales
B. Williams, MBCS
Director

S. Earls, BA
Editor / Marketing Co-ordinator

E. Nash
Placement Co-ordinator

tba
Placements Assistant

C. Jones, BA
Marketing Assistant

M. Kiddell, BSc
Web Developer

A. M. C. Richards, HND, MSc
Administration Manager

C. Williams
Events Co-ordinator

SAW
C. Williams
Programme Manager

M. Kiddell
Technical Services Manager

S. Jones
Communications Manager
A. M. C. Richards Administration Manager
M. Roach Business Services Manager
M. Moller Regional Interface Officer
Hr Allen Clerical Assistant
C. Jones Communications Co-ordinator
H. Robinson Communications Clerical Assistant
E. Waters Finance Support Assistant

**WAG Research Institute of Visual Computing**

**Research Staff**

R. Borgo, BSc, PhD
Visualisation, Functional Programming

R. Byrne, BSc, PhD
Human computer interaction

C. Charron, BSc, Phd
Computer Vision and Medical Image Analysis.

D. Chung
Video processing and visualization.

B. Daubney, BSc, PhD
Image and video processing, computer vision.

H. Fang, BSc, PhD
Image and video processing, computer vision.

A. M. Gimblett, BA, MPhil

R. D. Green, BSc

D. Hegde, BSc

M. Jiang, BSc, PhD
Computer vision, image and video processing

R. Kammaje, BSc, PhD
Computer Graphics.

P. Legg, BSc, PhD
Computer vision, medical imaging, video processing and visualization.

D. Lipsa, BSc
Visualisation.
P. Oladimeji, BSc, MRes Medical device design.
T. Owen, BSc, MSc Information retrieval and document triage.
M. Parry, BSc Video processing and visualization.
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.

To be appointed

To be appointed

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 Abstract data types. Concurrent process theory.
Professor of programming,
University of Amsterdam, The Netherlands.

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

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

Honorary Lecturer
K. Stephenson, BSc, PhD Qinetiq High integrity systems. Programming and specification languages.
Swansea University History of Computing Collection

Prof. J. V. Tucker Chair
Dr. S. R. Williams LIS
Dr. T. Davies Engineering
Dr. C. Evans Physics
Dr. P. W. Grant 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: Dr. Grant (tbc)
Deputy Head of Department: Prof. Chen (tbc)
Director of Teaching: Dr. Sharp
Director of Finance: Dr. Grant
Director of Research: Prof. Mosses
Director of Engagement and Impact: Prof. Jones
Director of Support Services: Dr. Beckman
Coordinator of Undergraduate Admission: Dr. Harman
Coordinator of Postgraduate Admission: Dr. Setzer
Coordinator of Assessment and Progression: Dr. Jones
Coordinator of Industrial Programmes: Mrs. B. Williams
Departmental Senior Secretary: Mrs. Edwards
Departmental Finance Officer: Mrs. Phillips
General Student Enquiries (Room 206): Mrs. Fenn, Mrs. Pellard
Disability Officer: Dr. Seisenberger

Year Heads:
- Foundation Year (Level Zero) Dr. Seisenberger
- First Year (Level One) Dr. Harman
- Second Year (Level Two) Dr. Laramee
- Third Year (Level Three) Dr. Roggenbach
- MEng Year (Level Four) Dr. Harman
- MSc Computing and Software Technology Dr. Stein
- MSc Computer Science (non-specialist) Dr. Stein
- MSc Computing and Future Interaction Technology Dr. Berger
- PhD/MPhil/MRes Prof. Jones

Liaison Officers:
- Engineering, ICCT Dr. Eslambolchilar
- Other Schools/Departments Dr. Blanck
- Exchange/Overseas Students/Erasmus Schemes Dr. Roggenbach
- Research Faculty/Graduate Office Dr. Berger
- Library and Information Services (Library) Dr. Berger
- Library and Information Services (Computing) Mr. Roberts-Davies, Mr. Theobald
- Schools and Colleges Dr. Eslambolchilar
Programme Coordinators:

Departmental Coordinator
BSc Computer Science
BSc Computing and Communications
BSc Computer Science with a Modern Language
BSc Mathematics for Computer Science
BSc Computing with Finance
MEng Computing
MSc/Diploma Computing and Software Technology
MSc/Diploma Computer Science (non specialist)
MSc/MRes Computing and Future Interaction Technology (FIT)
MRes Logic and Computation
MRes Visual Computing
PhD/MPhil

Examinations and Assessment:

Chair of Examining Boards
Examination Board Secretary
Examination Officers
Examination Papers
Examination Results
Student Database
Student Monitoring
Special Needs
Student Records

Recruitment and Admissions:

Recruitment Publications
Recruitment Events

Centre for Computing and Software Technologies (CAST):

Co-Directors
Commercial Manager

Research:

Director
Research Colloquia and Seminars
Research Student Progression
Research Degree Examination
Departmental Research Studentships
Departmental Online Research Reports

Prof. Chen, Dr. Grant
Mr. A. Morris

Prof. Mosses
Dr. Blanck, Dr. Wilson, Dr. Xie
Dr. Berger
Dr. Berger
Dr. Setzer
Prof. Mosses, Mr. Arter
Departmental Handbooks and Brochures:
- Course Handbook
- Online Laboratory Handbook
- BSc Project Handbook
- MEng Project Handbook
- MSc Project Handbook
- Undergraduate Recruitment
- Postgraduate Recruitment

Departmental Computing Facilities:
- Departmental Computing Infrastructure
  (e.g. servers, clusters and backup)
- Laboratories and Office Equipment
  (including cshelp@swansea.ac.uk)
- Laboratory Maintenance and Consumables
- Security System
- Audio/Visual Equipment

Erasmus Exchange with Humboldt-University Berlin:
- Dr. Roggenbach

General Teaching Management
- Director
- Deputy Director
- Programme Reviews and Audits
- University and Subject Statistics
- British Computer Society Accreditation
- Timetables/Lecture Room Bookings
- Departmental Room Booking
- Student Enrollment and Induction
- First Year Events
- Second Year Events
- Undergraduate Colloquium at Gregynog
- Project Demonstration Fair
- Postgraduate Demonstration
- Industrial Panel

ITWales:
- Director
- Project Co-ordinator
- Work Placement Scheme
- Events Coordinator
- Web Developer
- Industrial Advisory Panel
- Editor/Marketing Coordinator
- Marketing Assistant

Finance:

Director
Deputy Director
General Finance Enquiries
Account Management
Expenses Claims
Equipment Purchase
Small Item Purchase

Car Hire, Travel and Accommodation

Infrastrcture, Support Service, Health and Safety

Director
Deputy Director
Space and Estates
Departmental Website

Departmental Administrative Software Tools
Departmental Computing Infrastructure
(e.g. servers, clusters and backup)
Laboratories and Office Equipment
(including cshelp@swansea.ac.uk)
Laboratory Maintenance and Consumables
Security System and Access Control
Software Licences
Audio/Visual Equipment
Data Protection
Qualified First Aiders
Departmental Safety Officers

Project Co-ordinators:

Final Year Undergraduate
MEng Group Project
MSc Computing and Software Technology
MSc Computer Science (non specialists)
MSc/MRes Computing and FIT
MRes Logic and Computation
MRes Visual Computing

Dr. Grant
Dr. Sharp
Mrs. Phillips
Mrs. Phillips
Mrs. Phillips
Mr. Theobold, Mr. Roberts-Davies
Mrs. Pellard, Mrs. Fenn,
Mrs. Edwards
Mrs. Pellard, Mrs. Edwards

Dr. Beckmann
Dr. Jones
Dr. Beckmann
Prof. Mosses, Mr. Arter,
Prof. Jones, Dr. Harman
Mr. Arter

Mr. Theobold, Mr. Roberts-Davies
Mr. Theobold, Mr. Roberts-Davies
Mrs. Pellard
Mr. Theobold, Mr. Roberts-Davies
Mr. Theobold, Mr. Roberts-Davies
Dr. Stein, Mr. Whyley
Mrs. Fenn, Mr. Richards
Dr. Grant, Mr. Roberts-Davies,
Mrs. Pellard

Mr. Whyley, Dr. Roggenbach
Dr. Harman, Dr. Roggenbach
Dr. Stein
Dr. Stein
Prof. M. Jones
Dr. Berger, Prof. Mosses
Dr. Mora

Third Mission Activities and Industrial Programmes:

General Coordinator and IT Wales Director
SAW (Software Alliance Wales) Director
SAW Program Manager
SAW Finance
SAW...

Work Placement Scheme
RIVIC Knowledge Exploitation Officer
KEP Business Computing Analysis

Mrs. B. Williams
Dr. Harman
Mrs. C. Williams
Mr. Richards
Mr. Kiddell, Mrs. Earls,
Miss. C. Jones, Dr. Moller,...
Mrs. Nash
Mr. Morris
Dr. Walton, Mrs. Hegde
CONTACT POINTS: UNIVERSITY

Emergency Services 333

Admissions Office
Undergraduate Admissions 5111 admissions@swansea.ac.uk
Postgraduate Admissions 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
Information Desk 5697 library@swansea.ac.uk
IT Support 5060 itsupport@swansea.ac.uk

Safety Office 5152 safety@swansea.ac.uk

Shops and Services
Level 2 Shop 5473
Post Office 4373
Students' Union Travel Shop 5476
Waterstone's Bookshop 4374 enquiries@swan-uni.waterstones.co.uk
Nursery 3151 nursery@swansea.ac.uk

Student Volunteering Swansea (Discovery) 5743 discovery@swansea.ac.uk

Taliesin Arts Centre
Box Office 2060
Gallery 5526
Egypt Centre 5960

University Sports Centre
Reception 3555
Department of Sport and
Physical Recreation 3552
Wales National Pool 3513

sportcentre@swansea.ac.uk
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 513225 / 513393 / 295826 / 602979
Email: moneydoctors@swan.ac.uk
Web:http://www.swansea.ac.uk/study/current/StudentSupportServices/MoneyAdvice/
Students in financial difficulty should be referred to the Money Advice and Support Office in the Student Support Services Department Ground Floor, Keir Hardie. The office also provides information, advice and guidance in all areas of Student Finance including the following: Student Loans; Student Fees; Higher Education Grant; Welsh Assembly Learning Grants (ALG); Cross Border Funding; Financial Contingency Fund (FCF)-Hardship Grants & Loans; European and International Student Crisis Fund.
We also administer the University's Financial Contingency Fund and can provide short term interest free loans, bursaries and non-repayable grants from this fund. In addition to this, we have staff that are highly experienced in Debt Counselling and offer an excellent service, free of charge to students who may be unable to meet regular repayments to their creditors. To access any of these vital services, the Money Advice and Support Office has drop-in sessions Monday, Tuesday, Thursday and Friday between 10.00am and 12.30pm and with appointments held in the afternoons. Wednesdays are used for administration and service development, except for urgent or emergency cases.
All contacts and communications made between student and advisors will remain in the strictest of confidence.

Wellbeing Services
This is a new unit created in April 2009 to bring together the services of Student Counselling and Mental Health Coordination.

Student Counselling Service
Tel: 01729 295592. 8.30am - 4.30pm Mental Wellbeing Coordination
Tel: 01792 512000
Fax: 01792 295090 Minicom/text: 01792 513 100 Email: s.hookway@swan.ac.uk
The Service offers direct help and support as well as liaison internally with University departments and externally with various agencies. The Coordinator works closely with colleagues in the Disability team where appropriate to ensure students with mental health issues have equality of access to their studies. Students may also be eligible to apply for specialist mentors who can spend time helping them to manage diverse aspects of life in support of their studies.
The website also has links to other websites; one with an extensive range of leaflets on issues students commonly bring to counselling and another to a guide to student mental health for staff that may also be useful to students.

English Language Training Services (ELTS)
Tel: 01792 295391
Email: elts@swansea.ac.uk 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.
UNIVERSITY SUPPORT FOR STUDENTS

Academic Registry
The Academic Registry, located in the Stable Block of Singleton Abbey, administers several key areas that may directly affect your time at Swansea, such as: Examinations; Degree Ceremonies; Erasmus/Socrates Programme; Financial information for North American students (Federal/private loans, Montgomery GI Bill); Transcripts; Diploma Supplements; Suspension/Withdrawal of Studies; Transfers of Programmes/Modules; Appeals; Council Tax Exemption; Unfair Practice/Unfitness to Practice; Welsh Language Provision; Programme Specifications; Assessment; Complaints/Disciplinary.

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.
Tel: 01792 513546
Fax: 01792 295157
Email: academic.registry@swansea.ac.uk
Web: http://www.swan.ac.uk/registry/

Student Support Services
Student Support Services is part of the Student Services Directorate alongside Residential Services (Accommodation Office) 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:
Disability Office; International Student Advisory Service (ISAS); Money Advice and Support Office; Student Counselling 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.

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
The Disability Office both provides and coordinates 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.
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 well informed decisions about the forms of support that will most appropriately meet their needs.

Recording for the Blind Centre
We work closely with colleagues in the Disability Office and Library and Information Centre to provide materials in Braille, large print and/or tape.
Web: http://www.swansea.ac.uk/lis/library/rcfb/
For further information about the range of services on offer, please visit the website or consult the current Support Services Handbook for a summary.
Tel: 01792 513000
Fax: 01792 295090
Minicom/text: 01792 513 100
Email: disability@swansea.ac.uk
Web: http://www.swansea.ac.uk/study/current/StudentSupportServices/DisabilityOffice/

International Student Advisory Service (ISAS)
Tel: 01792 295984 or 01792 602243
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.
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-sessional 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.

Email: student.counselling@swansea.ac.uk
Web: http://www.swansea.ac.uk/counselling/

The University operates a free, confidential counselling service. The service offers counselling to those students who find that they are unable to study effectively or enjoy their life at University, whatever the cause – personal or academic. Most counselling is delivered by the Services’ professionally qualified counsellors, some students may be offered sessions with advanced trainees where appropriate. All work to the Ethical Framework of the British Association for Counselling and Psychotherapy. The Student Counselling Service has moved into Horton House. Appointments can be booked for 9.00am – 4.00pm on weekdays, if these times are not possible arrangements may be made for sessions outside these times. The Service also offers counselling at Carmarthen for students based there.

A copy of the Service booklet is available from the Service or Student Support Services.
SCHOOL OF PHYSICAL SCIENCES AND DEPARTMENTAL COMMITTEES

School of Physical Sciences Management Board:

Head of School
Prof. J. V. Tucker (Computer Science)

Deputy Heads of School
Dr. P. W. Grant (Computer Science. HOD)
Prof. G. M. Shore (Physics. HOD)
Prof. N. Jacob (Mathematics)

Departmental Representatives
Prof. Truman (Mathematics. HOD)
Prof. Hollowood (Physics)

School Officers:

Head of Research
Prof. N. Jacob (Mathematics)

Head of Learning and Teaching
Dr. J. A. Sharp (Computer Science)

Head of Admissions
Dr. N. A. Harman (Computer Science)

Collaborative Provision
Dr. I. M. Davies (Mathematics)

Disability/Equality
Dr. M. Seisenberger (Computer Science)

e-Learning
Prof. M. Jones (Computer Science)

Environmental Co-ordinator
Mr. D. Payne (Physics)

Departmental Committees

Departmental Management Committee:

Head of Department
Dr. Grant (Chair)

Deputy Head of Department
Prof. Chen

Director of Teaching
Dr. Sharp (Secretary)

Director of Research
Prof. Mosses

Director of Finance
Dr. Grant

Director of Engagement and Impact
Prof. Mosses

Director of Support Services
Dr. Beckmann

Coordinator of Undergraduate Admission
Dr. Harman

Coordinator of Postgraduate Admissions
Dr. Setzer

Coordinator of Assessment and Examination
Dr. Jones

Coordinator of Industrial Programmes
Mrs. B. Williams

Head of School
Prof. Tucker

Departmental Research Committee:

Director of Research
Prof. Mosses (Chair)

Head of FIT Group
Prof. Thimbleby

Head of Visual Computing
Prof. Chen

Head of Theoretical Computer Science
Prof. Tucker

PhD/MPhil Year Head and Research Faculty Liaison
Dr. Berger (Secretary)

Ethics and Experimental Risk Assessment
Prof. Jones

Head of Department
Dr. Grant

Co-opted Members:

Director of Finance
Dr. Grant

Director of Support Services
Dr. Beckmann

Research Staff Representative
tba

Research Student Representative
tba
Departmental Learning and Teaching Committee:

Director of Teaching
Deputy Director
Coordinator of Undergraduate Admission
Coordinator of Postgraduate Admission
Coordinator of Assessment and Progression
Students and Candidates Engagement
Student Monitoring
Disability Officer
Head of Department
Deputy Head of Department
Co-opted Members

Dr. Sharp (Chair)
tba
Dr. Harman (Secretary)
Dr. Setzer
Dr. Jones
Prof. Jones
Mr. Whyley
Dr. Seisenberger
Dr. Grant
Prof. Chen
All other teaching staff

Ethics and Risk Assessment Committee

PhD/MPhil Year Head
Research Staff Representative

Prof. Jones (Secretary)
Dr. Laramee
Dr. Wilson
Dr. Berger
tba

Staff/Student Consultative Committee:

Prof. Sharp (Chair)
Mr. Roberts-Davies (Convener)
Mrs. Fenn (Secretary)
All Teaching Staff
All Support Staff
Student Representatives
Representative of LIS

Infrastructure Committee:

Director of Support Services
Deputy Director of Support Services
Director of Finance
Ethics and Risk Assessment
Infrastructure, Laborities, Security
Web and Administrative Software

Dr. Beckmann (Chair)
Dr. Jones (Secretary)
Dr. Grant
Prof. Jones
Mr. Theobold, Mr. Roberts-Davies
Mr. Arter

Engagement and Impact Committee:

Director of Engagement and Impact
Recruitment (Candidates Engagement)
Retention (Students Engagement)
Research Impact
Industrial Engagement
External Contexts (UK, Wales, University, School)
Co-opted Members:

Director of Finance
University and Subject Statistics
Schools and Colleges
PhD/MPhil Admissions
SAW
RIVIC

Prof. Jones (Chair)
Dr. Harman, Dr. Setzer
Dr. Sharp, Dr. Jones
Prof. Mosses, Dr. Berger
Mrs. B. Williams (Secretary)
Prof. Tucker

Dr. Grant
Dr. Wilson, Dr. Xie
Dr. Eslambolchilar, Mr. Whyley
Dr. Kullmann
Mrs. C. Williams
Mr. Morris

21
DEPARTMENTAL COMMITTEE STRUCTURE

Management Committee

- Research Committee
- Learning and Teaching Committee
- Infrastructure Committee
- Ethics and Risk Assessment Committee
- Engagement and Impact Committee

Staff/Student Consultative Committee

External Bodies involved

Other Departments involved

Other Departments involved

involved

involved

involved
<table>
<thead>
<tr>
<th>Name</th>
<th>Room</th>
<th>Tel. Ext. No.</th>
<th>Email address</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Head of Department</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr. P. W. Grant</td>
<td>312</td>
<td>5396</td>
<td><a href="mailto:p.w.grant@swansea.ac.uk">p.w.grant@swansea.ac.uk</a></td>
</tr>
<tr>
<td><strong>Professors</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prof. M. Chen</td>
<td>504</td>
<td>5663</td>
<td><a href="mailto:m.chen@swansea.ac.uk">m.chen@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Prof. Matt Jones</td>
<td>304</td>
<td>3383</td>
<td><a href="mailto:matt.jones@swansea.ac.uk">matt.jones@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Prof. F. G. Moller</td>
<td>213</td>
<td>5160</td>
<td><a href="mailto:f.g.moller@swansea.ac.uk">f.g.moller@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Prof. P. D. Mosses</td>
<td>507</td>
<td>2249</td>
<td><a href="mailto:p.d.mosses@swansea.ac.uk">p.d.mosses@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Prof. H. W. Thimbleby</td>
<td>505</td>
<td>2299</td>
<td><a href="mailto:h.thimbleby@swansea.ac.uk">h.thimbleby@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Prof. J. V. Tucker</td>
<td>309</td>
<td>5649</td>
<td><a href="mailto:j.v.tucker@swansea.ac.uk">j.v.tucker@swansea.ac.uk</a></td>
</tr>
<tr>
<td><strong>Academic Staff</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr. A. Beckmann</td>
<td>511</td>
<td>2370</td>
<td><a href="mailto:a.beckmann@swansea.ac.uk">a.beckmann@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. U. Berger</td>
<td>306</td>
<td>3380</td>
<td><a href="mailto:u.berger@swansea.ac.uk">u.berger@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. J. E. Blanck</td>
<td>Talbot 59</td>
<td>2917</td>
<td><a href="mailto:j.e.blanck@swansea.ac.uk">j.e.blanck@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. P. Eslambolchilar</td>
<td>509</td>
<td>2658</td>
<td><a href="mailto:p.eslambolchilar@swansea.ac.uk">p.eslambolchilar@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. N. A. Harman</td>
<td>301</td>
<td>5394</td>
<td><a href="mailto:n.a.harman@swansea.ac.uk">n.a.harman@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. Mark W. Jones</td>
<td>305</td>
<td>3391</td>
<td><a href="mailto:m.w.jones@swansea.ac.uk">m.w.jones@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. O. Kullmann</td>
<td>212</td>
<td>3369</td>
<td><a href="mailto:o.kullmann@swansea.ac.uk">o.kullmann@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. R. S. Laramee</td>
<td>508</td>
<td>3383</td>
<td><a href="mailto:r.s.laramee@swansea.ac.uk">r.s.laramee@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. B. Mora</td>
<td>303</td>
<td>5575</td>
<td><a href="mailto:b.mora@swansea.ac.uk">b.mora@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. M. Roggenbach</td>
<td>210</td>
<td>3578</td>
<td><a href="mailto:m.roggenbach@swansea.ac.uk">m.roggenbach@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. M. Seisenberger</td>
<td>208</td>
<td>2131</td>
<td><a href="mailto:m.seisenberger@swansea.ac.uk">m.seisenberger@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. A. G. Setzer</td>
<td>211</td>
<td>3368</td>
<td><a href="mailto:a.g.setzer@swansea.ac.uk">a.g.setzer@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. J. A. Sharp</td>
<td>307</td>
<td>5560</td>
<td><a href="mailto:j.a.sharp@swansea.ac.uk">j.a.sharp@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. R. D. Stein</td>
<td>302</td>
<td>3026</td>
<td><a href="mailto:r.d.stein@swansea.ac.uk">r.d.stein@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mr. C. J. Whley</td>
<td>209</td>
<td>3576</td>
<td><a href="mailto:c.j.whley@swansea.ac.uk">c.j.whley@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mr. M. Wilson</td>
<td>510</td>
<td>2611</td>
<td><a href="mailto:m.l.wilson@swansea.ac.uk">m.l.wilson@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Dr. X. Xie</td>
<td>Talbot 56</td>
<td>2916</td>
<td><a href="mailto:x.xie@swansea.ac.uk">x.xie@swansea.ac.uk</a></td>
</tr>
<tr>
<td><strong>Support Staff</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mrs. J. Edwards</td>
<td>313</td>
<td>5393</td>
<td><a href="mailto:j.edwards@swansea.ac.uk">j.edwards@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mr. D. Arter</td>
<td>403</td>
<td>4047</td>
<td><a href="mailto:d.arter@swansea.ac.uk">d.arter@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mrs. S. Fenn</td>
<td>206</td>
<td>5651</td>
<td><a href="mailto:s.j.fenn@swansea.ac.uk">s.j.fenn@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mr. A. Morris</td>
<td>Talbot 55</td>
<td>5416</td>
<td><a href="mailto:a.morris@swansea.ac.uk">a.morris@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mrs. J. D. Pellard</td>
<td>206</td>
<td>4048</td>
<td><a href="mailto:j.d.pellard@swansea.ac.uk">j.d.pellard@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mrs. S. Phillips</td>
<td>313</td>
<td>3204</td>
<td><a href="mailto:s.phillips@swansea.ac.uk">s.phillips@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mr. P. H. Roberts-Davies</td>
<td>403</td>
<td>5007</td>
<td><a href="mailto:p.roberts-davies@swansea.ac.uk">p.roberts-davies@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mr. D. Q. A. Theobald</td>
<td>403</td>
<td>4033</td>
<td><a href="mailto:d.q.a.theobald@swansea.ac.uk">d.q.a.theobald@swansea.ac.uk</a></td>
</tr>
<tr>
<td><strong>ITWales</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mrs. S. Earls</td>
<td>218</td>
<td>3382</td>
<td><a href="mailto:s.e.m.earls@swansea.ac.uk">s.e.m.earls@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mrs. E. Nash</td>
<td>218</td>
<td>5881</td>
<td><a href="mailto:e.nash@swansea.ac.uk">e.nash@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Miss C. Jones</td>
<td>218</td>
<td>2702</td>
<td><a href="mailto:c.e.jones@swansea.ac.uk">c.e.jones@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mr. M. Kiddell</td>
<td>218</td>
<td>4532</td>
<td><a href="mailto:m.kiddell@swansea.ac.uk">m.kiddell@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mr. A. M. C. Richards</td>
<td>218</td>
<td>5806</td>
<td><a href="mailto:a.m.c.richards@swansea.ac.uk">a.m.c.richards@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mrs. B. Williams</td>
<td>207</td>
<td>5625</td>
<td><a href="mailto:b.williams@swansea.ac.uk">b.williams@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Mrs. C. Williams</td>
<td>218</td>
<td>3384</td>
<td><a href="mailto:christine.williams@swansea.ac.uk">christine.williams@swansea.ac.uk</a></td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General Student Enquiries</td>
<td>206</td>
<td>5651</td>
<td><a href="mailto:cshelp@swansea.ac.uk">cshelp@swansea.ac.uk</a></td>
</tr>
<tr>
<td>Departmental Fax</td>
<td></td>
<td>5708</td>
<td></td>
</tr>
</tbody>
</table>

Extension 6xxx can be dialled directly as 01792 606xxx.
Extension 5xxx can be dialled directly as 01792 295xxx.
Extension 3xxx can be dialled directly as 01792 513xxx.
Extension 2xxx can be dialled directly as 01792 602xxx.
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 School or Head of Department students are advised to contact Mrs. Edwards in Room 313.

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 school, 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 three open access laboratories for undergraduate use (one of which is primarily for students undertaking projects). 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 2000 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.
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 2011 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](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, old and 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;

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.

To assist in the mobility of students who study at Swansea University the University will be applying to be granted an ECTS (European Credits Transfer Scheme) label. Throughout this handbook 10 credits from Swansea University are equivalent to 5 ECTS 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

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Name</th>
<th>Educational Aims</th>
<th>Module Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-061</td>
<td>Introduction to Computing I</td>
<td></td>
<td>CS-061</td>
</tr>
<tr>
<td>CS-071</td>
<td>Introduction to Computing II</td>
<td></td>
<td>CS-071</td>
</tr>
<tr>
<td>CS-106</td>
<td>Discrete Mathematics for Computer Science I</td>
<td></td>
<td>CS-106</td>
</tr>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td></td>
<td>CS-108</td>
</tr>
<tr>
<td>CS-113</td>
<td>From Languages to Hardware</td>
<td></td>
<td>CS-113</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computing Systems</td>
<td></td>
<td>CS-116</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td></td>
<td>CS-124</td>
</tr>
<tr>
<td>CS-126</td>
<td>Discrete Mathematics for Computer Science II</td>
<td></td>
<td>CS-126</td>
</tr>
<tr>
<td>CS-141</td>
<td>Programming: Principles and Practice</td>
<td></td>
<td>CS-141</td>
</tr>
<tr>
<td>CS-142</td>
<td>Programming: Performance and Efficiency</td>
<td></td>
<td>CS-142</td>
</tr>
<tr>
<td>CS-144</td>
<td>Software Development: Tools and Techniques</td>
<td></td>
<td>CS-144</td>
</tr>
<tr>
<td>CS-161</td>
<td>Introduction to Computing I</td>
<td></td>
<td>CS-161</td>
</tr>
<tr>
<td>CS-171</td>
<td>Introduction to Computing II</td>
<td></td>
<td>CS-171</td>
</tr>
<tr>
<td>CS-191</td>
<td>Functional Programming I</td>
<td></td>
<td>CS-191</td>
</tr>
<tr>
<td>CS-199</td>
<td>Computers Unplugged</td>
<td></td>
<td>CS-199</td>
</tr>
<tr>
<td>CS-213</td>
<td>System Specification</td>
<td></td>
<td>CS-213</td>
</tr>
<tr>
<td>CS-215</td>
<td>Logic Programming and Artificial Intelligence</td>
<td></td>
<td>CS-215</td>
</tr>
<tr>
<td>CS-217</td>
<td>Computer Graphics I: Image Processing and Synthesis</td>
<td></td>
<td>CS-217</td>
</tr>
<tr>
<td>CS-219</td>
<td>Database Systems</td>
<td></td>
<td>CS-219</td>
</tr>
<tr>
<td>CS-236</td>
<td>Language and Computation</td>
<td></td>
<td>CS-236</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td></td>
<td>CS-242</td>
</tr>
<tr>
<td>CS-248</td>
<td>Concurrent Systems</td>
<td></td>
<td>CS-248</td>
</tr>
<tr>
<td>CS-254</td>
<td>Software Engineering</td>
<td></td>
<td>CS-254</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Name</th>
<th>Educational Aims</th>
<th>Module Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-307</td>
<td>Computer Graphics II: Modelling and Rendering</td>
<td></td>
<td>CS-307</td>
</tr>
<tr>
<td>CS-311</td>
<td>Concepts of Programming Languages</td>
<td></td>
<td>CS-311</td>
</tr>
<tr>
<td>CS-313</td>
<td>High Integrity Systems</td>
<td></td>
<td>CS-313</td>
</tr>
<tr>
<td>CS-317</td>
<td>Computer Graphics Laboratory</td>
<td></td>
<td>CS-317</td>
</tr>
<tr>
<td>CS-318</td>
<td>Cryptography and IT Security</td>
<td></td>
<td>CS-318</td>
</tr>
<tr>
<td>CS-337</td>
<td>Data Visualisation</td>
<td></td>
<td>CS-337</td>
</tr>
<tr>
<td>CS-338</td>
<td>Internet Computing</td>
<td></td>
<td>CS-338</td>
</tr>
<tr>
<td>CS-339</td>
<td>Advanced Topics in Computer Science</td>
<td></td>
<td>CS-339</td>
</tr>
<tr>
<td>CS-344</td>
<td>Project Implementation and Dissertation</td>
<td></td>
<td>CS-344</td>
</tr>
<tr>
<td>CS-345</td>
<td>Artificial Intelligence Applications</td>
<td></td>
<td>CS-345</td>
</tr>
<tr>
<td>CS-348</td>
<td>Building Reliable Web Applications</td>
<td></td>
<td>CS-348</td>
</tr>
<tr>
<td>CS-349</td>
<td>Mobile Interaction Design</td>
<td></td>
<td>CS-349</td>
</tr>
<tr>
<td>CS-354</td>
<td>Project Specification and Development</td>
<td></td>
<td>CS-354</td>
</tr>
<tr>
<td>CS-358</td>
<td>High-Performance Computing in C/C++</td>
<td></td>
<td>CS-358</td>
</tr>
<tr>
<td>CS-364</td>
<td>Software Testing</td>
<td></td>
<td>CS-364</td>
</tr>
<tr>
<td>CS-371</td>
<td>Design Patterns and Generic Programming</td>
<td></td>
<td>CS-371</td>
</tr>
<tr>
<td>CS-375</td>
<td>Logic for Computer Science</td>
<td></td>
<td>CS-375</td>
</tr>
<tr>
<td>CS-377</td>
<td>Fundamentals of Computer Vision</td>
<td></td>
<td>CS-377</td>
</tr>
<tr>
<td></td>
<td>Undergraduate Computer Science Colloquium</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Name</th>
<th>Educational Aims</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M04</td>
<td>Group Project</td>
<td></td>
</tr>
<tr>
<td>CS-M05</td>
<td>Advanced Topics in Logic and Computation</td>
<td></td>
</tr>
<tr>
<td>CS-M07</td>
<td>Data Visualisation</td>
<td></td>
</tr>
<tr>
<td>CS-M08</td>
<td>Future Interaction Technologies: MSc Project</td>
<td></td>
</tr>
<tr>
<td>CS-M09</td>
<td>Future Interaction Technologies: MRes Project</td>
<td></td>
</tr>
<tr>
<td>CS-M13</td>
<td>Critical Systems</td>
<td></td>
</tr>
<tr>
<td>CS-M14</td>
<td>Industrial Project</td>
<td></td>
</tr>
<tr>
<td>CS-M15</td>
<td>Directed Studies in Logic and Computation</td>
<td></td>
</tr>
<tr>
<td>CS-M17</td>
<td>Volume Graphics</td>
<td></td>
</tr>
<tr>
<td>CS-M18</td>
<td>IT Security: Theory and Practice</td>
<td></td>
</tr>
<tr>
<td>CS-M19</td>
<td>Interactive System Design</td>
<td></td>
</tr>
<tr>
<td>CS-M25</td>
<td>Research Methodology and Project Specification</td>
<td></td>
</tr>
<tr>
<td>CS-M29</td>
<td>Mobile Interaction Design</td>
<td></td>
</tr>
<tr>
<td>CS-M32</td>
<td>Software Project</td>
<td></td>
</tr>
<tr>
<td>CS-M34</td>
<td>Algorithm Design and Analysis</td>
<td></td>
</tr>
<tr>
<td>CS-M35</td>
<td>Logic and Computation Project</td>
<td></td>
</tr>
<tr>
<td>CS-M37</td>
<td>Graphics Surveys and Research Methodology</td>
<td></td>
</tr>
<tr>
<td>CS-M39</td>
<td>Interaction Technologies: Seminars and Readings</td>
<td></td>
</tr>
<tr>
<td>CS-M41</td>
<td>Programming in Java</td>
<td></td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Name</th>
<th>Educational Aims</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M49</td>
<td>Interactive Technologies: Lab and Field Work</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M51</td>
<td>Data Storage and Manipulation</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M57</td>
<td>Computer Graphics, Visualisation and Virtual Environments Project</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M58</td>
<td>Distributed O-O Programming</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M59</td>
<td>Relational and Object Oriented Database Systems</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M61</td>
<td>Concepts of Programming Languages</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M64</td>
<td>Computer Software Systems</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M65</td>
<td>Artificial Intelligence Applications</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M67</td>
<td>Graphics Processor Programming</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M68</td>
<td>Writing Web and Web Service Applications</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M69</td>
<td>Interaction Technologies: Information Retrieval</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M71</td>
<td>Design Patterns and Generic Programming</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M74</td>
<td>Software Product Development</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M76</td>
<td>Abstract Data Types and Program Synthesis</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M77</td>
<td>Fundamentals of Computer Vision</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M78</td>
<td>High Performance Computing in C/C++</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M79</td>
<td>Interaction Technologies, Hardware and Devices</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M81</td>
<td>Distributed Programming in Java</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>CS-M97</td>
<td>State of the Art Computer Graphics and Visualisation</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
</tbody>
</table>

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 for Programmes
## Levels One and Two

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Name</th>
<th>Semester 1</th>
<th>Semester 2</th>
<th>Whole Session</th>
<th>Compulsory</th>
<th>Optional</th>
<th>Not Available</th>
<th>Module Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-106</td>
<td>Discrete Mathematics for Computer Science I</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>CS-106</td>
</tr>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>CS-108</td>
</tr>
<tr>
<td>CS-113</td>
<td>From Languages to Hardware</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>1</td>
<td></td>
<td></td>
<td>CS-113</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computer Systems</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>CS-116</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>WS</td>
<td>S2</td>
<td>WS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>CS-124</td>
</tr>
<tr>
<td>CS-126</td>
<td>Discrete Mathematics for Computer Science II</td>
<td>S2</td>
<td>S2</td>
<td>WS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>CS-126</td>
</tr>
<tr>
<td>CS-141</td>
<td>Programming: Principles and Practice</td>
<td>WS</td>
<td>S2</td>
<td>WS</td>
<td>1</td>
<td>1</td>
<td></td>
<td>CS-141</td>
</tr>
<tr>
<td>CS-142</td>
<td>Programming: Performance and Efficiency</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>CS-142</td>
</tr>
<tr>
<td>CS-144</td>
<td>Software Development: Tools and Techniques</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>CS-144</td>
</tr>
<tr>
<td>CS-161</td>
<td>Introduction to Computing I</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>CS-161</td>
</tr>
<tr>
<td>CS-171</td>
<td>Introduction to Computing II</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>CS-171</td>
</tr>
<tr>
<td>CS-191</td>
<td>Functional Programming I</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>CS-191</td>
</tr>
<tr>
<td>CS-199</td>
<td>Computers Unplugged</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>CS-199</td>
</tr>
<tr>
<td>CS-213</td>
<td>System Specification</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>CS-213</td>
</tr>
<tr>
<td>CS-215</td>
<td>Logic Programming and Artificial Intelligence</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>CS-215</td>
</tr>
<tr>
<td>CS-217</td>
<td>Computer Graphics I: Image Processing and Synthesis</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>CS-217</td>
</tr>
<tr>
<td>CS-219</td>
<td>Database Systems</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>CS-219</td>
</tr>
<tr>
<td>CS-236</td>
<td>Language and Computation</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>CS-236</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>CS-242</td>
</tr>
<tr>
<td>CS-248</td>
<td>Concurrent Systems</td>
<td>S1</td>
<td>S1</td>
<td>S1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>CS-248</td>
</tr>
<tr>
<td>CS-254</td>
<td>Software Engineering</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>CS-254</td>
</tr>
</tbody>
</table>

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 | CS-00 | CS-01 | CS-02 | CS-03 | CS-04 | CS-05 | CS-06 | CS-07 | CS-08 | CS-09 | CS-10 | CS-11 | CS-12 | CS-13 | CS-14 | CS-15 | CS-16 | CS-17 | CS-18 | CS-19 | CS-20 | CS-21 | CS-22 | CS-23 | CS-24 |
|-------------|--------------------------------------------------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| CS-307      | Computer Graphics II: Modelling and Rendering    | S1       | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     |       |
| CS-311      | Concepts of Programming Languages                | S2       | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 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     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | CS-313|
| CS-317      | Computer Graphics Laboratory                     | WS       | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | CS-317|
| CS-318      | Cryptography and IT Security                     | S2       | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 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     | 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     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | CS-338|
| CS-339      | Advanced Topics in Computer Science              | V        | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | CS-339|
| CS-344      | Project Implementation and Dissertation          | WS       | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 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     | 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     | 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     | 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     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 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     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | CS-358|
| CS-364      | Software Testing                                 |          |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| CS-371      | Design Patterns and Generic Programming          | S2       | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 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     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | CS-375|
| CS-377      | Fundamentals of Computer Vision                  | S2       | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 3     | 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.
### BSc Computer Science

**Level Zero (Foundation Year-G401 only)**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
<th>Module No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-061</td>
<td>Introduction to Computing I</td>
<td>10</td>
<td>57</td>
</tr>
<tr>
<td>CS-071</td>
<td>Introduction to Computing II</td>
<td>10</td>
<td>58</td>
</tr>
<tr>
<td>MAC051</td>
<td>Foundation of Mathematics 1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>MAC053</td>
<td>Foundation of Mathematics 3</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

80 credits from level 0 modules offered by other departments

**Level One**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
<th>Module No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-106</td>
<td>Discrete Mathematics for Computer Science I</td>
<td>10</td>
<td>61</td>
</tr>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>CS-113</td>
<td>From Languages to Hardware</td>
<td>10</td>
<td>63</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computer Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-126</td>
<td>Discrete Mathematics for Computer Science II</td>
<td>10</td>
<td>66</td>
</tr>
<tr>
<td>CS-141</td>
<td>Programming: Principles and Practice</td>
<td>20</td>
<td>67</td>
</tr>
<tr>
<td>CS-142</td>
<td>Programming: Performance and Efficiency</td>
<td>10</td>
<td>68</td>
</tr>
<tr>
<td>CS-144</td>
<td>Software Development: Tools and Techniques</td>
<td>10</td>
<td>69</td>
</tr>
<tr>
<td>CS-191</td>
<td>Functional Programming I</td>
<td>10</td>
<td>72</td>
</tr>
</tbody>
</table>

**Other Level One Modules**

10

**Level Two**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
<th>Module No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-213</td>
<td>System Specification</td>
<td>10</td>
<td>77</td>
</tr>
<tr>
<td>CS-215</td>
<td>Logic Programming and Artificial Intelligence</td>
<td>10</td>
<td>78</td>
</tr>
<tr>
<td>CS-217</td>
<td>Computer Graphics I: Image Processing and Synthesis</td>
<td>10</td>
<td>79</td>
</tr>
<tr>
<td>CS-219</td>
<td>Database Systems</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>CS-236</td>
<td>Language and Computation</td>
<td>20</td>
<td>81</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td>20</td>
<td>82</td>
</tr>
<tr>
<td>CS-248</td>
<td>Concurrent Systems</td>
<td>20</td>
<td>83</td>
</tr>
<tr>
<td>CS-254</td>
<td>Software Engineering</td>
<td>20</td>
<td>84</td>
</tr>
</tbody>
</table>

**Level Three** - 120 credit points must be taken.

**Compulsory**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
<th>Module No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-344</td>
<td>Project Implementation and Dissertation (Core)</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>CS-354</td>
<td>Project Specification and Development</td>
<td>10</td>
<td>102</td>
</tr>
</tbody>
</table>

**Optional**

(90 credits must be taken from the list on page 49)

* For details of Mathematics modules see the Department of Mathematics
# MEng Computing

## Level One

Same as BSc Computer Science (G400)

## Level Two

Same as BSc Computer Science (G400)

**Level Three** - 120 credit points must be taken

**Compulsory:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-344</td>
<td>Project Implementation and Dissertation (core)</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>CS-354</td>
<td>Project Specification and Development (core)</td>
<td>10</td>
<td>102</td>
</tr>
</tbody>
</table>

90 Level Three credits from Computer Science and Electrical & Electronic Engineering Options (for Computer Science options see page 50).

## Level Four - 120 credit points must be taken

**Compulsory:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M04</td>
<td>Group Project</td>
<td>20</td>
<td>119</td>
</tr>
<tr>
<td>CS-M14</td>
<td>Industrial Project (core)</td>
<td>40</td>
<td>125</td>
</tr>
<tr>
<td>EBGM01</td>
<td>Management Theory</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>EBMM10</td>
<td>Marketing Management</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

40 credits from:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M07</td>
<td>Data Visualisation</td>
<td>10</td>
<td>121</td>
</tr>
<tr>
<td>CS-M13</td>
<td>Critical Systems</td>
<td>10</td>
<td>124</td>
</tr>
<tr>
<td>CS-M17</td>
<td>Volume Graphics</td>
<td>10</td>
<td>127</td>
</tr>
<tr>
<td>CS-M18</td>
<td>IT Security: Theory and Practice</td>
<td>10</td>
<td>128</td>
</tr>
<tr>
<td>CS-M19</td>
<td>Interactive System Design</td>
<td>10</td>
<td>129</td>
</tr>
<tr>
<td>CS-M29</td>
<td>Mobile Interaction Design</td>
<td>10</td>
<td>131</td>
</tr>
<tr>
<td>CS-M32</td>
<td>Algorithm Design and Analysis</td>
<td>10</td>
<td>132</td>
</tr>
<tr>
<td>CS-M58</td>
<td>Distributed O-O Programming</td>
<td>10</td>
<td>142</td>
</tr>
<tr>
<td>CS-M61</td>
<td>Concepts of Programming Languages</td>
<td>10</td>
<td>144</td>
</tr>
<tr>
<td>CS-M65</td>
<td>Artificial Intelligence Applications</td>
<td>10</td>
<td>147</td>
</tr>
<tr>
<td>CS-M68</td>
<td>Writing Web and Web Service Applications</td>
<td>10</td>
<td>149</td>
</tr>
<tr>
<td>CS-M71</td>
<td>Design Patterns and Generic Programming</td>
<td>10</td>
<td>151</td>
</tr>
<tr>
<td>CS-M77</td>
<td>Fundamentals of Computer Vision</td>
<td>10</td>
<td>154</td>
</tr>
<tr>
<td>CS-M79</td>
<td>Interaction Technologies: Hardware and Devices</td>
<td>10</td>
<td>156</td>
</tr>
</tbody>
</table>

**Level Four Business Options**

+ For details of Business modules consult the School of Business and Economics
## BSc Computing and Communications

### Level One

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-106</td>
<td>Discrete Mathematics for Computer Science I</td>
<td>10</td>
<td>61</td>
</tr>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-141</td>
<td>Programming: Principles and Practice</td>
<td>20</td>
<td>67</td>
</tr>
<tr>
<td>CS-142</td>
<td>Programming: Performance and Efficiency</td>
<td>10</td>
<td>68</td>
</tr>
<tr>
<td>CS-144</td>
<td>Software Development: Tools and Techniques</td>
<td>10</td>
<td>69</td>
</tr>
<tr>
<td>EG-143</td>
<td>Digital System Design</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>EG-150</td>
<td>Signals and Systems B</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>EG-189</td>
<td>Engineering Analysis I</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>EG-190</td>
<td>Engineering Analysis II</td>
<td>10</td>
<td>*</td>
</tr>
</tbody>
</table>

### Level Two - 120 credit points must be taken

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-213</td>
<td>System Specification</td>
<td>10</td>
<td>77</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td>20</td>
<td>82</td>
</tr>
<tr>
<td>CS-248</td>
<td>Concurrent Systems</td>
<td>20</td>
<td>83</td>
</tr>
<tr>
<td>CS-254</td>
<td>Software Engineering</td>
<td>20</td>
<td>84</td>
</tr>
<tr>
<td>EG-245</td>
<td>Communications</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>EG-247</td>
<td>Signals and Systems</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>EE-259</td>
<td>Web Application Technology</td>
<td>10</td>
<td>*</td>
</tr>
</tbody>
</table>

*plus any 20 second year Computer Science credits for which the pre-requisites are satisfied*

### Level Three - 120 credit points must be taken

**Compulsory:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-338</td>
<td>Internet Computing</td>
<td>10</td>
<td>95</td>
</tr>
<tr>
<td>CS-344</td>
<td>Project Implementation and Dissertation (core)</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>CS-354</td>
<td>Project Specification and Development</td>
<td>10</td>
<td>102</td>
</tr>
</tbody>
</table>

40 credit points from Computer Science modules listed on page 49

40 credit points from suitable Engineering modules:

*For details of electronics modules contact the School of Engineering.*
**BSc Mathematics for Computer Science**

**Level One**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-106</td>
<td>Discrete Mathematics for Computer Science I</td>
<td>10</td>
<td>61</td>
</tr>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-141</td>
<td>Programming: Principles and Practice</td>
<td>20</td>
<td>67</td>
</tr>
<tr>
<td>CS-142</td>
<td>Programming: Performance and Efficiency</td>
<td>10</td>
<td>68</td>
</tr>
<tr>
<td>CS-191</td>
<td>Functional Programming I</td>
<td>10</td>
<td>72</td>
</tr>
<tr>
<td>MAC151</td>
<td>Calculus I</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>MAC153</td>
<td>Calculus II</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>MAC161</td>
<td>Algebra I</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>MAC163</td>
<td>Algebra II</td>
<td>10</td>
<td>*</td>
</tr>
</tbody>
</table>

**Level Two**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-213</td>
<td>System Specification</td>
<td>10</td>
<td>77</td>
</tr>
<tr>
<td>CS-215</td>
<td>Logic Programming and Artificial Intelligence</td>
<td>10</td>
<td>78</td>
</tr>
<tr>
<td>CS-236</td>
<td>Language and Computation</td>
<td>20</td>
<td>81</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td>20</td>
<td>82</td>
</tr>
<tr>
<td>MAM211</td>
<td>Groups, Rings and Fields</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>MAP211</td>
<td>Co-ordinate Geometry</td>
<td>10</td>
<td>*</td>
</tr>
</tbody>
</table>

40 credits from appropriate Computer Science and Mathematics modules

**Level Three - 120 credit points must be taken**

**Compulsory**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-344</td>
<td>Project Implementation and Dissertation (core)</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>CS-354</td>
<td>Project Specification and Development</td>
<td>10</td>
<td>102</td>
</tr>
<tr>
<td>MAP442</td>
<td>Applied Algebra</td>
<td>20</td>
<td>*</td>
</tr>
</tbody>
</table>

Optional (60 credit points) from modules listed on page 49 and Appropriate Mathematics Options

* For details of mathematics modules contact the Department of Mathematics.
### BSc Computing with Finance

#### Level One
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-106</td>
<td>Discrete Mathematics for Computer Science I</td>
<td>10</td>
<td>61</td>
</tr>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-126</td>
<td>Discrete Mathematics for Computer Science II</td>
<td>10</td>
<td>66</td>
</tr>
<tr>
<td>CS-141</td>
<td>Programming: Principles and Practice</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>CS-142</td>
<td>Programming: Performance and Efficiency</td>
<td>10</td>
<td>68</td>
</tr>
<tr>
<td>EC-112</td>
<td>Current Issues in Economics</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

*One of the following modules depending upon A-level qualifications:*

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC-100</td>
<td>Principles of Economics A (core)</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>EC-103</td>
<td>Principles of Economics B (core)</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

#### Level Two
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-219</td>
<td>Database Systems</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td>20</td>
<td>82</td>
</tr>
<tr>
<td>CS-248</td>
<td>Concurrent Systems</td>
<td>20</td>
<td>83</td>
</tr>
<tr>
<td>CS-254</td>
<td>Software Engineering</td>
<td>20</td>
<td>84</td>
</tr>
<tr>
<td>EC-200</td>
<td>Intermediate Microeconomics</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>EC-246</td>
<td>Managerial Finance</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>EBG202</td>
<td>Statistics for Business and Economics</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

#### Level Three - 120 credit points must be taken

*Compulsory:*

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-344</td>
<td>Project Implementation and Dissertation (core)</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>CS-334</td>
<td>Project Specification and Development</td>
<td>10</td>
<td>102</td>
</tr>
<tr>
<td>EC-338</td>
<td>Financial Economics</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>EC-339</td>
<td>Financial Market Efficiency</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

*40 - 60 credits from:*

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-313</td>
<td>High Integrity Systems</td>
<td>10</td>
<td>91</td>
</tr>
<tr>
<td>CS-318</td>
<td>Cryptography and IT Security</td>
<td>10</td>
<td>93</td>
</tr>
<tr>
<td>CS-338</td>
<td>Internet Computing</td>
<td>10</td>
<td>94</td>
</tr>
<tr>
<td>CS-345</td>
<td>Artificial Intelligence Applications</td>
<td>10</td>
<td>99</td>
</tr>
<tr>
<td>CS-348</td>
<td>Building Reliable Web Applications</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>CS-349</td>
<td>Mobile Interaction Design</td>
<td>10</td>
<td>101</td>
</tr>
</tbody>
</table>

*10 - 30 credits from appropriate Economics modules*

*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)*

**Level One**

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-106</td>
<td>Discrete Mathematics for Computer Science I</td>
<td>10</td>
<td>61</td>
</tr>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-141</td>
<td>Programming: Principles and Practice</td>
<td>20</td>
<td>67</td>
</tr>
<tr>
<td>CS-142</td>
<td>Programming: Performance and Efficiency</td>
<td>10</td>
<td>68</td>
</tr>
<tr>
<td>CS-144</td>
<td>Software Development: Tools and Techniques</td>
<td>10</td>
<td>69</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-213</td>
<td>System Specification</td>
<td>10</td>
<td>77</td>
</tr>
<tr>
<td>CS-219</td>
<td>Database Systems</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td>20</td>
<td>82</td>
</tr>
<tr>
<td>CS-254</td>
<td>Software Engineering</td>
<td>20</td>
<td>84</td>
</tr>
</tbody>
</table>

Modern Language 30

*plus any 30 second year Computer Science credits for which the pre-requisites are satisfied*

**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:**

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-344</td>
<td>Project Implementation and Dissertation (core)</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>CS-354</td>
<td>Project Specification and Development</td>
<td>10</td>
<td>102</td>
</tr>
</tbody>
</table>

Modern Language 30

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

*For details of Language modules consult the School of Arts and Humanities*
**BSc Computer Science and Pure Mathematics**

<table>
<thead>
<tr>
<th>Level One</th>
<th>credit points</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-106 Discrete Mathematics for Computer Science 1</td>
<td>10</td>
<td>61</td>
</tr>
<tr>
<td>CS-108 Computers and Computing</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>CS-116 Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124 Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-161 Introduction to Computing I</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>CS-171 Introduction to Computing II</td>
<td>10</td>
<td>71</td>
</tr>
<tr>
<td>CS-191 Functional Programming I</td>
<td>10</td>
<td>72</td>
</tr>
<tr>
<td>MAA171 Computational Methods</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAC151 Calculus 1</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAC153 Calculus 2</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAC161 Algebra 1</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAC163 Algebra 2</td>
<td>10</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Level Two**

*Compulsory:*

<table>
<thead>
<tr>
<th>Course</th>
<th>credit points</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-236 Language and Computation</td>
<td>20</td>
<td>81</td>
</tr>
<tr>
<td>CS-242 Algorithms</td>
<td>20</td>
<td>82</td>
</tr>
<tr>
<td>MAA251 Numerical Methods</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAC211 Vector Spaces</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAC213 Inner Product Spaces</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAC221 Analysis 1</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAC223 Analysis 2</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAM211 Groups, Rings and Fields</td>
<td>10</td>
<td>✓</td>
</tr>
</tbody>
</table>

20 credit points from suitable Computer Science and Mathematics modules:

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

*Compulsory:*

<table>
<thead>
<tr>
<th>Course</th>
<th>credit points</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC321 Analysis 3</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAP343 Algebraic Coding Theory</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>MAP345 Rings, Modules and Categories</td>
<td>10</td>
<td>✓</td>
</tr>
</tbody>
</table>

Selection of between 40 and 80 credits from level 3 Computer Science modules from page 49, and the remainder chosen from Mathematics

* For details of mathematics modules contact the Department of Mathematics.
### BSc Computer Science and Geoinformatics (GF48)

#### Level One

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-116</td>
<td>Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-161</td>
<td>Introduction to Computing I (core)</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>CS-171</td>
<td>Introduction to Computing II</td>
<td>10</td>
<td>71</td>
</tr>
<tr>
<td>GEG101</td>
<td>Physical Geography: The Natural Environment I</td>
<td>10</td>
<td>♦</td>
</tr>
<tr>
<td>GEG102</td>
<td>Physical Geography: The Natural Environment II</td>
<td>10</td>
<td>♦</td>
</tr>
<tr>
<td>GEG103</td>
<td>Physical Geography: The Natural Environment III</td>
<td>10</td>
<td>♦</td>
</tr>
<tr>
<td>GEG104</td>
<td>Human Geography: Society and Space I</td>
<td>10</td>
<td>♦</td>
</tr>
<tr>
<td>GEG105</td>
<td>Human Geography: Society and Space II</td>
<td>10</td>
<td>♦</td>
</tr>
<tr>
<td>GEG106</td>
<td>Human Geography: Society and Space III</td>
<td>10</td>
<td>♦</td>
</tr>
<tr>
<td>GEG107</td>
<td>Geographical Methods I</td>
<td>10</td>
<td>♦</td>
</tr>
<tr>
<td>GEG108</td>
<td>Geographical Methods II</td>
<td>10</td>
<td>♦</td>
</tr>
</tbody>
</table>

#### Level Two

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-215</td>
<td>Logic Programming and Artificial Intelligence</td>
<td>10</td>
<td>78</td>
</tr>
</tbody>
</table>

60 credit points from Geoinformatics

50 credit points from Computer Science

#### Level Three - 120 credit points must be taken

**Compulsory:**

- Geoinformatics 60 ♦

60 credit points from Computer Science modules listed on page 49

* For details of topographic science modules contact the School of Environment and Society.
# BSc Computer Science and Psychology

## Level One
<table>
<thead>
<tr>
<th>Module No.</th>
<th>Module Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-161</td>
<td>Introduction to Computing I</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>CS-171</td>
<td>Introduction to Computing II</td>
<td>10</td>
<td>71</td>
</tr>
<tr>
<td>CS-191</td>
<td>Functional Programming I</td>
<td>10</td>
<td>72</td>
</tr>
<tr>
<td>PS-109</td>
<td>Experimental Methods for Joint Honours Students</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PS-111</td>
<td>Cognitive Psychology</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PS-112</td>
<td>Social Psychology</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PS-113</td>
<td>Psychobiology</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PS-114</td>
<td>Individual and Abnormal Psychology</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PS-115</td>
<td>Developmental Psychology</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

## Level Two
<table>
<thead>
<tr>
<th>Module No.</th>
<th>Module Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-215</td>
<td>Logic Programming and Artificial Intelligence</td>
<td>10</td>
<td>78</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td>20</td>
<td>82</td>
</tr>
<tr>
<td>PS-231</td>
<td>Memory, Attention and Learning</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>PS-233</td>
<td>Social Psychology</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>PS-234</td>
<td>Language and Lifespan Development</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>PS-249</td>
<td>Experimental methods and Statistics</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

*30 credit points from suitable Computer Science and Psychology modules*

## Level Three

### Compulsory:
<table>
<thead>
<tr>
<th>Module No.</th>
<th>Module Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-345</td>
<td>Artificial Intelligence Applications</td>
<td>10</td>
<td>99</td>
</tr>
<tr>
<td>PS-325</td>
<td>Project for Joint Honours</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>PS-395</td>
<td>Brain &amp; Behaviour</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>PS-396</td>
<td>Individual Differences (B)</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

*55 credit points from suitable Computer Science and Psychology modules*

Computer Science modules are listed on page 49

**Note:** This course is no longer available to new applicants.

* For details of psychology modules contact the School of Human Sciences.
### BSc COMPUTER SCIENCE AND PHYSICS

**Level One**

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Title</th>
<th>Credits</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-108</td>
<td>Computers and Computing</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>CS-116</td>
<td>Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124</td>
<td>Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-161</td>
<td>Introduction to Computing I</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>CS-171</td>
<td>Introduction to Computing II</td>
<td>10</td>
<td>71</td>
</tr>
<tr>
<td>CS-191</td>
<td>Functional Programming I</td>
<td>10</td>
<td>72</td>
</tr>
<tr>
<td>PH-101</td>
<td>Dynamics</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH-102</td>
<td>Vibrations and Waves</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH-104</td>
<td>Introduction to Astronomy and Cosmology</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH-108</td>
<td>Experimental Techniques II</td>
<td>20</td>
<td>✓</td>
</tr>
</tbody>
</table>

*Other Level One Modules*

**Level Two**

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Title</th>
<th>Credits</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-217</td>
<td>Computer Graphics I: Image Processing and Synthesis</td>
<td>10</td>
<td>79</td>
</tr>
<tr>
<td>CS-242</td>
<td>Algorithms</td>
<td>20</td>
<td>82</td>
</tr>
<tr>
<td>PH-201</td>
<td>Atomic and Nuclear Physics</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH-202</td>
<td>Mechanics and Special Relativity</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH-203</td>
<td>Statistical and Thermal Physics</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH-204</td>
<td>Introduction to Physics Simulation</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH-207</td>
<td>Solid State Physics</td>
<td>10</td>
<td>✓</td>
</tr>
</tbody>
</table>

*40 credits from any Level two Computer Science modules for which the pre-requisites are satisfied*

**Level Three** - 120 credit points must be taken

**Compulsory:**

<table>
<thead>
<tr>
<th>Module Code</th>
<th>Module Title</th>
<th>Credits</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH207</td>
<td>Solid State Physics</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH301</td>
<td>Electromagnetism</td>
<td>10</td>
<td>✓</td>
</tr>
<tr>
<td>PH317</td>
<td>Project (20 credits)</td>
<td>20</td>
<td>✓</td>
</tr>
</tbody>
</table>

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

(for Computer Science options see page 49)

* 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).

<table>
<thead>
<tr>
<th>Level One</th>
<th>credit points</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-116 Modelling Computing Systems</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>CS-124 Computers and Society</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>CS-161 Introduction to Computing I</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>CS-171 Introduction to Computing II</td>
<td>10</td>
<td>71</td>
</tr>
</tbody>
</table>

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)

CS-219 Database Systems 10 80

110 further credit points including 20-40 Computer Science credit points depending on individual programmes

(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 50.
### Level Three Modules for Various Degree Programmes

<table>
<thead>
<tr>
<th>Level Three</th>
<th>points</th>
<th>no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-307 Computer Graphics II: Modelling and Rendering</td>
<td>10</td>
<td>89</td>
</tr>
<tr>
<td>CS-311 Concepts of Programming Languages</td>
<td>10</td>
<td>90</td>
</tr>
<tr>
<td>CS-313 High Integrity Systems</td>
<td>10</td>
<td>91</td>
</tr>
<tr>
<td>CS-318 Cryptography and IT Security</td>
<td>10</td>
<td>93</td>
</tr>
<tr>
<td>CS-337 Data Visualisation</td>
<td>10</td>
<td>94</td>
</tr>
<tr>
<td>CS-338 Internet Computing</td>
<td>10</td>
<td>95</td>
</tr>
<tr>
<td>CS-339 Advanced Topics in Computer Science</td>
<td>10</td>
<td>96</td>
</tr>
<tr>
<td>CS-345 Artificial Intelligence Applications</td>
<td>10</td>
<td>99</td>
</tr>
<tr>
<td>CS-348 Building Reliable Web Applications</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>CS-349 Mobile Interaction Design</td>
<td>10</td>
<td>101</td>
</tr>
<tr>
<td>CS-358 High-Performance Computing in C/C++</td>
<td>10</td>
<td>104</td>
</tr>
<tr>
<td>CS-364 Software Testing</td>
<td>10</td>
<td>105</td>
</tr>
<tr>
<td>CS-371 Design Patterns and Generic Programming</td>
<td>10</td>
<td>106</td>
</tr>
<tr>
<td>CS-375 Logic for Computer Science</td>
<td>10</td>
<td>107</td>
</tr>
<tr>
<td>CS-377 Fundamentals of Computer Vision</td>
<td>10</td>
<td>108</td>
</tr>
<tr>
<td>MS-306 New Screen Technologies</td>
<td>20</td>
<td>109</td>
</tr>
<tr>
<td>MS-353 Digital Philosophy: The Roots of the Virtual</td>
<td>20</td>
<td>110</td>
</tr>
<tr>
<td>HIP300 The Classical Tradition in the Sciences</td>
<td>10</td>
<td>112</td>
</tr>
<tr>
<td>HIP301 From Natural Philosophy to Science</td>
<td>10</td>
<td>113</td>
</tr>
</tbody>
</table>

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**

**Part 1**

**Compulsory Modules**

**Semester 1**
- CS-M32 Algorithm Design and Analysis 10 132
- CS-M41 Programming in Java 20 138
- CS-M51 Data Storage and Manipulation 10 140

**Whole Session**
- CS-M64 Computer Software Systems 30 145

**Optional Modules (50 credits must be taken)**

**Semester 1**
- CS-M13 Critical Systems 10 124
- CS-M58 Distributed Object-Oriented Programming 10 142
- CS-M59 Relational and Object-Oriented Database Systems 10 143
- CS-M69 Interaction Technologies: Information Retrieval 10 150

**Semester 2**
- CS-M18 IT Security: Theory and Practice 10 128
- CS-M29 Mobile Interaction Design 10 131
- CS-M65 Artificial Intelligence Applications 10 147
- CS-M68 Writing Web and Web-Service Applications 10 149
- CS-M71 Design Patterns and Generic Programming 10 151
- CS-M79 Interaction Technologies: Hardware and Devices 10 156

**Whole Session**
- CS-M19 Interactive System Design 10 129

**Part 2**
- CS-M34 Software Project 60 133

*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

<table>
<thead>
<tr>
<th>Compulsory Modules</th>
<th>credit points</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-sessional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-M01 Distributed Programming in Java</td>
<td>20</td>
<td>123</td>
</tr>
<tr>
<td>Whole Session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-M74 Software Product Development</td>
<td>20</td>
<td>162</td>
</tr>
<tr>
<td>Summer Vacation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-M34 Software Project</td>
<td>60</td>
<td>139</td>
</tr>
<tr>
<td>Optional Modules - students take 80 credits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semester 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-M07 Data Visualisation</td>
<td>10</td>
<td>126</td>
</tr>
<tr>
<td>CS-M13 Critical Systems</td>
<td>10</td>
<td>129</td>
</tr>
<tr>
<td>CS-M32 Algorithm Design and Analysis</td>
<td>10</td>
<td>138</td>
</tr>
<tr>
<td>CS-M51 Data Storage and Manipulation</td>
<td>10</td>
<td>149</td>
</tr>
<tr>
<td>CS-M58 Distributed O-O Programming</td>
<td>10</td>
<td>153</td>
</tr>
<tr>
<td>CS-M59 Relational and Object-Oriented Database Systems</td>
<td>10</td>
<td>154</td>
</tr>
<tr>
<td>CS-M69 Interaction Technologies: Information Retrieval</td>
<td>10</td>
<td>160</td>
</tr>
</tbody>
</table>

| Semester 2                              |               |          |
| CS-M18 IT Security: Theory and Practice | 10            | 134      |
| CS-M29 Mobile Interaction Design        | 10            | 137      |
| CS-M61 Concepts of Programming Languages| 10            | 155      |
| CS-M65 Artificial Intelligence Applications | 10 | 158 |
| CS-M68 Writing Web and Web Service Applications | 10 | 159 |
| CS-M71 Design Patterns and Generic Programming | 10 | 161 |
| CS-M77 Fundamentals of Computer Vision  | 10            | 164      |
| CS-M79 Interaction Technologies: Hardware and Devices | 10 | 165 |

| Whole Session                           |               |          |
| CS-M19 Interactive System Design        | 10            | 135      |

*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 & Future Interaction Technologies

Compulsory Modules

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M08</td>
<td>MSc Project</td>
<td>60</td>
<td>122</td>
</tr>
<tr>
<td>CS-M19</td>
<td>Interactive Systems Design</td>
<td>10</td>
<td>129</td>
</tr>
<tr>
<td>CS-M25</td>
<td>Research Methodology and Project Specification</td>
<td>20</td>
<td>130</td>
</tr>
<tr>
<td>CS-M29</td>
<td>Mobile Interaction Design</td>
<td>10</td>
<td>131</td>
</tr>
<tr>
<td>CS-M39</td>
<td>Interaction Technologies: Seminars &amp; Readings</td>
<td>10</td>
<td>137</td>
</tr>
<tr>
<td>CS-M49</td>
<td>Interaction Technologies: Lab &amp; Field Work</td>
<td>10</td>
<td>139</td>
</tr>
<tr>
<td>CS-M69</td>
<td>Interaction Technologies: Information Retrieval</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td>CS-M79</td>
<td>Interaction Technologies: Hardware &amp; Devices</td>
<td>10</td>
<td>156</td>
</tr>
</tbody>
</table>

Optional modules - students take 40 credits

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M07</td>
<td>Data Visualisation</td>
<td>10</td>
<td>121</td>
</tr>
<tr>
<td>CS-M13</td>
<td>Critical Systems</td>
<td>10</td>
<td>124</td>
</tr>
<tr>
<td>CS-M17</td>
<td>Volume Graphics</td>
<td>10</td>
<td>127</td>
</tr>
<tr>
<td>CS-M18</td>
<td>IT Security: Theory and Practice</td>
<td>10</td>
<td>128</td>
</tr>
<tr>
<td>CS-M32</td>
<td>Algorithm Design and Analysis</td>
<td>10</td>
<td>132</td>
</tr>
<tr>
<td>CS-M58</td>
<td>Distributed O-O Programming</td>
<td>10</td>
<td>142</td>
</tr>
<tr>
<td>CS-M61</td>
<td>Concepts of Programming Languages</td>
<td>10</td>
<td>144</td>
</tr>
<tr>
<td>CS-M65</td>
<td>Artificial Intelligence Applications</td>
<td>10</td>
<td>147</td>
</tr>
<tr>
<td>CS-M68</td>
<td>Writing Web and Web Service Applications</td>
<td>10</td>
<td>149</td>
</tr>
<tr>
<td>CS-M71</td>
<td>Design Patterns and Generic Programming</td>
<td>10</td>
<td>151</td>
</tr>
<tr>
<td>CS-M77</td>
<td>Fundamentals of Computer Vision</td>
<td>10</td>
<td>154</td>
</tr>
</tbody>
</table>

MRRes / Certificate
Computing & Future Interaction Technologies

Compulsory Modules

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M09</td>
<td>Research Project (includes 10 credits from lab &amp; field work)</td>
<td>120</td>
<td>123</td>
</tr>
<tr>
<td>CS-M25</td>
<td>Research Methodology and Project Specification</td>
<td>20</td>
<td>130</td>
</tr>
<tr>
<td>CS-M39</td>
<td>Interaction Technologies: Seminars &amp; Readings</td>
<td>10</td>
<td>137</td>
</tr>
<tr>
<td>CS-M49</td>
<td>Interaction Technologies: Lab &amp; Field Work</td>
<td>10</td>
<td>139</td>
</tr>
</tbody>
</table>

Optional Modules - students take 30 credits

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credit Points</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M19</td>
<td>Interactive Systems Design</td>
<td>10</td>
<td>129</td>
</tr>
<tr>
<td>CS-M29</td>
<td>Mobile Interaction Design</td>
<td>10</td>
<td>131</td>
</tr>
<tr>
<td>CS-M69</td>
<td>Interaction Technologies: Information Retrieval</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td>CS-M79</td>
<td>Interaction Technologies: Hardware &amp; Devices</td>
<td>10</td>
<td>156</td>
</tr>
</tbody>
</table>

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

**Logic and Computation**

<table>
<thead>
<tr>
<th>Compulsory Modules</th>
<th>credit points</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M05 Advanced Topics in Logic and Computation</td>
<td>10</td>
<td>120</td>
</tr>
<tr>
<td>CS-M15 Directed Studies in Logic and Computation</td>
<td>10</td>
<td>126</td>
</tr>
<tr>
<td>CS-M25 Research Methodology and Project Specification</td>
<td>10</td>
<td>130</td>
</tr>
<tr>
<td>CS-M35 Logic and Computation Project</td>
<td>120</td>
<td>135</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optional Modules - students take 30 credits</th>
<th>credit points</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M13 Critical Systems</td>
<td>10</td>
<td>124</td>
</tr>
<tr>
<td>CS-M18 IT Security: Theory and Practice</td>
<td>10</td>
<td>128</td>
</tr>
<tr>
<td>CS-M61 Concepts of Programming Languages</td>
<td>10</td>
<td>144</td>
</tr>
<tr>
<td>CS-M65 Artificial Intelligence Applications</td>
<td>10</td>
<td>147</td>
</tr>
<tr>
<td>CS-M71 Design Patterns and Generic Programming</td>
<td>10</td>
<td>151</td>
</tr>
</tbody>
</table>

**MRes/Certificate**

**Visual Computing**

| CS-M07 Data Visualisation                              | 10            | 121      |
| CS-M37 Graphics Surveys and Research Methodology       | 10            | 136      |
| CS-M57 Computer Graphics, Visual Computing Project     | 120           | 141      |
| CS-M67 Graphics Processor Programming                  | 10            | 148      |
| CS-M77 Fundamentals of Computer Vision                 | 10            | 154      |
| CS-M97 State of the Art Computer Graphics and Visualisation | 10        | 158      |

<table>
<thead>
<tr>
<th>Optional Modules - students take 10 credits</th>
<th>credit points</th>
<th>page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-M17 Volume Graphics</td>
<td>10</td>
<td>127</td>
</tr>
<tr>
<td>CS-M49 Interaction Technologies: Lab and Field Work</td>
<td>10</td>
<td>139</td>
</tr>
<tr>
<td>CS-M69 Interaction Technologies: Information Retrieval</td>
<td>10</td>
<td>150</td>
</tr>
</tbody>
</table>

*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

<table>
<thead>
<tr>
<th>UCAS Code</th>
<th>programme</th>
<th>UWS Route Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>G400</td>
<td>BSc Computer Science</td>
<td>XCSCS</td>
</tr>
<tr>
<td>G401</td>
<td>BSc Computer Science (with foundation year)</td>
<td>FCSCS</td>
</tr>
<tr>
<td>G403</td>
<td>MEng Computing</td>
<td>4CSPS</td>
</tr>
<tr>
<td>G420</td>
<td>BSc Computing and Communications</td>
<td>XCSOS</td>
</tr>
<tr>
<td>G4GC</td>
<td>BSc Mathematics for Computer Science</td>
<td>XMACS</td>
</tr>
<tr>
<td>G4R1</td>
<td>BSc Computer Science with a Modern Language (French)</td>
<td>XCSCWFRF</td>
</tr>
<tr>
<td>G4R2</td>
<td>BSc Computer Science with a Modern Language (German)</td>
<td>XCSCWGRM</td>
</tr>
<tr>
<td>G4R3</td>
<td>BSc Computer Science with a Modern Language (Italian)</td>
<td>XCSCWITA</td>
</tr>
<tr>
<td>G4R4</td>
<td>BSc Computer Science with a Modern Language (Spanish)</td>
<td>XCSCWHSS</td>
</tr>
<tr>
<td>G4Q5</td>
<td>BSc Computer Science with a Modern Language (Welsh)</td>
<td>T/XCSCWCYM</td>
</tr>
<tr>
<td>G4G4</td>
<td>BSc Computer Science and Pure Mathematics</td>
<td>XCSCSMAP</td>
</tr>
<tr>
<td>GF48</td>
<td>BSc Computer Science and Geoinformatics</td>
<td>XCSCAGEI</td>
</tr>
<tr>
<td>CG84</td>
<td>BSc Computer Science and Psychology</td>
<td>XCSCAPSS</td>
</tr>
<tr>
<td>FG34</td>
<td>BSc Computer Science and Physics</td>
<td>XCSAPHY</td>
</tr>
<tr>
<td>G720</td>
<td>BSc Business Information Technology</td>
<td>SXEBYS</td>
</tr>
<tr>
<td>G4L1</td>
<td>BSc Computing with Finance</td>
<td>XCSHS</td>
</tr>
<tr>
<td>R1G4</td>
<td>BA One Modern Language with Computer Studies (French)</td>
<td>SFREWWCSS</td>
</tr>
<tr>
<td>R2G4</td>
<td>BA One Modern Language with Computer Studies (German)</td>
<td>SGRMWCSS</td>
</tr>
<tr>
<td>R3G4</td>
<td>BA One Modern Language with Computer Studies (Italian)</td>
<td>SITAWCSS</td>
</tr>
<tr>
<td>R4G4</td>
<td>BA One Modern Language with Computer Studies (Spanish)</td>
<td>SHSSWCSS</td>
</tr>
<tr>
<td>Q5G4</td>
<td>BA One Modern Language with Computer Studies (Welsh)</td>
<td>XCYMWCSS</td>
</tr>
<tr>
<td>RRD2</td>
<td>BA Two Modern Languages with Computer Studies (French and German)</td>
<td>SFREWGRMWCSS</td>
</tr>
<tr>
<td>RRD3</td>
<td>BA Two Modern Languages with Computer Studies (French and Italian)</td>
<td>SFREWITAWCSS</td>
</tr>
<tr>
<td>RRD4</td>
<td>BA Two Modern Languages with Computer Studies (French and Spanish)</td>
<td>SFREWHSWCSS</td>
</tr>
<tr>
<td>RQC5</td>
<td>BA Two Modern Languages with Computer Studies (French and Welsh)</td>
<td>SFREWCYWMWSC</td>
</tr>
<tr>
<td>RRG3</td>
<td>BA Two Modern Languages with Computer Studies (German and Italian)</td>
<td>SGRMWITAWCSS</td>
</tr>
<tr>
<td>RRG4</td>
<td>BA Two Modern Languages with Computer Studies (German and Spanish)</td>
<td>SGRMWHSWCSS</td>
</tr>
<tr>
<td>RQG5</td>
<td>BA Two Modern Languages with Computer Studies (German and Welsh)</td>
<td>SGRMWYMWCSS</td>
</tr>
<tr>
<td>RRJ4</td>
<td>BA Two Modern Languages with Computer Studies Italian and Spanish)</td>
<td>SITAWHSWCSS</td>
</tr>
<tr>
<td>RQH5</td>
<td>BA Two Modern Languages with Computer Studies Italian and Welsh)</td>
<td>SITAWCYWMWSC</td>
</tr>
<tr>
<td>RQK5</td>
<td>BA Two Modern Languages with Computer Studies (Spanish and Welsh)</td>
<td>SHSSWCYWMWSC</td>
</tr>
<tr>
<td>H641</td>
<td>BSc Internet Technology</td>
<td>XEEIS</td>
</tr>
<tr>
<td>H646</td>
<td>BSc Mobile Communications and Internet Technology</td>
<td>XEEBS</td>
</tr>
<tr>
<td>H645</td>
<td>BSc E-Commerce Technology</td>
<td>XEEETS</td>
</tr>
<tr>
<td>H6G4</td>
<td>BEng Electronics with Computing Science</td>
<td>F/I/SXEEMS</td>
</tr>
<tr>
<td>H6GK</td>
<td>MEng Electronics with Computing Science</td>
<td>I/SEEEMS</td>
</tr>
</tbody>
</table>
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:**
INTRODUCTION TO COMPUTING II

Synopsis:
This module is a continuation of the module Introduction to Computing I. It is designed to enable students who are not majoring in Computer Science to enhance their skills in programming so they will be able to apply computing knowledge to their other studies. It will also provide a basic understanding of algorithms and data structures. It can also provide (along with other Level 0 modules) a suitable preparation for Level 1 Computer Science.

Syllabus:
Advanced programming techniques and applications.
Introduction to algorithms – sorting and searching;
simple complexity analysis.
Overview of various design techniques.
Introduction to Data Structures – stacks, queues, lists, trees, maps

Learning Outcomes:
Students will be able design, and implement programs to the standard required for Computer Science modules offered to non-computer science majors. They will be able to develop advanced programs for applications related to their main degree subject. They will have a basic knowledge of algorithm design and analysis, and will be able to select and implement standard data structures. 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:
DISCRETE MATHEMATICS FOR COMPUTER SCIENCE I

Synopsis:
The aims of this module are to introduce a vocabulary for modern mathematics and computer science, and to develop knowledge of topics in discrete mathematics.

Syllabus:
Logic: Statements and logical operations. What is the meaning of implication? Logical identities and truth tables.
Quantifiers: For all, there exists. Negation of quantifiers.
Functions: Examples of functions, domain, codomain, one-to-one and onto. Inverse functions. The definition of function used is an input set, output set and black box operation. Composition of functions (using the strict categorical definition with domain = codomain). The inclusion map of a subset in a set can be viewed as a type conversion map.
Relations: Relations as logical statements. Transitive relations and equivalence relations. Partitions of sets. Partial and total orders. The transitive closure of a relation. Relations considered as subsets of the Cartesian product.
Counting: 1-1 correspondences. Finite and infinite sets. There are more real numbers than integers. Power sets. The "who shaves the barber?" problem, and Bertrand Russell's paradox.
Binary operations: Commutativity, associativity, distribution. Axioms or formal rules for a system.
Groups: The rules. Examples, permutation groups, integers, integers mod n under addition, symmetries of a triangle and square.
Rings: The rules. Examples, the integers, the real numbers, the rationals. New rings from old, the matrix construction.

Learning Outcomes:
Students will be familiar with a fundamental vocabulary for modern mathematics and computer science. They will have an enhanced understanding of various topics in discrete mathematics.

Transferable Skills:
Mathematical modelling and analysis. Problem solving.

Reading:
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:
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-decode-execute 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

Course Texts:
**Modelling Computing Systems**

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

*Syllabus:*
**Motivation:** The role of modelling in (software) engineering. A survey of system failures.

**Mathematical Background:** A review of sets, functions and relations; propositional and predicate logic; directed graphs; and induction. Games and winning strategies.

**Modelling Computing Systems:** States and Labelled Transition Systems (LTSs). A simple language for modelling systems, and its translation into LTSs.

**Reasoning about Computations:** A simple modal logic for expressing properties of systems. Distinguishing between systems via the bisimulation game.

**Learning Outcomes:**
The student will be familiar with some mathematical design methods, primarily as applied to requirements analysis, specification and documentation. The student will gain skills in scientific modelling: abstraction, the precise formulation of informal notions, rigorous reasoning and analysis.

**Transferable Skills:**

**Course Text:**

**Reading:**
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. 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. The impact of these issues on the development and testing of software, and also including software project management, 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:

**DISCRETE MATHEMATICS FOR COMPUTER SCIENCE II**

<table>
<thead>
<tr>
<th>Code:</th>
<th>CS-126</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>10</td>
</tr>
<tr>
<td>Taught:</td>
<td>Semester 2</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>20 lectures, 10 problem classes</td>
</tr>
<tr>
<td>Lecturers:</td>
<td>Dr. Beggs (Mathematics - 20 hours lectures), Dr. Blanck (Problem Classes)</td>
</tr>
<tr>
<td>Co-requisite:</td>
<td>CS-141 or CS-161</td>
</tr>
<tr>
<td>Assessment:</td>
<td>30% paper based exercises, 70% written examination (in May/June)</td>
</tr>
</tbody>
</table>

**Synopsis:**
This module gives the student some mathematical techniques that are important in applications of computing, and it deepens the student's knowledge of the logical techniques taught in MAM111.

**Syllabus:**
Sequences, series, induction, recursive definitions, recursive algorithms, recurrence relations, difference equations.
Combinatorics: Permutations and combinations, binomial coefficients, multinomial coefficients. The connection with probability.
Vectors and Matrices in 2 and 3 dimensions: Multiplication, inverses.
Transformations in the Plane: Two-dimensional trigonometry, polar co-ordinates, rotations, reflections, shears, matrix interpretations.

Co-ordinates in Three Dimensions: Planes and lines.
Axioms: Groups and rings as examples, and maps between them.
Truth and proof: Deductions from the axioms.

**Learning Outcomes:**
Students will be able to construct simple proofs by induction, and solve simple combinatorial problems. They will be able to solve small systems of linear equations, invert small matrices, and know of the existence of more general algorithms. They will be able to work with axiomatic definitions, and have an idea of deduction from the axioms. Also the student's geometrical intuition will have improved and been made more precise.

**Transferable Skills:**
Mathematical modelling and analysis. Problem solving.

**Course Texts:**
PROGRAMMING: PRINCIPLES AND PRACTICE

Synopsis:
The aim of this course is to teach students to program in the Object-Oriented programming language Java. The students will be able to understand the fundamental principles underlying imperative programming languages. Students will be able to write programs in Java to solve a variety of simple problems. They will be able to critically analyse the quality of their solutions.

Syllabus:

Learning Outcomes:
The ability to write programs to solve specific problems. The ability to read programs written by other people. The ability to reason about, and analyse the performance of algorithms by experimentation.

Transferable Skills:
Problem solving and analysis.

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.
**Programming: Performance and Efficiency**

**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. Practical exercises linked to the topics discussed in this module will form part of the co-requisite module CS-141 Programming: Principles and Practice.

**Syllabus:**
Introduction to the concept of algorithm and program efficiency.

Introduction to Abstract Data Types including Stacks, Queues, Lists, Tables.

Basic Sorting and Searching Algorithms.

Impact of Memory Management on program performance.

**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 datatypes.

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

Students will be aware of how memory is managed and how this can affect program performance.

**Transferable Skills:**
Simple mathematical reasoning.

Problem solving.

**Course Texts:**


**SOFTWARE DEVELOPMENT: TOOLS AND TECHNIQUES**

**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 Development Strategies.
Agile Programming/Extreme Programming.
Specifying a program test set.
Program debugging tools and debugging strategies.
Basic unit testing and tools for unit testing.
Version control systems.

**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:**
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.

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 be able to develop straightforward programs for applications related to their (non-Computer Science) majors. They will learn to implement programs to the standard required for Level 2 Computer Science modules offered to non-Computer Science majors.

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

Reading:
**INTRODUCTION TO COMPUTING II**

**Synopsis:**
This module is a continuation of the module Introduction to Computing I. It is designed to enable students who are not majoring in Computer Science to enhance their skills in programming so they will be able to apply computing knowledge to their other studies. It will also provide a basic understanding of algorithms and data structures.

**Syllabus:**
Advanced programming techniques and applications. Introduction to algorithms - sorting and searching; simple complexity analysis. Overview of various design techniques. Introduction to Data Structures - stacks, queues, lists, trees, maps.

**Learning Outcomes:**
Students will be able design, and implement programs to the standard required for Level 2 Computer Science modules offered to non-computer science majors. They will be able to develop advanced programs for applications related to their main degree subject. They will have a basic knowledge of algorithm design and analysis, and will be able to select and implement standard data structures.

**Transferable Skills:**
Problem solving. Ability to learn and use computer systems and software packages effectively.

**Reading:**
**Functional Programming 1**

**Synopsis:**
This module introduces the student to functional programming both as a useful and practical programming paradigm and as a formally sound model of computation. The main teaching language is Haskell.

**Syllabus:**
The functional 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.
Controlling input and output by actions.

**Learning Outcomes:**
Students will be able to specify and write programs in a functional language. They will be able to solve simple algorithmic problems using functional concepts. Students will have an appreciation of the reliability and productivity of functional programming.

**Transferable Skills:**
Problem solving, abstract modelling.

**Reading:**
COMPUTERS UNPLUGGED

Synopsis:
This module gives an overview of the main principles underlying computers and computer science 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:
General introduction to the principles underlying computers and computer science. A selection of lecture topics will be chosen from the following list:
• Representing data (numbers, data compression, error detection)
• Representing procedures (programming languages, automata)
• Algorithms (sorting and searching, network routing)
• Complexity (graph coloring, optimisation)
• Cryptography (public keys, digital signatures)
• Human-Computer Interaction (interface design, intelligent machines)
• Visual and Interactive Computing (information visualisation, visual representations of data)

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:
Ability to understand the terminology of computer science, problem solving.

Reading:
## LEVEL ONE AWAYDAY

<table>
<thead>
<tr>
<th>Code:</th>
<th>CS-124</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>see CS-124</td>
</tr>
<tr>
<td>Taught:</td>
<td>Semester 1</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>Half day</td>
</tr>
<tr>
<td>Coordinator:</td>
<td>Mr. Whyley</td>
</tr>
<tr>
<td>Assessment:</td>
<td>Attendance at the Awayday is considered as part of the coursework for CS-124 and a mark will be awarded accordingly</td>
</tr>
</tbody>
</table>

### 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.
LEVEL TWO MODULES
SYSTEM SPECIFICATION

Synopsis:
This module provides an introduction to the application of formal methods to the production of correct system specifications, with particular reference to hardware systems. It introduces students to the practice of formal software specification.

Syllabus:
Motivation: Why we need formal methods. Examples of fiascos and problems.
Algebra and Term Rewriting: How we can implement algebraic models in software.
Introduction to Maude: The Maude algebraic language, with examples of simple software and hardware specifications.
Microprocessor Modelling in Maude: How to represent microprocessors and similar systems algebraically?
Microprocessor Examples: A range of examples including the PDP-8 and Gordon's Computer.
Correctness: How can we tell if an implementation of a specification is correct?
Software Examples: Using Maude to specify software systems.

Learning Outcomes:
Students will be familiar with the principles of algebraic modelling techniques, and their implementation in the Maude programming language. They will have seen simple illustrative examples in software and hardware, microprocessors of moderate complexity. They will have been exposed to the concept of verifying correctness.

Transferable Skills:
Mathematical modelling and analysis. Abstract modelling and formal reasoning. Ability to evaluate and deploy new technologies.

Reading:
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.


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:

Reading:
COMPUTER GRAPHICS I: IMAGE PROCESSING AND SYNTHESIS

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, kdtree, 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.

Course Texts:
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.

Course Texts:

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:


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:

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.

Course Texts:
**CONCURRENT SYSTEMS**

**Synopsis:**
An introduction to the issues raised in developing and using concurrent and distributed systems. Discussion of practical and formal solutions to example problems from operating systems, networking, web applications (and other distributed software systems) and multi-core processors.

**Syllabus:**
Background and motivation — Examples from Operating Systems and Dual Processors etc
Concept of Concurrency — Including mutual exclusion, semaphores etc
Problems in concurrent systems — Deadlock; Livelock
Need for formal solutions — Pathological cases
Practical Solutions — Java Threading model (monitors); Synchronisation, locking, waiting and notifying
Solutions to real problems — Operating Systems examples; Readers and Writers; Concurrency in SMT (‘hyperthreaded’) and CMT (‘multi-core’) processors
Distributed Systems — Networking Issues; Web Issues; Transactions; Sockets
Core concepts behind Operating systems (Virtual Memory, File Systems, Scheduling, Process and Threads, IPCs, etc.).

**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.

**Course Text:**
SOFTWARE ENGINEERING

<table>
<thead>
<tr>
<th>Code:</th>
<th>CS-254</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>20</td>
</tr>
<tr>
<td>Taught:</td>
<td>Whole session</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>30 lectures, 5 discussion meetings</td>
</tr>
<tr>
<td>Lecturers:</td>
<td>Dr. Laramee, Dr Wilson</td>
</tr>
<tr>
<td>Pre-requisite:</td>
<td>CS-141, CS-142, CS-124, CS-144</td>
</tr>
<tr>
<td>Assessment:</td>
<td>100% continuous assessment</td>
</tr>
</tbody>
</table>

Synopsis:
This module exposes the student to the major components of a practical software life-cycle 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. Building on the programming knowledge and experience acquired by the students in the pre-requisites, this module introduces students to the 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 through 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: General Software Engineering and Project Management:
Review of 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 (taught in conjunction with Part C)

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 (in conjunction with Part C)

Part C: Object-oriented Software Engineering:
Requirement analysis and specification, Object oriented software architecture design, GUI design and development (in conjunction with Part B), 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.

Assignments:
1. Group work: system evaluation [15%]
2. Individual work: system redesign [15%]
3. Group work: OO software construction: phase 1 (to be handed to a different group) [25%]
4. Group work: OO software construction: phase 2 (after taking over from a different group) [25%]
5. Individual work: software re-engineering [20%]
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 GUI; 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. The ability to read critically, to precis and judge information. 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:
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.


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:**

**Reading:**


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 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:
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:
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:

Reading:
COMPUTER GRAPHICS LABORATORY

Synopsis:
This module builds upon the materials presented in modules CS-217 and CS-307, and provides students with an opportunity to further their knowledge of computer graphics, and develop their hand-on skills in graphics programming. Whilst the module focuses on the OpenGL API as the main platform for the study, it also provides scope for students to explore other graphics environments. The assessment includes two pieces of programming-based coursework, an investigative report, and an oral presentation. The report and presentation are to be completed in a simulated research conference setting.

Syllabus:
Graphics Programming (Main Study): 3D object modelling; geometrical transformation; simple texture mapping; 3D viewing and projection; illumination and shading; atmospheric effects; anti-aliasing.

Graphics Programming (Advance Study): display list; texture coordinates, multi-texturing; tessellators, quadrics, evaluators and NURBS; shadows.

Graphics Programming (Investigative Study): ray tracing (e.g., Pov-Ray); web-graphics (e.g., X3D and VRML); game engines and environments (e.g., Horde3D, Ogre3D, Irrlicht, Blender); similar API (e.g., Direct3D), major OpenGL extensions (e.g. OpenSceneGraph), GPU programming (e.g., Vertex Shader).

Investigative Study: Each student will undertake an independent investigation, involving problem specification, self-learning, solution identification, and result analysis. In particular, the student will compare the two approaches taken in the two programming-based assignments, write an investigative report in the form of a conference paper, and present the paper in a 25 minute oral presentation to an audience.

Learning Outcomes:
Students will acquire an understanding of computer graphics systems and development methodologies and hand-on experience of computer graphics programming. Their confidence and competence in system analysis and design, and fluency in using programming languages and tools will be improved.

Transferable Skills:
Written communication and documentation, oral presentation and interactive discussions. Time management. Problem solving. Ability to manage learning processes.

Course Texts:
Cryptography and IT Security

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:

Reading:
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:
Additional reading materials will be distributed during lectures.
INTERNET COMPUTING

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 synchronous 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.java.sun.com
The course notes (supplied) contain numerous links to sources of additional information.
### ADVANCED TOPICS IN COMPUTER SCIENCE

#### MODEL-BASED DESIGN OF EMBEDDED SYSTEMS

<table>
<thead>
<tr>
<th><strong>Code:</strong></th>
<th>CS-339</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Credit Points:</strong></td>
<td>10</td>
</tr>
<tr>
<td><strong>Taught:</strong></td>
<td>Semester 1</td>
</tr>
</tbody>
</table>
| **Contact Hours:** | Block course  
19/11/10 - 21/11/10 and  
26/11/10 - 28/11/10 |
| **Lecturers:** | Prof. H. Schlingoff |
| **Co-ordinator:** | Dr. Roggenbach |
| **Assessment:** | 30% presentation,  
70% dissertation |

**Synopsis:**

This module will introduce students to the challenge of security of embedded system design.

**Syllabus:**

Model-based development is an important paradigm for the design of embedded systems as we find them, e.g., in cars, trains, airplanes, satellites, medical equipment or automation. Starting from an abstract model capturing the requirements for a control system, model-based development derives an implementation model by refinement and/or transformation. From such an implementation model it is possible to automatically generate executable code in the target language. Parallel to the development process runs the quality assurance which validate the models with automatically generated test cases.

This module discusses the foundations of model based design and discusses technologies and tools by the means of practical examples.

**Learning Outcomes:**

In general, students will gain a thorough appreciation of a leading edge or state-of-the-art topic in Computer Science from a leading researcher or practitioner. They will have researched an aspect of the topic in detail and will be able to present and defend a report on it.

Specifically in this year, students will learn about embedded systems (special-purpose computer system which are embedded as part of a complete device including hardware and mechanical parts) and the special challenges that the security of such systems involves.

**Transferable Skills:**

Written communication and documentation. Oral Presentation.

**Reading:**


S T. Karris: *Introduction to Stateflow with Applications* Orchard Publications 2007
**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.

**Description:**

**Dissertation.**

*Deadline: Semester 2, Week 11, Wednesday May 18th 10am.*

The dissertation (40-100 pages) is a comprehensive and self-contained report on the work done on the project (see CS-334). For internal students the module CS-334 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 195 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:

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 CLIPS 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:
J. L. Kolodner, Case Based Reasoning, Morgan Kauffman, 1993.
# BUILDING RELIABLE WEB APPLICATIONS

<table>
<thead>
<tr>
<th>Code:</th>
<th>CS-348</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>10</td>
</tr>
<tr>
<td>Taught:</td>
<td>Semester 2</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>20 lectures</td>
</tr>
<tr>
<td>Lecturer:</td>
<td>Dr. Harman</td>
</tr>
<tr>
<td>Incompatables:</td>
<td>CS-M68</td>
</tr>
<tr>
<td>Assessment:</td>
<td>30% coursework, 70% written examination (in May/June)</td>
</tr>
</tbody>
</table>

## 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:
**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.

**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:**
**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. An additional member of staff will be assigned to moderate the marking of documents. 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.**

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 16th November 2010.*
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 (15th - 17th November 2010).

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.
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, Thursday May 12th 2010.*
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 195 for details. Students who are unsure as to whether their
project meets the requirements should discuss it with their supervisor.
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 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:
Reading:
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.
• 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:
**Design Patterns and Generic Programming**

<table>
<thead>
<tr>
<th>Code:</th>
<th>CS-371</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>10</td>
</tr>
<tr>
<td>Taught:</td>
<td>Semester 2</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>20 lectures</td>
</tr>
<tr>
<td>Incompatible:</td>
<td>CS-M71</td>
</tr>
<tr>
<td>Lecturer:</td>
<td>Mr. Whyley</td>
</tr>
<tr>
<td>Assessment:</td>
<td>20% coursework, 80% written examination (in May/June)</td>
</tr>
</tbody>
</table>

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

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 system.
• 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:
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:


**Synopsis:**
This module introduces students to the fundamental topics of computer vision, including image processing, segmentation, feature extraction, motion analysis, object tracking, and recognition.

**Syllabus:**
This course is divided in four parts: Introduction, Image Processing, Video Analysis and Applications.

Introduction: The first lecture gives an overview of image processing and computer vision (IPCV) and a road show of this course. The second lecture provides a revision of basic and important mathematical techniques frequently used in IPCV.

Image processing: edge detection, Hough transform, scale space representation, object extraction, shape recognition, image filtering, image segmentation, texture analysis, colour analysis.

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

Applications: face detection and recognition, medical application of computer vision, real time computer vision.

**Learning Outcomes:**
Students will become familiar with the fundamental concepts of computer vision, 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:**


NEW SCREEN TECHNOLOGIES

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:
**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.
- 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:**


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


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


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

THE CLASSICAL TRADITION IN THE SCIENCES

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 and the rationality of other forms of the scientific enterprise. So too can the image of science as, in some sense, an apolitical activity divorced from its social and cultural settings. This module will study scientific institutions, theories, and methods in Antiquity and demonstrate how these - shaped and were shaped by society and culture.

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:
FROM NATURAL PHILOSOPHY TO SCIENCE

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 well 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:

Code: HIP301
Credit Points: 10
Taught: Semester 1
Contact Hours: 10 lectures plus 2 seminars
Lecturer: Dr. Mosely (History and Classics)
Assessment: 100% written examination (January)
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 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-334);
• 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.
**Project Demonstration Fair**

**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.

<table>
<thead>
<tr>
<th>Taught:</th>
<th>Semester 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>see CS-334</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>Half a day for the fair plus a lecture on poster design</td>
</tr>
<tr>
<td>Organisers:</td>
<td>Dr. Roggenbach, Dr Laramee</td>
</tr>
<tr>
<td>Assessment:</td>
<td>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-334</td>
</tr>
</tbody>
</table>
LEVEL M MODULES
GROUP PROJECT

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.
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:


Additional reading materials will be distributed during lectures.
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.

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.
**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.
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:


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 Texts:

Reading:
INDUSTRIAL PROJECT

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.
**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.
VOLUME GRAPHICS

Synopsis:
Volume Graphics is a newly-emerging sub-field of computer graphics. The aim of the module is to study a range of techniques for modelling, rendering and manipulating volumetric data types.

Syllabus:
Volumetric Data Types: Scalar fields, regular and irregular data types. Objects in volumetric data. Typical applications of volume graphics (including medical imaging and scientific visualisation).


Volume Deformation: Distortion and morphing. Applications in forensic science.

Other Techniques: Volume graphics hardware. Volume-based animation.

Learning Outcomes:
Students will be aware of the theoretical foundations and applications of volume graphics, and gain an understanding of the main techniques for modelling, manipulating and rendering true 3D spatial representations. They will acquire skills in designing computer graphics algorithms.

Transferable Skills:
Problem solving. Information retrieval, ability to read critically, to précis and judge information, and ability to manage learning processes. Ability to evaluate and deploy new technologies. Oral presentation, and interactive discussions.

Course Texts:


Reading:
A collection of research papers (compiled by lecturers).
IT SECURITY: THEORY AND PRACTICE

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:
Reading:
**Interactive System 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.

**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:**
RESERVE METHDOLOGY AND PROJECT SPECIFICATION

Synopsis:
In this module students 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 are introduced into their project by their supervisor. Guidance is given on appropriate research methodologies and students are given appropriate research tasks. They write a detailed specification of their project explaining the background and the aims of the project.

The project specification also describes one or more possible methods with which the project is to be carried out and contains a time plan. The project specification is assessed by the supervisor and, independently by another lecturer.

Learning Outcomes:
Students will have researched into a number of areas related to the topic of their research 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. They will have gained an understanding of appropriate research methodologies.

Transferable Skills:
Problem solving, literature research, planning.

Reading:
Recommended texts may vary.
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.

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:

<table>
<thead>
<tr>
<th>Code</th>
<th>CS-M29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points</td>
<td>10</td>
</tr>
<tr>
<td>Taught</td>
<td>Semester 2</td>
</tr>
<tr>
<td>Contact Hours</td>
<td>20 lectures</td>
</tr>
<tr>
<td>Lecturer</td>
<td>Prof. Jones</td>
</tr>
<tr>
<td>Incompatables</td>
<td>CS-349</td>
</tr>
<tr>
<td>Assessment</td>
<td>30% coursework, 70% written examination (in May/June)</td>
</tr>
</tbody>
</table>
Synopsis:
This module will introduce students to various general techniques for algorithmically solving computational problems, as well as equip them with tools to analyse problems to determine their inherent difficulty.

Syllabus:
Algorithmic Design Techniques: divide-and-conquer (min-max, mergesort, closest-points, matrix multiplication); greedy algorithms (making change, minimum spanning trees, activity selection, data compression); dynamic programming (making change revisited, matrix chain multiplication, longest common subsequence, all-pairs shortest paths).

Analysis Techniques for Lower Bounds: comparison trees for information-theoretic lower bounds; adversary techniques; linear and polynomial reductions.

Randomised Algorithms: Las Vegas algorithms versus Monte Carlo algorithms; quicksort, primality testing, polynomial identities.

Learning Outcomes:
The students will
• gain a thorough knowledge of several advanced problem solving paradigms (divide-and-conquer, greedy algorithms, dynamic programming, randomisation);
• a clear understanding of which paradigms to consider when faced with new problems.

Moreover, the students will have studied the main analysis techniques for proving lower bounds for algorithmic problems, and thus learn the intrinsic difficulty of various problems.

Transferable Skills:
Skills in designing solutions to general problems, and in presenting clear arguments as to the validity of the solutions.

Reading:

SOFTWARE PROJECT

Synopsis:
The project will:- provide students with the opportunity of exploring a particular topic in computer science in some considerable depth; provide the opportunity of specifying, designing and implementing a complete system and experiencing the major phases of the life-cycle of an IT project; enhance students' competence in system design, algorithm analysis and mathematical reasoning, and their fluency in using programming languages and tools; 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.

Syllabus:
During the Summer the student will be supervised by a member of staff. Additionally, a different member of staff will be assigned to assess the marking of documents and the student's progress. The project work will build upon the research undertaken in CS-M74 or CS-M64.

Project Dissertation.

Maximum Length: 20,000 words.
Deadline: 15th October, 12 months after start of full-time course, or 3 years after start of part-time course.

Students who fail to submit their dissertation by 15th October will be awarded a 0% mark and will be disqualified from obtaining a degree with distinction. Such students may resubmit on one further occasion (within six months) and re-submission will be capped at 50%. A document will be provided giving details for the submission of project dissertations. The dissertation is a comprehensive and self-contained report on the work done on the project. The document 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 and mathematical prerequisites;
• Technical problems considered and methods used to solve them;
• Discussion of issues arising in specifying, designing and implementing the system (e.g. requirements analysis, user interface, system architecture, algorithms, major data structure, etc.);
• Evaluation of results (e.g. complexity, efficiency, user-friendliness, reliability, comparison with other work, etc.);
• User and system manual;
• Progress and achievements of the project;
• Suggestions for further work.
Submission of Project Dissertation:
Students should ensure that their dissertations contain the following:
Title page.
Summary page (maximum of 300 words).
Declaration and statements page.
Contents page.
References.
The main body of the dissertation, in the form of numbered chapters, sections and sub-sections, should be between the contents and reference pages. Optional appendices may also be added at the end where appropriate.
For final submission we require two bound copies (software binding available from the Academic Registry Print Room is acceptable) and one CD with the full source code / system and include an electronic copy of the dissertation (PDF version preferred). An electronic copy in either .pdf or Microsoft Word format must be submitted to the Blackboard turnitin site.
A more detailed guideline document for the conduct of the project and the format and submission of the dissertation is available from the module co-ordinator.

Learning Outcomes:
Students will have researched into a number of related areas of Computer Science and one or more application areas. They will have designed and implemented a significant piece of software and documented its development. They will have critically compared and evaluated their work in the context of other similar or comparable developments in the field.

Transferable Skills:
Problem solving; literature searching; report writing; comparative analysis.
LOGIC AND COMPUTATION PROJECT

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.

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
**Graphics Surveys and Research Methodology**

**Synopsis:**
This module will allow students to explore a particular topic in Computer Graphics to a great depth. Students will practice using various methods to acquire knowledge, will undertake the writing of a scientific survey paper, and will present their work to an audience of knowledgeable academics within the chosen area.

**Syllabus:**
Verbal communication, presentation skills, literature searches on a specific topic or set of topics in Computer Graphics, scientific writing, project development, managing research, motivation.

**Learning Outcomes:**
Students will have:
- demonstrated detailed understanding of a topic or set of topics in Computer Graphics;
- the ability to review and critically assess the literature on a specific topic or set of topics in Computer Graphics that are 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:**
Students will demonstrate an ability to:
- engage and present their own academic work to other professionals in their own field;
- study autonomously, and become fully aware of resources to aid the study, of research topics in Computer Graphics;
- manage their own research effectively (in terms of time, direction of study, sources of information and relevance to the topic).

**Course Texts:**
- ACM Transaction on Graphics
- ACM Transactions on Graphical Tools
- Computer Graphics Forum
- IEEE Computer Graphics and Applications
- Computer Vision and Image Understanding

Other journals and conference proceedings (including SIGGRAPH, IEEE Visualisation, and Eurographics).
**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.

**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:**
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.

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

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:

| Code:    | CS-M49 |
| Credit Points: | 10 |
| Taught:     | Whole session |
| Contact Hours: | 15 lectures |
| Lecturer:   | Dr. Wilson |
| Assessment: | 100% continuous assessment |

This module is normally only available to FIT Masters students in the Department of Computer Science. It may be available to other Masters students taking FIT modules.
**Synopsis:**

The module will enable students to:

- Determine the appropriate data structures and data management techniques in a variety of software applications.
- Formally specify a range of abstract data types.
- Understand the implications of the object-oriented paradigm on the specification of data types and on data management.
- Understand the implications of working in a distributed environment on data management.
- Implement a range of data structures and data management techniques in different ways, and be able to determine the efficiency and complexity implications of the different approaches.

The object-oriented language Java will be used for practical implementations.

**Syllabus:**

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.

**Transferable Skills:**

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

**Course Texts:**


---

**DATA STORAGE AND MANIPULATION**

<table>
<thead>
<tr>
<th>Code:</th>
<th>CS-M51</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>10</td>
</tr>
<tr>
<td>Pre-requisite:</td>
<td>CS-M01</td>
</tr>
<tr>
<td>Taught:</td>
<td>Semester 1</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>20 lectures</td>
</tr>
<tr>
<td>Lecturer:</td>
<td>Dr. Sharp</td>
</tr>
<tr>
<td>Assessment:</td>
<td>20% coursework,</td>
</tr>
<tr>
<td></td>
<td>80% written examination</td>
</tr>
<tr>
<td></td>
<td>(in January)</td>
</tr>
</tbody>
</table>

The object-oriented language Java will be used for practical implementations.
**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).
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:
The following web site contains substantial tutorial information:
http://www.java.sun.com
The course notes (supplied) contain numerous links to sources of additional information.
RELATIONAL AND OBJECT-ORIENTED 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 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:
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:

Reading:
COMPUTER SOFTWARE SYSTEMS

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. Students will study the principles of hardware and software design including formal specification methods. Professional Issues and Ethics will be considered. Students will undertake practical work in C and Linux as well as participating in a group project. They will also plan and prepare for their individual Masters Project.

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.
Machine instructions: characteristics, representation format. Types of operations and operands. Addressing modes. Little-, Big-, and Bi-Endian.
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 principles of professional software development;
Professional Issues (Ethics, Legal Issues, Copyright, Intellectual Property Rights, Data Protection Act, Professional Codes of Conduct and Practice (BCS));
Software Process Models (e.g. waterfall, spiral).
The Capability-Maturity Model;
Software Project Management including Requirements Specification and Quality Assurance;
Team work;
the Linux system;
C programming;
Planning an individual project;

Code: CS-M64
Credit Points: 30
Taught: Whole Session
Contact Hours: 40 lectures
Lecturers: Dr. Stein, Dr. Sharp
Incompatible: CS-M74, CS-M25
Assessment: 55% continuous assessment, 45% written examination (in May/June)
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 fully aware of the major professional issues associated with software development;
- worked together on a significant software project and have a thorough understanding of the advantages and disadvantages of team working;
- be fully familiar with many aspects of managing software projects;
- have enhanced their knowledge through guided learning;
- have prepared an individual project plan, including a significant literature review;
- be competent in using the programming language C and in using the Linux operating system;
- 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:**
Effective use of general IT facilities; Appreciating the need for professional standards in software engineering; effective information-retrieval skills; team working; experience of report writing; problem solving.

**Reading:**
ARTIFICIAL INTELLIGENCE APPLICATIONS

Synopsis:
The module details the fundamentals of knowledge based systems, and presents expert system shells such as CLIPS. It also introduces the student to aspects of soft computing such as genetic algorithms, genetic programming and neural networks, and presents the fundamentals of fuzzy expert systems.

Syllabus:

Fuzzy logic. Fuzzy expert systems.
Expert system shell and other tools, including CLIPS. Case based reasoning.
Evolutionary computing - genetic algorithms, genetic programming.
An introduction to Neural Networks.

Learning Outcomes:
An in depth knowledge of the expert system shell CLIPS will be acquired. Students will be able to produce an expert system using this tool. Students will gain a deep understanding of the concepts of evolutionary programming and be able to solve advanced problems using this paradigm. They will have a thorough knowledge of fuzzy logic and the ability to produce rules for a fuzzy controller.

Transferable Skills:
Problem solving, and especially abstract modelling and reasoning. Ability to evaluate and deploy new technologies.

Reading:
J. L. Koloder, Case based reasoning, Morgan Kauffman, 1993.
**Summary:**

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.

**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 to 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:**


WRITING WEB AND WEB SERVICE APPLICATIONS

Synopsis:
Building web and web service applications, typically to implement dynamic web pages.

Syllabus:

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

Code: CS-M68
Credit Points: 10
Taught: Semester 2
Contact Hours: 20 lectures
Incompatible: CS-348
Lecturer: Dr. Harman
Assessment: 40% coursework,
60% written examination
(in May/June)
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 searcer scenarios).

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:
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:
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.
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. Students will be introduced to the Linux system and the language C. Students will prepare an individual project plan.

Syllabus:
Introduction to principles of professional software development.
Professional Issues: ethics, legal issues, copyright, intellectual property rights, data protection act, professional codes of conduct and practice (BCS).

Software Process Models (e.g. waterfall, spiral).
The Capability-Maturity Model.
Software Project Management: requirements specification and quality assurance.
Team work.
The Linux system.
C programming.
Planning an individual project.
Research methodology.

Learning Outcomes:
Students will have 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; enhanced their knowledge through guided learning; prepared an individual project plan, including a significant literature review and be familiar with how to plan and research a project.

Transferable Skills:
Effective use of general IT facilities. Appreciating the need for professional standards in software engineering. Effective information-retrieval skills. Team working. Experience of report writing.

Course Texts:
Abstract data types are an important tool in the design of reliable software. This course gives an introduction to the fundamental principles: abstraction, encapsulation, and modularity. It is shown how to use abstract data types for modelling and reasoning about data and programs. The course discusses techniques for the efficient implementation of abstract data types and gives an introduction to advanced techniques for program synthesis from formal proofs.

Syllabus:

Data Modelling: Using algebras to model data. Basic constructions, isomorphisms. The fundamental principles: abstraction, encapsulation and modularity.


Applications: Modelling, specification and verification of common programming structures and software architectures.

Implementation: Efficient implementation of abstract data types. Persistency, amortised cost, correctness. Examples of programs built on abstract data types will be implemented in an appropriate language.


Learning Outcomes:

Students will have an understanding of the basic theory of abstract data types and will know how it can be applied in modelling and reasoning about data and programs. This will include term rewriting as a programming and verification technique as well as program synthesis from proofs.

Transferable Skills:

Problem solving, abstract modelling, formal reasoning

Reading:


A. Baader and T. Nipkow, Term rewriting and all that, CUP, 1998.

FUNDAMENTALS OF COMPUTER VISION

<table>
<thead>
<tr>
<th>Code:</th>
<th>CS-M77</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>10</td>
</tr>
<tr>
<td>Taught:</td>
<td>Semester 2</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>20 lectures</td>
</tr>
<tr>
<td>Lecturer:</td>
<td>Dr. Xie</td>
</tr>
<tr>
<td>Prerequisite:</td>
<td>CS-217</td>
</tr>
<tr>
<td>Incompatible:</td>
<td>CS-377</td>
</tr>
<tr>
<td>Assessment:</td>
<td>20% coursework, 80% written examination (in May/June)</td>
</tr>
</tbody>
</table>

Synopsis:
This module introduces students to the fundamental topics of computer vision, including image processing, segmentation, feature extraction, motion analysis, object tracking, and recognition.

Syllabus:
This course is divided in four parts: Introduction, Image Processing, Video Analysis and Applications.

Introduction: The first lecture gives an overview of image processing and computer vision (IPCV) and a road show of this course. The second lecture provides a revision of basic and important mathematical techniques frequently used in IPCV.

Image processing: edge detection, Hough transform, scale space representation, object extraction, shape recognition, image filtering, image segmentation, texture analysis, colour analysis.

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

Applications: face detection and recognition, medical application of computer vision, real time computer vision.

Learning Outcomes:
Students will gain a detailed understanding of the fundamental concepts of computer vision, be able to perform the analysis of digital images and videos, and develop the skills necessary to program computer vision systems.

Transferable Skills:
Ability to learn and use computer systems and software package. Ability to evaluate new technologies. General mathematical analysis.

Reading:


HIGH-PERFORMANCE COMPUTING IN C/C++

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 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/pragmas

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:
S. Prata, C++ Primer Plus, 5th Ed.
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/
**DISTRIBUTED PROGRAMMING IN JAVA**

**Synopsis:**
This course provides a solid introduction to the Java programming language and development process. It will consider those aspects of Java which will enable programmers to develop web and distributed applications, including the principles of threading.

**Syllabus:**

**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, and will have a clear understanding of the application of Java in a distributed environment.

**Transferable Skills:**
Effective use of general IT facilities; Problem solving skills; Ability to learn and use computer systems and software packages effectively.

**Course Texts:**


**Reading:**

<table>
<thead>
<tr>
<th>Code:</th>
<th>CS-M81</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Points:</td>
<td>20</td>
</tr>
<tr>
<td>Taught:</td>
<td>Semester 1</td>
</tr>
<tr>
<td>Contact Hours:</td>
<td>20 lectures plus 10 x 2 hour laboratory classes</td>
</tr>
<tr>
<td>Lecturer:</td>
<td>Mr. Whyley</td>
</tr>
<tr>
<td>Assessment:</td>
<td>50% continuous assessment, 50% written examination (in January)</td>
</tr>
</tbody>
</table>

To be awarded credits for this module, students must pass both the examination and coursework components to a tolerable level. Previous experience in a programming language (not necessarily Java) is assumed.
STATE OF THE ART COMPUTER GRAPHICS AND VISUALISATION

Synopsis:
This module is based on seminars, self-studies, and group discussions. The aim of the module is to develop a broad appreciation of the development of technologies in visual computing. Students will have the opportunities to engage visual computing professionals, and discuss their research. They will follow their own study plan within an advanced topic area, and present their own précis of the topic. At the same time, they will acquire new knowledge from group discussions and other students' presentations.

Syllabus:
Each student chooses two topic areas, and will lead the literature collection and discussions on these two topics. The topics are typically chosen from the latest research in computer graphics, visualisation, virtual environments, geometric processing, multimedia, and major applications of visual computing. For the short survey and presentation, each student will normally focus on one topic area. The presentation is typically scheduled in December, and the deadline for the survey will be in January (prior to the examination). Students are required to attend all relevant seminars including departmental seminars, purposely scheduled seminars and other students' presentations. In addition, all students are required to take part in scheduled group discussions, and lead the discussions on their chosen topics. The survey will be made available to the students in the same class, and the presentation is in the form of a public presentation to the class and other appropriate audience. Students are also required to share the information and findings about the background literature. The collection of all these topics will form the broad scope of the subject on which the open-book examination will be set.

Learning Outcomes:
The students will be conversant in many of the most recent developments in computer graphics and visualisation; acquire broad knowledge of various areas of visual computing, and a good understanding of the technical developments in one particular topic area; and develop confidence in appraising and comparing research in different aspects of the subject.

Transferable Skills:
Information retrieval, ability to read critically, to précis and judge information, and ability to manage learning processes. Ability to evaluate and deploy new technologies. Oral presentation, and interactive discussions.

Course Texts:
ACM SIGGRAPH Course Notes and Presentation Slides
Eurographics State-of-the Art Reports
Eurographics Tutorial Notes and Presentation Slides
IEEE Visualisation Tutorial Notes and Presentation Slides
Relevant papers in journals and conferences, such as ACM Transaction on Graphics, IEEE Transactions on Visualisation and Computer Graphics, Computer Graphics Forum,
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. Today, it is a unit whose Director is Beti Williams, the Department's full-time industrial liaison officer. 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. The ITWales Team is responsible for:

Student & Graduate Placements — 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.

Research & Development Partnerships — which are designed for students to complete their final year or MSc projects with direct industrial applications. It provides students with the opportunity to sharpen their technical skills on real world problems. Such projects can also lead to important scientific developments and innovation.

ITWales Business Club — which brings together a large variety of companies, business support agencies and academics and provides a platform for communication, networking, collaboration, and Continuing Professional Development.

ITWales.com — an online IT Magazine for business, which delivers the latest IT news, technology reviews, interviews with leading figures in IT, and features on specific important IT topics.

ITWales Programme of Special Events — which offers innovative, high profile events that are of interest and relevance to not only business, but also schools and the community in general. All events and initiatives within this sector are developed in partnership with key ICT organisations representing business, business support and academia. Further information can be found at www.itwales.com/services.
**Synopsis**

This programme is designed for students to complete their final year or MSc projects with direct industrial applications. It provides students with the opportunity to use their technical skills on real world problems, which often lead to important scientific developments and innovation.

Industrial companies who propose and support such projects benefit through the opportunity of conducting a substantial project which is often curiosity-driven and involves system prototyping. The companies will also benefit from the access to high level technological skills at no extra cost, the introduction to a potential graduate employee, and the opportunity to enhance links with the University.

**Description**

Students studying computer science programmes are required to complete a final year or MSc project that is mainly aimed to provide students 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. Industrially related projects are those projects with direct industrial applications, and they are usually proposed and supported by a company. 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.

An industrially related project requires a fairly tight specification that highlights the aims and objectives, the structure and timing and the required hardware and software resources. 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.
ITWales Business Club

Synopsis:
The ITWales Business Club provides a platform for Welsh business to network on a regular basis through a variety of topical events. The events programme serves businesses throughout Wales, out-reaching from both the programme headquarters in Swansea and the satellite office in Bangor.

Description:
All ITWales Business Club activities are developed to increase understanding and awareness of the strategic and practical applications of technology in business. The activities aim to give competitive edge in the Welsh and global economies.

On average there are four targeted activities every month, focusing primarily on business development, software development and advanced technical issues.

All events provide Welsh business with opportunities to network and collaborate with academia and partner organisations such as the British Computer Society.

<table>
<thead>
<tr>
<th>Period</th>
<th>Throughout the year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commitment</td>
<td>Seminars, workshops, meetings &amp; conferences</td>
</tr>
<tr>
<td>Co-ordinator</td>
<td>Mrs. C. Williams</td>
</tr>
</tbody>
</table>
ITWales Programme of Special Events

**Synopsis:**
The special events programme is a valuable extension to the core activities of ITWales. It comprises a wide range of high profile initiatives developed in partnership with key organisations. Through these events ITWales is able to highlight and promote the continuing development of the Computing Sciences to business, schools and the community.

**Description**

ICT Forum Wales: The special events programme is a valuable extension to the core activities of ITWales. It comprises a wide range of high profile initiatives developed in partnership with key organisations. Through these events ITWales is able to highlight and promote the continuing development of the Computing Sciences to business, schools and the community.

Technocamps: A series of technology-focused workshops for schools and community groups launched in March 2004. The programme is designed to inspire and technically empower young people interested in the computing sciences. Technocamps also supports and enhances school technology club activities providing a network of learning opportunities.

AnimateWales: A series of events for primary and comprehensive schools focusing the animation industry and the technology that drives it. Launched in November 2002 the programme brought together HE institutions, local, national and international animation companies for a full day of animation presentations, aimed at 14/15-year-old pupils, in the Grand Theatre Swansea. This was followed by primary school animation workshops and competitions supported by organisations including the BBC.

Public Lectures: A programme of public lectures presented by guest speakers from academia and industry. The lectures attract an audience from secondary schools, FE and HE institutions and also from business and industry. Two public lectures are currently planned annually and in the past have included speakers such as Sir Terry Matthews (Mitel Networks Corporation), Dr Lyn Evans (CERN), Ed Parsons (Google) and Alan Cox (Linux).
**STUDENT AND GRADUATE PLACEMENTS**

**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 an academic 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 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 £240 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.

**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); Accelero Digital Solutions Ltd. (May 2008).

**Reviews:** Creating a Web Site: The Missing Manual (May 2009); Head First-Rails (April 2009); Head First-PHP and MySQL (February 2009); Making your own portable applications (July 2007).
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

   66 easy-to-read “excursions” into 66 of the main points of interest in the subject.

   An established reference on computing, containing entries explaining terms from across the subject.

   This text provides a very broad overview of the subject, touching on the subject matter of much of the computer science curriculum.

   A compendium of 13 essays, written for programmers, which develop insightful solutions to various programming tasks.

   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

   A study of the causes, and methods for prevention, of system failures, through the analysis of various famous major accidents caused by system failure.

   A compendium of computer mishaps, a study of their causes, and discussion of possible technologies for preventing similar mishaps.

   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

   A readable presentation of the design of algorithms for problem solving, including their efficiency and inherent limitations.
   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**

   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.

   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, etc.

   An entertaining presentation of the logical underpinnings of computers and intelligent behaviour.

   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.

   An introduction to the basic mathematics and logic needed in Computer Science, presented in the style and rigour of “structured” programming.

   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**

   Informative and amusing account of the American PC industry.

   An eye-opening account of instructive and exciting applications of computer science.

   The remarkable 30-year story of the internet.
   A presentation in Scientific American's illustrative style of the capabilities of the world's most powerful computers.

**History**

    A history of the concepts underlying computers, and the scientists who developed them, written in an engaging style.

    Superb picture-book on the origins and development of information processing, compilers and computing.

    Fascinating biography of an original thinker on computing and computers.

    Entertaining history of the important problem of the encryption of information.

    Excellent account of the development of the software industry.
TUTORIALS

Each student whose Home Department is Computer Science is assigned a Personal Tutor. 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. The student will retain their Personal Tutor for their entire stay within the Department. 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 him/her. 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 under the Publications tab on the Academic Registry Website. http://www.swan.ac.uk/registry/Publications/.

You can expect to meet your Personal Tutor at least once a term. In addition to their Personal Tutor students will be supported by seminar and tutorial sessions throughout their degree programme. Attendance at such sessions is mandatory. Attendance at tutorials is mandatory.

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-124 Computers and Society: the report and presentation assessed as part of CS-124 are organised as part of these seminar sessions.

At Level Two, for students taking the CS-254 Software Engineering module, fortnightly academic tutorials will be held to discuss all aspects of the course but will also have the additional role of supporting a 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-124, CS-254, CS-334 or CS-344, 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 in Computing and Software Technology and MSc Computer Science students, tutorials will normally be held fortnightly during both semesters and have the additional role of supporting the modules CS-M74 Software Product Development or CS-M64 Computer Software Systems. 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, biros) 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 stationary provided - all rough work shall be done on the stationary provided and handed in with the completed script. No script, rough work or official stationary 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 ymwnneud 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.

Rhad 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 Academaidd a fydd wedyn yn trefnu i gyfieithu cwestiynau, sgriptiau a traethodau 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 copïau 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 copïau 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 School 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 School within 3 months of initial enrolment, as part of the confirmation of candidature requirements. The School shall forward copies to the Academic Registry for translation arrangements.

LATE SUBMISSION OF WORK
Candidates who fail to submit work by the deadlines shall have the mark awarded reduced by 10% per calendar day. Candidates who submit work more than 1 week (7 calendar days) after the deadline shall receive 0%. However, in such cases the lecturer is expected to mark the work as normal for formative feedback purposes.

Candidates, who are prevented from meeting such deadlines due to extenuating circumstances, may apply to the appropriate School Committee for consideration.

Students should be encouraged to refer to the regulations provided in the Academic Guide: http://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.

'Everyday' Examples of Plagiarism:
Robbie Williams plagiarised lyrics from songs by Guthrie and Loudon in his track 'Jesus in a Camper Van'. The consequences of his actions were expensive with 25% of all royalties being awarded to the original authors.

The University is committed to providing support and guidance to prevent cases of accidental plagiarism and you are advised to seek further advice from the Department 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.

Some Schools/Colleges may allow students to access this software to check whether there are any unattributed passages within an essay/assignment. It is also possible to conduct an oral examination on any piece of work, if the Department suspects a candidate of unfair practice. Again, the Department 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 School 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 School 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 School 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; universal...
second 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.
**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 (see Page 195) 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 School 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 Programs 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.
ENROLLMENT:

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 enroll 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 School 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 School 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 School 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.


**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.

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
### 2010/2011 Session Calendar

<table>
<thead>
<tr>
<th>September</th>
<th>October</th>
<th>November</th>
</tr>
</thead>
<tbody>
<tr>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
</tr>
<tr>
<td>1 2 3 4 5</td>
<td>1 2 3</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>6 7 8 9 10 11 12</td>
<td>4 5 6 7 8 9 10</td>
<td>8 9 10 11 12</td>
</tr>
<tr>
<td>13 14 15 16 17 18 19</td>
<td>11 12 13 14 15 16 17</td>
<td>15 16 17 18 19 20 21</td>
</tr>
<tr>
<td>20 21 22 23 24 25 26</td>
<td>18 19 20 21 22 23 24</td>
<td>22 23 24 25 26</td>
</tr>
<tr>
<td>27 28 29 30</td>
<td>25 26 27 28 29 30 31</td>
<td>27 28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>December</th>
<th>January</th>
<th>February</th>
</tr>
</thead>
<tbody>
<tr>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
</tr>
<tr>
<td>1 2 3 4 5</td>
<td>1 2 3 4 5 6</td>
<td>1 2 3 4 5 6</td>
</tr>
<tr>
<td>6 7 8 9 10 11 12</td>
<td>3 4 5 6 7 8 9</td>
<td>7 8 9 10 11 12 13</td>
</tr>
<tr>
<td>13 14 15 16 17 18 19</td>
<td>10 11 12 13 14 15 16</td>
<td>14 15 16 17 18 19 20</td>
</tr>
<tr>
<td>20 21 22 23 24 25 26</td>
<td>17 18 19 20 21 22 23</td>
<td>21 22 23 24 25 26 27</td>
</tr>
<tr>
<td>27 28 29 30 31</td>
<td>24 25 26 27 28 29 30</td>
<td>28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>March</th>
<th>April</th>
<th>May</th>
</tr>
</thead>
<tbody>
<tr>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
</tr>
<tr>
<td>1 2 3 4 5 6</td>
<td>4 5 6 7 8 9</td>
<td>2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>7 8 9 10 11 12 13</td>
<td>11 12 13 14 15 16 17</td>
<td>9 10 11 12 13 14 15</td>
</tr>
<tr>
<td>14 15 16 17 18 19 20</td>
<td>18 19 20 21 22 23 24</td>
<td>16 17 18 19 20 21 22</td>
</tr>
<tr>
<td>28 29 30 31</td>
<td>30 31</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>June</th>
<th>July</th>
<th>August</th>
</tr>
</thead>
<tbody>
<tr>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
<td>M T W T F S S</td>
</tr>
<tr>
<td>1 2 3 4 5</td>
<td>1 2 3</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>6 7 8 9 10 11 12</td>
<td>4 5 6 7 8 9 10</td>
<td>8 9 10 11 12 13 14</td>
</tr>
<tr>
<td>13 14 15 16 17 18 19</td>
<td>11 12 13 14 15 16 17</td>
<td>15 16 17 18 19 20 21</td>
</tr>
<tr>
<td>20 21 22 23 24 25 26</td>
<td>18 19 20 21 22 23 24</td>
<td>22 23 24 25 26 27 28</td>
</tr>
<tr>
<td>27 28 29 30</td>
<td>25 26 27 28 29 30 31</td>
<td>29 30 31</td>
</tr>
</tbody>
</table>

### Important Academic Dates

**Michaelmas Term:** Monday 27 September 2010 ~ Friday 17 December 2010  
**Lent Term:** Monday 17 January 2011 ~ Friday 1 April 2011  
**Summer Term:** Monday 2 May 2011 ~ Wednesday 29 June 2011  
**Gregynog Colloquium:** Monday 15 November 2010 ~ Wednesday 17 November 2010  
**Project Demonstration Fair:** Thursday May 12 2011  
**Induction/Enrolment:** Monday 27 September 2010 ~ Friday 1 October 2010  
**Semester One:** Monday 4 October 2010 ~ Friday 28 January 2011  
**Mid-Sessional Exams:** Monday 17 January 2011 ~ Friday 28 January 2011  
**Semester Two:** Monday 31 January 2011 ~ Thursday 30 June 2011  
**Sessional Revision:** Monday 16 May 2011 ~ Friday 20 May 2011  
**Sessional Exams:** Monday 23 May 2011 ~ Friday 10 June 2011  
**Supplementary Exams:** Wednesday 17 August 2011 ~ Saturday 27 August 2011  

*All dates are subject to confirmation.*