



Undergraduate Programme Handbook

BSc Computer Systems

2016-2017

Table of Contents

Introduction	1
Degree Specialism	1
Diploma in Industrial Training / Industrial Placements	1
Go Global.....	1
Graduate Attributes.....	2
Programme Structure	2
How To Use This Catalogue.....	3
Timetable	4
Session Dates.....	4
Staff/Student Expectations.....	4
Personal Tutor	5
Key Contacts.....	6
Staff-Student Liaison	7
Enrolment for Courses	7
Course Requirements	8
Attendance	8
Plagiarism & Cheating.....	8
Submission of Coursework.....	8
Late submission of coursework.....	9
Examinations	9
Calculators, Dictionaries & Electronic Devices	9
Unauthorised Material	9
Grades & Assessments.....	10
Feedback	10
Assessment Results and Progress Decisions.....	10
Final Degree Assessment	10
Graduation	11
University Prizes	11
Notification of Mitigating Circumstances	12
Thinking of Leaving	12
Complaints and Appeals	12
Miscellaneous.....	12
Departmental Contacts.....	13
Non Departmental Contacts	14
Programme Structure & Notes – Computer Systems.....	15
Programme Description – Computer Systems	19
Course Descriptors	22
Year 1, Semester 1	22
Software Development 1.....	23
Interactive Systems.....	24
Praxis.....	25
Elective	26

Year 1, Semester 2	27
Software Development 2	28
Software Development 3	29
Introduction to Computer Systems	30
Web Design and Databases	31
Year 2, Semester 1	32
Interaction Design	33
Web Programming	34
Data Structures & Algorithms	35
Programming Languages	36
Year 2, Semester 2	37
Software Design	38
Database Management Systems	39
Hardware-Software Interface	40
Discrete Mathematics	41
Creative Design Project	42
Operations Management	43
Year 3, Semester 1	44
Software Engineering	45
Data Communications and Networking	46
Knowledge Management	47
Artificial Intelligence & Intelligent Agents	48
Year 3, Semester 2	49
Professional Development	50
Operating Systems & Concurrency	51
Language Processors	52
Sociotechnical & Soft Systems	53
Year 4, Semester 1	54
Research Methods & Requirements Engineering	55
Computing in the Classroom	56
Computer Network Security	58
Information Systems Methodologies	59
Mobile Communications and Programming	60
Rigorous Methods for Software Engineering	61
Industrial Programming	62
3D Graphics & Animation	63
Year 4, Semester 2	64
Project: Design & Implementation	65
Project: Testing & Presentation	65
Advanced Interaction Design	66
Big Data Management	67
Digital Knowledge Economy	68
E-Commerce Technology	70
Computer Games Programming	72
Network Applications	73

Introduction

This programme specific handbook should be read in conjunction with the School Undergraduate Handbook for the School of Mathematical and Computer Sciences (MACS), which can be found on the School website <http://www.macs.hw.ac.uk/students/>

Further information for current undergraduate students can be found at:

<http://www.macs.hw.ac.uk/students/cs/>

<http://www.hw.ac.uk/registry/>

<http://www.hw.ac.uk/students.htm>

The University operates the Heriot Watt Assessment and Progression System (HAPS) which specifies minimum progression requirements. Schools have the option to apply progression requirements above the minimum University requirement, which are approved by the Studies Committees. Students should refer to the programme specific information on progression requirements. This information is detailed later in this handbook. You can also contact your Personal Tutor for information.

Degree Specialism

This handbook contains information on the programme structure, notes, description and the courses offered on the BSc Computer Systems degree. The degree can also be offered as BSc Computer Systems (*Computer Games Programming*) by specialising in courses in this area in the final year.

To graduate with the degree BSc Computer Systems (*Computer Games Programming*) candidates must take the course F20GP Computer Games Programming and do at least one of F20SC Industrial Programming or F20AD Advanced Interaction Design and do an approved dissertation topic suitable for this specialism. You must inform the 4th Year Supervisor, at the start of the honours/4th year if you want to exit with this degree specialism

Diploma in Industrial Training / Industrial Placements

The Department encourages students to undertake up to a year-long paid study-relevant work-placement during their studies. This can be done through the Diploma in Industrial Training (for eligible students only) or by temporarily suspending your studies. More information about the Diploma in Industrial Training can be found at: <http://www.macs.hw.ac.uk/students/cs/ug-programmes/>

In all cases the student is responsible for securing a work placement. The University's Careers Advisory Service and the DIT Programme Director, Dr Lilia Georgieva, can advise anyone interested on how to go about researching and applying for a placement. You are strongly advised to contact the Careers Advisory Service for help on writing CVs, online tests and assessment centres.

Go Global

At Heriot-Watt, you have the opportunity to become a global student by taking part in an Inter-Campus Transfer, Erasmus+ or Exchange. Studying abroad is a unique opportunity and likely to provide some of your most memorable life experiences. Further information can be found at:

<https://www.hw.ac.uk/students/studies/go-global.htm>

Graduate Attributes

As a student of Heriot-Watt University, you are part of our global community. You will meet new people, discover new interests, develop your life skills and enhance your employability and career prospects.

The University will provide you with opportunities to develop skills, qualities and academic abilities during your time as a Heriot-Watt student. These are known as the **Four Heriot-Watt Graduate Attributes**:

- *Specialist*
- *Creative*
- *Global*
- *Professional*

Further information can be found at:

<https://www.hw.ac.uk/students/doc/StudentGraduateAttributes.pdf>

Programme Structure

Our academic year is divided into 2 semesters corresponding to 30 weeks. There will be 12 weeks teaching in each semester. You are expected to study 4 courses each semester, giving a total of 8 courses in a full year. Each course is worth 15 credits. Courses may be mandatory or optional.

Mandatory courses: These courses are compulsory

Optional courses: Students are required to choose from a specified list of courses relevant to the subject area of their degree discipline.

All undergraduate courses are designed to be of equal length in terms of student effort. The average student is expected to put in a total effort of 150 hours per course. These 150 hours include all lectures, tutorials, computing labs, workshops, background reading, writing up notes, coursework, revision and examinations for the course.

How To Use This Catalogue

The course information, which appears in the format below, is designed to provide you with sufficient details about courses, their content and assessment methods and will help you choose your optional courses.

Course Code:	Course Title:	Course Co-ordinator:
Pre-requisites:		
Aims:		
Syllabus:		
Learning Outcomes: Subject Mastery	<i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i>	
Learning Outcomes: Personal Abilities	<i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i>	
Assessment Methods:	Assessment:	Re-assessment:

Many of the courses have on-line material available at the University's Virtual Learning Environment (VISION) which can be found at: <http://vision.hw.ac.uk/>

Terminology

Course Code	The first character identifies the School (F = MACS) The second digit identifies the discipline area (2=Computer Science). The next digit is the SCQF level of the course: SCQF Level 7 normally studied in Year 1 SCQF Level 8 normally studied in Year 2 SCQF Level 9 normally studied in Year 3 SCQF Level 10 normally studied in Year 4 (A zero in course codes) SCQF Level 11 normally studied in Year 5/Postgraduate (A one in course codes) The next 2 letters identify the topic. Courses coded C are delivered by the School of Management & Languages
Course Co-ordinator:	The name of the member of staff who is responsible for delivery of the course
Pre-requisites:	Students <i>must have gained Grade D</i> or above in the courses listed here in order to gain entry to the course.
Aims:	A brief statement of what the course aims to do
Syllabus:	A brief summary of what is included in the course
Learning Outcomes: Subject Mastery	These will include Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)
Learning Outcomes: Personal Abilities	Industrial, Commercial & Professional Practice; Autonomy, Accountability Working with Others; Communication, Numeracy & ICT

Assessment Methods: Details of the weighting and type of assessment(s) and re-assessment (if any) for the course

Timetable

A timetable of classes will be available online at:

<https://www.hw.ac.uk/students/studies/timetables.htm>

Any timetable problems should be notified to Jill Gunn (Room EM1.20)

Session Dates

Dates

5 - 9 September 2016
12 September – 2 December 2016
5 – 16 December 2016
19 December 2016 – 6 January 2017
9 January – 31 March 2017
3 – 21 April 2017 (Easter: 16 April)
24 April – 19 May 2017
22 May – 11 August 2017
16 June 2017 (Scottish Borders Campus)
20 – 23 June 2017 (Edinburgh Campus)
3 – 11 August 2017
16 – 17 November 2017

Activity

Welcome Week
Semester 1 teaching
Semester 1 exams
Semester 1 break
Semester 2 teaching
Semester 2 break
Semester 2 exams
Dissertation work / Revision
Graduations
Graduations
Resits
Graduations

Staff/Student Expectations

What you can expect from staff

Teaching is one of the most important duties for staff. Although we have research and admin duties which need our attention too, we promise students:

- ◆ Commitment to helping you learn
- ◆ Politeness and respect
- ◆ A regular office hour slot for face to face meetings
- ◆ Written feedback and a mark for coursework within 3 teaching weeks after the hand-in time.
- ◆ A reply to general email questions within 5 working days
- ◆ A response from your Personal Tutor within 2 working days
- ◆ If you would like to see your exam script to see where you went wrong, go along to the school office and ask for a form to request this. (It'll take a bit of time to get the paper out of archives). You can also make an appointment with the lecturer to get further advice on how to improve your work.
- ◆ Sometimes staff members are away on university business (for example at a research project meeting outside the UK), and won't be able to respond as quickly as normal. If this happens, they will tell you about it (e.g. on an "out of office" message) and will advise you who to contact instead.

What staff can expect from students

Most importantly, we expect you to take charge of your own learning. This is your degree! To get the most of your time at university you need to be independent and proactive. We understand that you have other demands on your time, such as paid employment, but as full time students, your studies should come first.

- ◆ Commitment to your learning
- ◆ Politeness and respect
- ◆ Attendance at classes, unless they are specifically identified as voluntary. During semester it is your responsibility to be available on campus to attend classes and in particular class tests.
- ◆ Attention, courtesy and participation during classes
- ◆ Preparation and practice for classes as specified by the lecturers, such as reading or coding. For every hour of timetabled class, we expect you to spend 2-3 hours in private study.
- ◆ Practice, practice, practice! In order to become a good programmer, you need to program regularly. If you are having trouble we will help, but the most useful thing you can do for yourself is devote time to programming.
- ◆ Basic organisation skills, such as coming to classes with pen and paper ready to take notes, and using a calendar so you don't forget deadlines and appointments
- ◆ If you can't make a scheduled meeting with a staff member, please notify them in advance rather than just not turning up
- ◆ Check your email and logging into Vision at least every other day
- ◆ A reply to email from staff within 5 working days (if it requires a reply!)
- ◆ We expect you to pay attention to the feedback we give you, and to attempt to improve your work based on that feedback.
- ◆ We encourage you to keep yourself informed about new and interesting developments in computer science above and beyond what you learn in the taught courses. The department is full of experts in a wide range of areas who would love to chat to keen students about their research. Seek them out!
- ◆ If you have a problem which is interfering with your studying, please discuss it with your Personal Tutor. We are here to help.

Personal Tutor

You will be allocated a Personal Tutor when you arrive at the University and, normally, you will retain the same Personal Tutor as long as you are registered in the Department of Computer Science. The Personal Tutor is your main academic link with the University, and is there to provide you with help and advice about your studies. Under certain circumstances, with the permission of the Head of Computer Science, it may be possible to change your Personal Tutor. A list of Personal Tutors can be found on the internal departmental web pages.

Every year a few students run into personal difficulties (e.g. family illness, accommodation, financial, etc.). As well as being generally supportive, Personal Tutors can help in a number of practical ways. For example, if illness prevents you from completing project work or sitting examinations, your Personal Tutor can sometimes help with re-scheduling or making alternative arrangements for assessment. However, you must notify your Personal Tutor as soon as possible, or there is very little that can be done. This is particularly important if illness affects your Examinations. Also, it is essential to provide a medical certificate (see **Notification of Special Circumstances**, p 10). With other problems, your Personal Tutor can put you in touch with the appropriate University support service (Chaplaincy, Medical Centre, Student Welfare Services or

Student Association). **Personal Tutors are there to help; do not hesitate to contact yours if you need them.**

Our Personal Tutoring team have been selected because they specialise in different areas of expertise. If you have a problem you can go to see your allocated Personal Tutor, or contact one of the others depending on the nature of your difficulty.

Personal Tutor Contact Details

To direct dial a member of staff: (0131) 451 plus extension number

Dr Tessa Berg	T.Berg@hw.ac.uk	Ext 8223
Ms Jenny Coady	J.Coady@hw.ac.uk	Ext 4178
Dr Frank Broz	F.Broz@hw.ac.uk	Ext 3430
Dr Peter King	P.J.B.King@hw.ac.uk	Ext 3433
Dr Stefano Padilla	S.Padilla@hw.ac.uk	Ext 3424
Dr Manuel Maarek	M.Maarek@hw.ac.uk	Ext 3287
Dr Fiona McNeill	F.McNeill@hw.ac.uk	Ext 3435
Prof Ruth Aylett	R.S.Aylett@hw.ac.uk	Ext 4189
Dr Diana Bentall	D.S.Bental@hw.ac.uk	Ext 4196

Further information on the role of personal tutors can be found at:

<https://www.hw.ac.uk/students/studies/personal-tutors.htm>

Key Contacts

To direct dial a member of staff: (0131) 451 plus extension number

 Hans Wolfgang Loidl	Computer Systems degree specific Issues	H.W.Loidl@hw.ac.uk	Ext 3421
 Dr Michael Lones	First Year Issues	M.Lones@hw.ac.uk	Ext 8434

Prof Nicholas Taylor 	Second Year Issues	N.k.taylor@hw.ac.uk	Ext 3436
Helen Hastie 	Third Year Issues	H.Hastie@hw.ac.uk	Ext 3344
Peter King 	Fourth and fifth Year Issues	P.J.B.King@hw.ac.uk	Ext 3433
Greg Michaelson 	Help for students in crisis	G.Michaelson@hw.ac.uk	Ext 3422

Staff-Student Liaison

Students are asked to elect a class representative at the start of every academic year. Your representative will keep the staff up to date with any problems which students in the year have identified, and they also keep students informed of actions taken by staff to address these problems. The School Officer is a student appointed by the Heriot Watt Student Union to work closely with the class representatives and staff to make sure that the students' needs are met. If you have a request or suggestion about a course, the first thing to do is to talk to the lecturer in question, or ask the class representative to do this for you. If that doesn't work, you can talk to the year supervisor or the Programme Director for your degree.

Enrolment for Courses

You must be enrolled for the courses which you are studying. This will be done initially during on-line enrolment.

Any subsequent changes to optional choices must be agreed between you and your Programme Director or Personal Tutor, and then recorded on a Change of Course Form available from MACS School Office (EM1.25). The form must then be returned to Room EM1.25 for processing.

All course changes must be made by the end of week 3 of each semester. Any changes submitted after this will incur a charge of £10 per course. No changes can take place after week 5 of each semester.

Course Requirements

Attendance

In order to achieve course and programme learning outcomes, students are expected to attend all scheduled course learning sessions (e.g. timetabled lectures, tutorials, lab sessions, etc). Should you have to miss a timetabled session due to ill health or other legitimate reasons, you should submit a self-certification or medical certification or an application for consideration of Mitigating Circumstances <https://www.hw.ac.uk/students/studies/examinations/mitigating-circumstances.htm>.

Students who fail to satisfy course attendance requirements may, after due warning, be disallowed from presenting themselves for examination in the course (see <http://www.hw.ac.uk/students/doc/withdrawalprocedures.pdf>).

Plagiarism & Cheating

Cheating in examination and plagiarism, which is, the presentation of another person's ideas or work as one's own, are very serious offences and are dealt with severely. They carry a range of penalties up to and including expulsion from the University.

Students are responsible for familiarizing themselves with University policy on these matters. For more detail, see the School Undergraduate Handbook (http://www.macs.hw.ac.uk/students/wp-content/uploads/UG_Handbook.pdf) and the sections on Plagiarism and Regulation 9 on the Registry's website.

Submission of Coursework

All courses will include some coursework which must be done during the semester. Coursework Submission front sheets are available in the first floor corridor between the Earl Mountbatten Building and Colin Maclaurin (Near the MACS School Office). The coursework submission front sheets are printed on **lilac** coloured paper. The **CS/IS coursework box** can be found at the same location.

Please ensure that you:

1.	state which degree programme you are studying and year of study
2.	Complete your personal details on the form, i.e., your name, matriculation number.
3.	Write the course code and course title on the front sheet.
4.	Sign and date the front sheet to confirm that it is your "sole and original work"
5.	Staple the front sheet to your coursework before you put it in the CS/IT coursework box.

All coursework must be submitted by 3.30pm on the deadline date unless otherwise specified by the lecturer. A list of coursework deadlines will normally appear on the undergraduate student website from week 3 each semester, which will also detail the amount of effort that is expected for each piece of coursework. Penalties may be imposed for late submission of coursework.

Late submission of coursework

Coursework that is submitted late will normally be subject to a penalty. The standard penalty system is that 10% of the maximum available mark is deducted from the mark awarded for each day late. Days are counted as working days for the School Office. Any coursework submitted more than five days late will be awarded a mark of zero.

A Course Leader may decide to adopt a different penalty system for a particular course. In this case the penalty system to be applied will be set out in the course documentation or in the coursework specification.

Extensions to submission dates

Students who have serious concerns about meeting submission dates for coursework should consult the Course Leader as soon as possible. Any extension to the submission deadline must be approved by the Course Leader, and the reason for the extension will be recorded. Applications for extensions made **after** the due submission date will not normally be approved.

Examinations

It is the student's responsibility to check all relevant examination timetables (including resits) on the Registry web page <https://www.hw.ac.uk/students/studies/examinations.htm>

Should you be required to be re-assessed in any examinations, you *must* make yourself available to take them. All exams must be taken at the Edinburgh Campus.

Past exam papers for F2 courses can be found at:

<http://www.macs.hw.ac.uk/students/cs/cs-first-year-past-exam-papers/>

THESE ARE ONLY ACCESSIBLE ON-CAMPUS OR IF YOU USE THE VPN

(<https://hwvpn.hw.ac.uk/workplace/access/home>)

Calculators, Dictionaries & Electronic Devices

Where a calculator is required for the completion of an examination, a student may use any basic scientific calculator, except the following: graphics calculator, programmable calculator and a calculator which features text storage or retrieval facilities.

No translation dictionaries are permitted in any of the University's examinations. The only exception to the policy is in the case of individual students who had been assessed by the University's Disability Service as requiring access to a translation dictionary.

Students are not allowed to have mobile phones or other communication devices on or about their person during examinations. Phones may be left at the front of the examination room but must be switched off.

Unauthorised Material

You must not have any unauthorised pre-printed materials or electronic devices including mobile phones in the examination room. Cheating in an examination is treated very seriously by the University. If you do have any material relevant to the exam which you have brought in by mistake, you must hand it over to an invigilator before the start of the examination. Invigilators will carry out checks on authorised materials and calculators.

Grades & Assessments

Grades for each course are awarded as follows:

Grade A	Excellent	Overall mark of approximately 70% or more
Grade B	Very Good	Overall mark of approximately 60% to 69%
Grade C	Good	Overall mark of approximately 50% to 59%
Grade D	Satisfactory	Overall mark of approximately 40% to 49%
Grade E	Adequate	Minimum required for the award of credits but at least a grade D is needed for progression to subsequent courses
Grade F	Inadequate	Fail

Feedback

Feedback is a two-way process. Feedback is provided to students in a variety of ways in order to help you to reflect on and to evaluate your progress and to assist you to take steps to improve before the next relevant assessment. For most courses, students can expect feedback on assessed coursework within three teaching weeks of the coursework due date.

Feedback is sought from students via Student-Staff Liaison Committees and various surveys so that the School can continue to enhance the student learning experience. Your feedback is valued by the School, so please be sure to provide feedback whenever it is sought.

Assessment Results and Progress Decisions

You will get your assessment results online following the relevant Assessment Boards which take place in January, May and August (resit diet).

The Progression Board meets at the end of the academic year to decide which students will be allowed to proceed to the next year of their degree programme. You will receive an email from the University containing a link to a summary of your results for the year and the Board's progression decision, and whether you must resit any courses.

In years 1, 2 and 3 if you do not pass a course at the first attempt, you have one opportunity to resit the course during the resit diet in early August. In Year 3, re-assessment is for credit only and you cannot improve your overall average (which accounts for 20% of your final degree results) unless you are re-sitting for medical reasons. There are no re-sit opportunities for courses in Year 4.

If you receive a pass/proceed decision that allows you to progress at the Summer Progression Board you can enrol online from mid-August. If you have resits, and are able to progress following the Resit Progression Board you may enrol online once you have received your Assessment Results email confirming this.

Final Degree Assessment

The Award Board meets in the last week of May to consider the assessment marks and make recommendations on degree classifications.

For the BSc Computer Systems honours degree, the Examiners take into account 3rd and 4th year course marks in deciding the class of Honours. The final mark is the average of those marks, weighted as: 20% from 3rd year average, 50% from the 5 taught courses in 4th year and 30% from the individual dissertation in 4th year. In broad terms, an average mark of over 70% for first class

honours, 60% - 70% for upper second class honours, 50% - 60% for lower second class honours, and 40% - 50% for third class honours, would be required, subject to the agreement of the Examiners. (Note that 480 credits are required for the award of an honours degree.).

Graduation

When you have completed your degree your award is conferred at a graduation ceremony. Details on graduation, including how to apply, deadlines for applying and the cost, can be found at: <https://www.hw.ac.uk/students/studies/graduation.htm>. This website also includes details of gown hire and guest tickets.

University Prizes

Final Year Awards

Watt Club Medal

Awarded for exceptional merit and distinction in the **final year** of any degree course in the Department of Computer Science. No more than one medal can be awarded in each discipline within a School in any year.

Systems Consultants Ltd Prize (£200)

The best student in the final year of the course for the degree of BSc in Computer Science.

Cooper-Walker Engineering Ltd Prize (£200)

For outstanding project work in a degree course in the Department of Computer Science.

Andrew Stewart Prize 1 (£200)

For the most deserving student in the **fourth year** of a degree course in the Department of Computer Science.

Continuing Years Awards

University Prizes, Years 1, 2 & 3 (£100)

For outstanding merit (In practice an average mark of at least 70% is regarded as the minimum standard). Available to students on any undergraduate course in the Department of Computer Science. There is one prize per academic year per year of study.

Andrew Stewart Prize 2 (£200)

For the most deserving student in the **second year** of a degree course in the Department of Computer Science.

ICL Prize (£200)

The best student in the **first year** of the course for the degree of BSc in Computer Science.

Lockheed Martin/Santander Prize

For the best Group Project in the third year of a degree programme. Each member of the winning group will be awarded £75.00, and each member of the 2 runner up groups will be awarded £50. Students who have contributed so little that their individual grade is below E are not eligible for this prize.

Notification of Mitigating Circumstances

If you have been affected by mitigating circumstances which have affected your ability to complete your assessments you must notify us as soon as possible.

You should read the University's Policy on Mitigating Circumstances in Relation to Assessment at: and then complete the application form at: <http://www.hw.ac.uk/students/studies/examinations/mitigating-circumstances.htm>.

This form along with any relevant evidence should be submitted to the School Office.

It is **very important** that you also notify your Personal Tutor **as soon as possible** of any mitigating circumstances, such as illness or bereavement, which could adversely affect your assessment performance. In the case of illness, a medical certificate must be supplied to the School Office (EM1.25). The Examiners will always take such circumstances into account where appropriate, but the later the notification, the less scope there is to do so.

In particular, notification should be before the results are announced. Late notification will mean that either no account can be taken, or that formal procedures have to be invoked. In the latter case, final year students will not be permitted to graduate until these procedures have been completed. For further details, see the University Regulations and the School Undergraduate Handbook.

Thinking of Leaving

Many students think about leaving university at some stage during their studies. If anything is bothering you or you are thinking about leaving, please speak to a member of staff to explore and understand what you can do.

There may be other options such as additional help (perhaps advice on how to improve your academic skills), a temporary suspension of studies or transferring to another programme.

Further information is available at: <https://www.hw.ac.uk/students/studies/leaving.htm>

Complaints and Appeals

Our aim at Heriot-Watt is to ensure that your experience while studying with us is of the highest quality. However, we recognise that during your time at the University there may be circumstances that occur where you feel you need to make a complaint or to appeal a decision.

Further information is available at: <https://www.hw.ac.uk/students/studies/complaints-appeals.htm>

Miscellaneous

Lockers and buildings access cards

Lockers for use by students are available at a number of sites in the Earl Mountbatten Building. They are allocated for the duration of each academic year on a first-come first-served basis. Keys for lockers in the EM Building are available at School's Office EM1.25 for a deposit of £10. In addition, final year students may request extended access to enter the Earl Mountbatten Building until 10pm on a weekday and also during the weekends. If you wish to access the building in those hours, please see Alistair Houston in room EM 1.31

Mail

Mail (internal and external) to students is delivered to pigeon holes on the first floor of the Earl Mountbatten Building, inside the School Office (EM1.25). Check yours regularly.

Noticeboard

Various notices are posted on the noticeboard in the corridor along from the School Office.

Departmental Contacts

To direct dial a member of staff: (0131) 451 plus extension number

Head of School	Beatrice Pelloni	B.Pelloni@hw.ac.uk	Ext 8306
Head of Computer Science	Andrew Ireland	A.Ireland@hw.ac.uk	Ext 3409
Director, Undergraduate Study and Computer Science/Software Engineering degree	Hans-Wolfgang Loidl	H.W.Loidl@hw.ac.uk	Ext 3421
Director, Information Systems Degree	Jenny Coady	J.Coady@hw.ac.uk	Ext 4178
Director of Studies, Year 1	Michael Lones	M.Lones@hw.ac.uk	Ext 8434
Director of Studies, Year 2	Nick Taylor	N.K.Taylor@hw.ac.uk	Ext 3436
Director of Studies, Year 3	Helen Hastie	H.Hastie@hw.ac.uk	Ext 3344
Director of Studies, Year 4	Peter King	P.J.B.King@hw.ac.uk	Ext 3433
Special Needs Advisor	Tessa Berg	T.Berg@hw.ac.uk	Ext 8223
Administrator until Jan 2017	Oliwia (Olivia) Kupinska	O.Kupinska@hw.ac.uk	Ext 3432
Administrator from Jan 2017	Lisa Kinnaird	L.M.Kinnaird@hw.ac.uk	Ext 3432

Lecturers	E-Mail	Room	Extension
Prof Ruth Aylett	R.Aylett@hw.ac.uk	EM1.59	4189
Prof Lynne Baillie	L.Baillie@hw.ac.uk	EMG.30	4160
Dr Phil Barker	Phil.Barker@hw.ac.uk	EMG.41	3278
Dr Diana Bental	D.S.Bental@hw.ac.uk	EM1.67	8338
Dr Tessa Berg	T.Berg@hw.ac.uk	EMG.35	8223
Dr Frank Broz	F.Broz@hw.ac.uk	EM1.46	3430
Prof Albert Burger	A.G.Burger@hw.ac.uk	EMG.36	3428
Prof Mike Chantler	M.J.Chantler@hw.ac.uk	EM1.48	3352
Dr Jessica Chen-Burger	Y.J.Chenburger@hw.ac.uk	EMG.38	3434
Dr Santiago Chumbe	S.Chumbe@hw.ac.uk	EMG.41	3762
Ms Jenny Coady	J.Coady@hw.ac.uk	EMG.37	4178
Prof David Corne	D.W.Corne@hw.ac.uk	EMG.39	3410
Dr Chris Fensch	C.Fensch@hw.ac.uk	EMG.29	3416
Dr Jamie Gabbay	M.Gabbay@hw.ac.uk	EMG.50	3425
Dr Lilia Georgieva	L.Georgieva@hw.ac.uk	EMG.54	8159
Dr Alasdair Gray	A.J.Gray@hw.ac.uk	EM1.39	3429
Dr Gudmund Grov	G.Grov@hw.ac.uk	EMG.49	3412
Dr Helen Hastie	H.Hastie@hw.ac.uk	EM 1.42	3344
Prof Andrew Ireland	A.Ireland@hw.ac.uk	EMG.57	3409
Dr Mike Just	M.Just@hw.ac.uk	EM1.37	3336
Prof Fairouz Kamareddine	F.D.Kamareddine@hw.ac.uk	EM1.65	3868
Dr Peter King	P.J.B.King@hw.ac.uk	EMG.51	3433
Dr Ekaterina (Katya) Komendantskaya	E.Komendantskaya@hw.ac.uk	EMG26	8283

Prof Oliver Lemon	O.Lemon@hw.ac.uk	EM1.40	3782
Dr Katrin Lohan	L.Lohan@hw.ac.uk	EM1.44	8338
Dr Hans-Wolfgang Loidl	H.W.Loidl@hw.ac.uk	EMG.48	3421
Dr Michael Lones	M.Lones@hw.ac.uk	EMG.31	8434
Dr Manuel Maarek	M.Maarek@hw.ac.uk	EM1.63	3287
Dr Fiona McNeill	F.McNeill@hw.ac.uk	EMG.52	3435
Prof Greg Michaelson	G.Michaleson@hw.ac.uk	EMG.56	3422
Dr Stefano Padilla	S.Padilla@hw.ac.uk	EM1.38	4166
Dr Ron Petrick	R.Petrick@hw.ac.uk	EM1.05	3943
Dr Verena Rieser	V.T.Rieser@hw.ac.uk	EM 1.36	4129
Prof Sven Bodo Scholz	S.Scholz@hw.ac.uk	EM G.27	3814
Dr Hamish Taylor	H.Taylor@hw.ac.uk	EM1.43	3427
Prof Nick Taylor	N.K.Taylor@hw.ac.uk	EM1.62	3436
Dr Patricia Vargas	P.A.Vargas@hw.ac.uk	EMG.28	4161
Dr Joe Wells	Joe.Wells@hw.ac.uk	EM1.61	3869

In the first instance all undergraduate enquiries should be directed to the School Office room EM 1.25, email to macs-schooloffice@hw.ac.uk

Non Departmental Contacts

Lecturer	Email	Room	Extension	Course
Department of Mathematics (F1 courses)				
Prof Mark Lawson	M.V.Lawson@hw.ac.uk	CMS.21	3210	F17LP
Dr Anatoly Konechny	A.Konechny@hw.ac.uk	CMT.09	3077	F17SC

Programme Structure & Notes – Computer Systems

7. OPTIONAL COURSES

Edinburgh/ Orkney/SBC	HWUM	Dubai	ALP	IDL	Collaborative Partner	Stage	Semester	Phase (Part- time only)	Courses: (Please highlight any new courses and include the course descriptors)	Credit Value		Notes		
										SCQF	MQA			
									Code Course Title					
STAGE 1														
✓		✓				1	1		F17LP	Logic & Proof	15		7	Students in Edinburgh must choose either option or elective in semester 1
STAGE 2														
✓		✓				2	2		F28HS	Hardware-Software Interface	15		8	Students must choose 2 options in semester 2
✓		✓				2	2		F17SC	Discrete Mathematics	15		7	
✓						2	2		F28CD	Creative Design Project	15		8	
✓						2	2		C18OP	Operations Management	15		8	
STAGE 4														
✓		✓				4	1		F20CL	Computing in the Classroom	15		10	Student choose 3 courses in semester 1 and 2 courses in semester 2
✓		✓				4	1		F20CN	Computer Network Security	15		10	
✓		✓				4	1		F20IF	Information Systems Methodologies	15		10	
✓		✓				4	1		F20MC	Mobile Communications & Programming	15		10	
✓		✓				4	1		F20RS	Rigorous Methods for Software Eng.	15		10	
✓		✓				4	1		F20SC	Industrial Programming	15		10	
✓		✓				4	1		F20GA	3D Graphics & Animation	15		10	
✓		✓				4	2		F20AD	Advanced Interaction Design	15		10	
✓		✓				4	2		F20BD	Big Data Management	15		10	
✓		✓				4	2		F20DE	Digital and Knowledge Economy	15		10	
✓						4	2		F20EC	E-Commerce Technology	15		10	
✓		✓				4	2		F20GP	Computer Games Programming	15		10	
✓		✓				4	2		F20NA	Network Applications	15		10	

8. ELECTIVES (please provide a detailed description and course lists where possible)

Stage 1:	Any SCQF Level 7 course, which must be approved by the Board of Studies.
Stage 2:	
Stage 3:	
Stage 4:	
Stage 5:	

PROGRAMME NOTES

9. COMPOSITION & STAGE NOTES e.g. xx taught Courses (xx mandatory & xx optional)

Stage 1:	8 taught courses: 7 mandatory 1 optional/elective in Edinburgh 1 optional in Dubai
Stage 2:	8 taught courses: 6 mandatory and 2 optional Direct entrants to Stage 2 and internal transfers from other degrees will be expected have an appropriate background in programming and database technology
Stage 3:	8 taught courses, 4 mandatory and 4 optional Direct entrants to Stage 3 will be expected have appropriate programming experience and background knowledge.

	Candidates shall pursue a group project throughout the year, which shall be synoptically assessed in conjunction with material from the associated courses (F29SO and F29PD).
Stage 4:	Edinburgh and Dubai Programmes have 3 mandatory and 5 optional In any one year not all optional courses may be offered. Guidance in course choice will be given by academic Personal Tutors. Candidates are required to undertake an individual dissertation project which shall run throughout the year.

10. NOMINAL PASS MARK/GRADE			11. SUMMARY OF ASSESSMENT METHODS (Expressed as a percentage)			
<i>(Highlight any changes)</i>	Mark	Grade	Coursework:	Varies in courses from 100% to 20%	Examination:	Varies in courses from 0% to 80%
Integrated Masters			Variations in assessment methods across campuses/modes of study are as follows:			
Honours	40%	D	Variations in assessment methods across campuses/modes of study are as follows:			
Ordinary	40%	D				
Diploma	40%	D				
Certificate	40%	D				

12. PROGRESSION REQUIREMENTS	
Part A. Minimum number of credits required to progress through each stage are as follows	
Stage 1 to 2:	120 credits (8 courses)
Stage 2 to 3:	240 credits (16 courses)
Stage 3 to 4:	360 credits (24 Courses) and an overall exam average of 50% at the first attempt.
Stage 4 to 5:	
Part B. Minimum grade D required in the following courses: (progression requirements exceeding a grade D must be qualified)	
Stage 1:	Software Development (F27SA), Interactive Systems (F27IS), Web Design & Databases (F27WD), Introduction to Computer Systems (F27CS), Software Development 2 (F27SB) and Software Development 3 (F27SG)
Stage 2:	Interaction Design (F28IN) Web Programming (F28WP), Data Structures & Algorithms (F28DA), Database Management Systems (F28DM), Software Design (F28SD) and Programming Languages (F28PL)
Stage 3:	6 courses including Software Engineering (F29SO) & Professional Development (F29PD). Re-assessment in Stage 3 is available for credit only and not to improve overall average
Stage 4:	

13. RE-ASSESSMENT OPPORTUNITIES	
The re-assessment policy for this programme is in line with University Regulations as set out below	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>
<i>(please tick)</i>	
<i>If you have selected "No" please amend the statement below and highlight changes.</i>	
<ol style="list-style-type: none"> 1. A student who has been awarded a Grade E or a Grade F in a course may be re-assessed in that course. 2. A student shall be permitted only one re-assessment opportunity to be taken at the Resit diet of examinations following the first assessment of the course. 3. A student shall not be re-assessed in any qualifying course taken in the final stage of a course of study. 4. The Progression Board may permit a student to be re-assessed in any qualifying course not taken in the final stage in order to gain credits for the course, provided that the mark or grade obtained in the first assessment of any such course is used in determining the classification of the degree to be awarded. 	

14. AWARDS, CREDITS & LEVEL	
The awards, credits and level for this programme is in line with University Regulations as set out below	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>
<i>(please tick)</i>	
<i>If you have selected "No" please amend the statement below and highlight changes.</i>	
Part A. Credit Requirements	
Integrated Masters	N/A
Honours Degree (inc MA)	480 SCQF credits including a minimum of 180 credits at Level 9 and 10 of which at least 90 credits at Level 10

Ordinary or General Degree	360 SCQF credits including a minimum of 60 credits at Level 9
Diploma of Higher Education	240 SCQF credits including a minimum of 90 credits at Level 8
Certificate of Higher Education	120 SCQF credits including a minimum of 90 credits at Level 7
Part B. Mark/Grade Requirements	
Integrated Masters	N/A
Honours Degree (inc MA)	<p>1st: Weighted Average >=70% over all qualifying courses at grades A-D</p> <p>2.1: Weighted Average >=60% over all qualifying courses at grades A-D</p> <p>2.2: Weighted Average >=50% over all qualifying courses at grades A-D</p> <p>3rd: Weighted Average >=40% over all qualifying courses at grades A-D</p> <p>These are default marks/grades. The Board of Examiners may exercise some discretion in accordance to University Regulations</p>
Ordinary or General Degree	<p>Minimum of grade D in all pre-requisite courses</p> <p>These are default marks/grades. The Board of Examiners may exercise some discretion in accordance to University Regulations</p>
Diploma of Higher Education	<p>Minimum of grade D in all pre-requisite courses</p> <p>These are default marks/grades. The Board of Examiners may exercise some discretion in accordance to University Regulations</p>
Certificate of Higher Education	<p>Minimum of grade D in all pre-requisite courses</p> <p>These are default marks/grades. The Board of Examiners may exercise some discretion in accordance to University Regulations</p>

Part C. Additional Award Requirements

Honours degree classification is determined by performance in:

- Stage 3 averaged over all 8 courses (20%) at the first attempt
- The 5 assessed courses in Stage 4 (50%)
- The individual dissertation project in Stage 4 (30%)

15. ADDITIONAL PROGRAMME INFORMATION

16. Programme Accredited by	British Computer Society	17. QAA Subject Benchmarking Group(s)	Computing	18. UCAS Code(s)	I100
------------------------------------	---------------------------------	--	------------------	-------------------------	-------------

Programme Description – Computer Systems						
Programme Code(s) (recruitment & exit awards) F2CC – CSE F2CD - CSE		Programme Titles for all awards (unabbreviated) Computer Systems		Main Award(s) (to be recruited to) BSc Honours (F2CC-CSE),		Exit Awards (for graduation only) BSc (Ord) F2CD-ZZZ BSc (Hons) F2CC-CSE
Type School degree	specialist	Programme Accredited by British Computer Society	UCAS Code I100	School Mathematical & Computer Sciences	QAA Subject Benchmarking Group(s) Computing	Date of Production/ Revision 09 October 2013
Educational Aims of the Programme						
The educational aim is to provide students with a blend of Computer Science and Information Systems. Students will gain expertise in management and socio-technical systems together with a theoretical foundation in, and applied skills with, Computer Science. They will also be provided with professional skills which will enable graduates to communicate clearly, work effectively and efficiently in industry and commerce.						
The Programme provides opportunities for learners to achieve the following outcomes:						
Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills</i></p> <ul style="list-style-type: none"> ◆ To develop knowledge and skills in the elicitation and analysis of user requirements, design and evaluation of solutions, and the implementation and quality assurance of the chosen solution. ◆ To be able to develop well-structured, efficient, usable and well-documented programs. ◆ To know what general classes of problems are amenable to computer solution and be able to select the appropriate tools required for particular problems. ◆ To be able to develop an abstract model for a given problem and devise appropriate mechanized techniques to solve the problem. • To develop knowledge of the aspects of management required to understand the commercial and business contexts within which information systems are used. <p><i>Scholarship, Enquiry and Research</i></p> <ul style="list-style-type: none"> ◆ To gain an understanding of the theoretical foundations of computation and its relevance to everyday computing. ◆ To be able to design, implement, document, verify and validate relatively large heterogeneous software systems. ◆ To be able to assess the quality of software systems, both in terms of their functional and non-functional properties. 					

Personal Abilities	
	<p><i>Industrial, Commercial and Professional Practice</i></p> <ul style="list-style-type: none"> ◆ To maintain and update technical knowledge; to take responsibility for personal and professional development. ◆ To appraise the impact of computers on society and the influence of society on the development of the technology and use of computers. ◆ To assess aspects of the law related to computer-based information, or the role of standards in safety, quality and security, of security issues and of the BCS Codes of Practice and Conduct.
	<p><i>Autonomy, Accountability and Working with Others</i></p> <ul style="list-style-type: none"> ◆ To undertake self-directed work; to assimilate information from multiple sources; to examine results and generate conclusions; to impart ideas effectively in visual, verbal or written form. ◆ To work effectively either individually or as part of a team. ◆ To apply subject-mastery outcomes to monitor, analyse, model, specify, design, communicate, implement, evaluate, control and plan. ◆ To be aware of, and be able to respond to, the social and legal implications and consequences of the use of computers. ◆ To be able to analyse problem spaces; develop and work with abstractions; appraise material and ideas; to apply a methodical and innovative approach to problem solving; to integrate theory and practice <p><i>Communication, Numeracy and ICT</i></p> <ul style="list-style-type: none"> ◆ To be able to present, analyse and interpret numerical and graphical data ◆ To be able to communicate effectively, informally or formally, to knowledgeable or lay audiences.

Approaches to Teaching and Learning:

Lectures, Tutorials (practical, laboratories), Coursework, (assignments, individual projects, group projects, essays, reports, presentations, log/journals, dissertation), Self-study are linked to *lecture-based, resource-based and problem-based* teaching styles, to relate with *motivational, assimilative, consolidative and evaluative* phases of learning.

Approaches to teaching and learning are continually reviewed and developed with the aim of matching them to the abilities and experiences of students, with regard also for the subject area. Specific details about teaching and learning methods are provided in the appropriate course descriptors.

Assessment Policies:

The following assessment methods are used:

Understanding, knowledge and subject specific skills are assessed through the range of methods reflected by *written examinations, coursework assignments, software artefacts, group and individual projects, written reports and oral presentations.* *Diagnostic, formative, continuous and summative* types of assessment aim to correlate with methods of assessment.

Approaches to assessment are continually reviewed. Specific details about methods of assessment are provided in the appropriate course descriptors.

The accompanying Programme Structure template provides details of courses, awards and credits for the programme.

The accompanying Programme Notes provide details of stage notes, progression requirements and award requirements for the programme.

BSc Computer Systems

Course Descriptors

Year 1, Semester 1

The University reserves the right to withdraw or modify the content of any course

Course Code: F27SA	Course Title: Software Development 1	Course Co-ordinator: Katrín Lohan/Peter King
Pre-requisites:	None	
Aims:	To introduce the object-oriented paradigm and the use of an object-oriented language	
Syllabus:	<ul style="list-style-type: none"> ◆ Objects and classes ◆ Class definitions: fields, constructors, methods, parameters ◆ Selection and iteration ◆ Object interaction: abstraction, modularisation, types ◆ Grouping objects: collection classes, iterators, arrays ◆ Library classes, documentation ◆ Testing and debugging ◆ Designing classes: coupling, cohesion, main method 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Understanding the object-oriented paradigm ◆ Awareness of the contrast with other programming paradigms ◆ Manipulating objects in an IDE ◆ Understanding and using documentation in an API ◆ Reading, understanding, adapting, creating, and documenting object-oriented code 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Sharing work with random partners in laboratories (pair programming) ◆ Deriving and creating own solutions to problems (PDP) ◆ Competence in the use of a command-line shell (PDP) ◆ Reading and evaluating code, and modifying it 	
Assessment Methods:	Assessment: Examination: (weighting – 50%) Coursework: (weighting – 50%)	Re-assessment: Examination: (weighting – 100%)

Course Code: F27IS	Course Title: Interactive Systems	Course Co-ordinator: Frank Broz/Phil Barker
Pre-requisites:	None	
Aims:	To give students an opportunity to explore current technological media and creative approaches	
Syllabus:	<ul style="list-style-type: none"> ◆ Basic comparison and evaluation of designs and prototypes ◆ Reflecting on one's own learning and progress ◆ Development of Interactive Systems, for example <ul style="list-style-type: none"> - Web site development: page layout, navigation, graphics, animation/interaction - Game development using a current game authoring tool: level design, storyline, game mechanics 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ To give students experience of designing and developing an interactive system. ◆ To give students experience of evaluating and critiquing interactive systems. 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ To gain an awareness of the benefits and pitfalls of different approaches to multimedia project work ◆ To raise awareness of the legal and ethical responsibilities within the discipline ◆ To appreciate and enjoy the challenges of creative work (PDP) ◆ To take responsibility for one's own learning and managing workload (PDP) ◆ For students to appreciate their own strengths and weaknesses, and what is possible within time constraints (PDP) ◆ To develop skills in written, oral and media based communication (PDP) ◆ To present solutions to design challenges in the subject area (PDP) ◆ To develop experience and skills in giving and receiving constructive criticism (PDP) 	
Assessment Methods:	Assessment: Coursework: (weighting – 100%)	Re-assessment: Coursework: (weighting – 100%)

Course Code: F27PX	Course Title: Praxis	Course Co-ordinator: Tessa Berg
Pre-requisites:	None	
Aims:	<ul style="list-style-type: none"> ◆ To instruct students in undertaking self-directed study ◆ To instruct students in presenting their findings ◆ To acquaint students with the work of the department ◆ To deepen students' understanding of the degree courses for which they are registered ◆ To familiarise students with the computer systems used by the department ◆ To present skills that students could use to become successful students 	
Syllabus:	<ul style="list-style-type: none"> ◆ Writing reports; sources and referencing; group presentation; the matter of plagiarism ◆ Current departmental research ◆ Exploration of the departmental computer system ◆ Study skills 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Acquaintance with new research in computing ◆ Consideration of difficult and even perplexing ideas in their chosen field of study ◆ Knowledge of and ability to use departmental computer systems 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Undertaking responsibility for self-directed research ◆ Assimilating information from multiple sources ◆ Analysing results to formulate conclusions ◆ Writing reports to professional standards ◆ Constructively evaluating the work of peers ◆ Reacting sensibly to peer evaluation ◆ Re-writing work in response to criticism ◆ Co-operating in a group to investigate a complex topic ◆ Making spoken and visual presentations ◆ Acquiring study skills that can be used for both academic studies and later in a professional context 	
Assessment Methods:	Assessment: Coursework: (weighting – 100%)	Re-assessment: Coursework: 100%

Elective

You should choose one of the following courses to take in semester 1:

Course Code	Semester	Course Title
A17IB	1	Introductory Biology
C97NY	1	Introduction to Psychology 1
C27OA	1	Introductory Economics
C47AX	1	Arabic Beginners Elective 1
C47BX	1	British Sign Language 1
C47CE	1	Chinese Beginners Elective 1
C47FI	1	French Intermediate Elective 1
C47FX	1	French Post-Beginners Elective 1
C47GE	1	Gaelic Beginners Elective 1
C47GP	1	German Post-Beginners Elective 1
C47GX	1	German Beginners Elective 1
C47SP	1	Spanish Post-Beginners Elective 1
C47SX	1	Spanish Beginners Elective 1
F17LP	1	Logic & Proof

For further information on electives, see the links below:

C-Coded courses

<http://www2.hw.ac.uk/sml/undergraduate/2016-2017/courses/index.html>

F17LP

<http://www.macs.hw.ac.uk/students/mathscourses/f17lp-logic-and-proof/>

You will need to physically input the code for this elective in your online registration!

BSc Computer Systems

Course Descriptors

Year 1, Semester 2

Course Code: F27SB	Course Title: Software Development 2	Course Co-ordinator: Verena Rieser/Michael Lones
Pre-requisites:	None	
Aims:	<ul style="list-style-type: none"> ◆ To impart further techniques of object orientation ◆ To introduce simple data structures and algorithms 	
Syllabus:	<ul style="list-style-type: none"> ◆ Inheritance and Generics: hierarchies, subclasses, polymorphism, static and dynamic type, overriding, dynamic method lookup. ◆ Designing classes: coupling, cohesion, abstraction, modularisation, types ◆ Abstract classes, abstract methods, interfaces ◆ State machines & state diagrams GUIs: components, layout, event handling ◆ Error-handling: defensive programming, exceptions, assertions, JUnit tests ◆ Collection classes 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Understanding and application/ mastery of the object-oriented paradigm ◆ Understanding of inheritance and generics. ◆ Understand how to produce well designed, i.e. extendable and maintainable, code. ◆ Ability to critically evaluate and improve the quality of code. ◆ Ability to develop simple state diagrams ◆ Ability to design and implement simple graphic user interfaces 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Possession of fundamental skills in computer science, applicable throughout the remainder of the degree ◆ Understand challenges in developing , designing and maintaining code. ◆ Understanding of the importance of regular working habits (pdp) ◆ Understanding of the use of chatboards and other devices to learn from and instruct others in the class (pdp) ◆ Ability to compare and evaluate the applicability of simple data structures and code design choices to relevant problems (pdp) 	
Assessment Methods:	<p>Assessment: Examination: (weighting – 50%) Coursework: (weighting – 50%)</p>	<p>Re-assessment: Examination: (weighting – 100%)</p>

Course Code: F27SG	Course Title: Software Development 3	Course Co-ordinator: Gudmund Grov
Pre-requisites:	None	
Aims:	To develop further skills and techniques in programming in a high-level language.	
Syllabus:	<ul style="list-style-type: none"> ◆ Static structures – tables ◆ Linear techniques e.g. Search, delete, update ◆ String & text processing ◆ Dynamic structures - stacks & queues ◆ Recursive techniques – linear recursion, accumulation recursion ◆ Sorting & searching e.g. Binary search, quicksort, merge sort, hash tables ◆ Linked structures – lists – construction, traversal, delete, update ◆ Linked structures – trees – construction, traversal, delete, update, balance ◆ File processing ◆ Introductory complexity & “big o” notation 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ To understand properties of and algorithms for fundamental static, dynamic and linked data structures ◆ To know when to deploy fundamental data structures and algorithms in practical problem solving ◆ To gain mastery of fundamental linear and recursive programming techniques ◆ To know when to deploy linear and recursive programming techniques in practical problem solving ◆ To understand fundamental techniques for processing very large data sets from files ◆ To gain skill in elementary analyses of fundamental algorithms and data structures to give insight into their time and space complexity bounds ◆ To understand correspondences between different programming techniques ◆ To understand correspondences between different data structures and algorithms 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ To understand how the choice of algorithms and data structures determines the efficacy of proposed solutions to problems ◆ To be able to explain the implications of choosing particular algorithms and data structures for the time and space behaviour of solutions 	
Assessment Methods:	Assessment: Examination: (weighting – 50%) Coursework: (weighting – 50%)	Re-assessment: Examination: (weighting – 100%)

Course Code: F27CS	Course Title: Introduction to Computer Systems	Course Co-ordinator: Peter King
Pre-requisites:	None	
Aims:	<ul style="list-style-type: none"> ◆ To introduce students to modern computer systems architecture ◆ To give students an appreciation of logical design and data representation 	
Syllabus:	<ul style="list-style-type: none"> ◆ Overview ◆ Hardware components - peripherals, memory & CPU. ◆ Boolean algebra. ◆ Low-level information representation. ◆ CPU organisation. ◆ Introductory assembly language programming. ◆ Operating system: I/O; interrupts; scheduler; virtual memory; file system. ◆ Concurrency: processes; threads; synchronisation; shared & distributed memory; distributed & parallel architectures. ◆ Language processors: compiler; interpreter; assembler; loader. ◆ Linux shell scripting 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Overview of hardware/software hierarchy in contemporary computer systems; ◆ Understanding of purpose and function of major system hardware and software components; ◆ Understanding of information representation in computer systems; ◆ Ability to write Linux shell scripting 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ To be able to express arguments/problems in propositional and predicate calculus. ◆ To be able to communicate in using formal notations 	
Assessment Methods:	<p>Assessment:</p> <p>Examination: (weighting – 70%)</p> <p>Class Tests: (weighting – 20%)</p> <p>Coursework: (weighting - 10%)</p>	<p>Re-assessment:</p> <p>Examination: (weighting – 100%)</p>

Course Code: F27WD	Course Title: Web Design and Databases	Course Co-ordinator: Santiago Chumbe/Frank Broz
Pre-requisites:	None	
Aims:	To develop knowledge and understanding of fundamental web design concepts and combine these with database structuring and querying techniques applying this knowledge by implementing an easy-to-use website.	
Syllabus:	<ul style="list-style-type: none"> ◆ Introduction to web development. ◆ Information architecture. ◆ Web design and usability. ◆ Fundamentals of Mark-up and CSS. ◆ Introduction to database systems. ◆ Databases and Information Systems. ◆ Modelling of data/entity-relationship modelling. ◆ The relational data model. ◆ The Structured Query Language (SQL). ◆ Web-based database applications including the use of PHP. 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ To explain fundamental web design concepts including usability. ◆ To implement a simple web site which satisfies current standards and uses a database. ◆ To describe the use of CSS and mark-up within a web site and the advantage this gives the developer. ◆ To describe the need for standard XHTML and how this aids cross browser compatibility. ◆ To have knowledge and understanding of data analysis and structuring techniques. ◆ To design database structures as a relational data model. ◆ To implement and query a designed database structure through a web site. 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ To analyse complex information and organise it in a structured way for a web site. ◆ To understand stakeholders' requirements and address them. ◆ To design a web site that is easy and cost efficient to manage. ◆ To analyse data sources and represent them in an efficient structured form. ◆ Problem solving (PDP). ◆ Paired work (PDP). ◆ Time management (PDP). ◆ Reflection, constructive criticism and learning from peers (PDP). 	
Assessment Methods:	Assessment: Examination: (weighting – 60%) Coursework: (weighting – 40%)	Re-assessment: Examination: (weighting – 100%)

BSc Computer Systems

Course Descriptors

Year 2, Semester 1

Course Code: F28IN	Course Title: Interaction Design	Course Co-ordinator: Diana Bental/Helen Hastie
Pre-requisites:	F27IS1 Interaction Systems or equivalent	
Aims:	<p>The course aims to give students the opportunity to develop:</p> <ul style="list-style-type: none"> ◆ A broad knowledge and understanding of requirements gathering, design and evaluation theory and techniques in interaction design. ◆ An introduction to commonly used design techniques and pattern for user interfaces. ◆ A selection of routine skills and methods involved in working with users. 	
Syllabus:	<p>Current topics in Interaction Design including: interaction design lifecycles, user interface design patterns, basic qualitative and quantitative data gathering and presentation techniques, accessibility.</p>	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Critically analyse interaction design and interfaces. ◆ Propose solutions in response to interface design problems ◆ Evaluate the effectiveness of user interfaces with respect to user requirements. 	
Learning Outcomes:: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Use discipline appropriate software for data analysis, ◆ Present, analyse and interpret simple numerical and graphical data gathered as part of evaluation studies. (PDP) ◆ Communicate effectively to knowledgeable audiences by preparing informal presentations and written reports. (PDP) ◆ Exercise autonomy and initiative by planning and managing their own work within a specified project; (PDP) ◆ Take responsibility for their own and other's work by contributing effectively and conscientiously to the work of a group (PDP) 	
Assessment Methods:	<p>Assessment:</p> <p>Examination: (weighting – 60%)</p> <p>Coursework: (weighting – 40%)</p>	<p>Re-assessment:</p> <p>Examination: (weighting – 100%)</p>

Course Code: F28WP	Course Title: Web Programming	Course Co-ordinator: Hamish Taylor/Santiago Chumbe
Pre-requisites:	F27WD Web Design and Databases <i>or equivalent</i>	
Aims:	To familiarise students with current techniques and paradigms in web programming. To enable them to design and implement robust and scalable web based applications.	
Syllabus:	<ul style="list-style-type: none"> ◆ History of web development technologies ◆ Design patterns (such as REST, Separation of content and presentation, and abstraction of resources) ◆ Server side programming using an appropriate scripting language ◆ General architecture of a web server ◆ Templating systems ◆ Client side programming topics, including the Document Object Model ◆ Security relating to web applications ◆ Deployment, including coping with scale 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Broad knowledge and understanding of the history of web programming ◆ The ability to apply the concepts, patterns and architectures used in web programming to new problems ◆ Detailed technical skills to use a scripting language for both server side and client side programming ◆ The ability to make informed decisions about appropriate web technologies to use for a particular task 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Practice in working on a development project in a small group under the guidance of a tutor ◆ Practice in defining the subject and scope of a development project (PDP) ◆ Deconstructing a problem and synthesizing a solution ◆ Time management 	
Assessment Methods:	Assessment: Examination: (weighting – 50%) Coursework: (weighting – 50%)	Re-assessment: Examination: (weighting – 100%)

Course Code: F28DA	Course Title: Data Structures & Algorithms	Course Co-ordinator: Manuel Maarek
Pre-requisites:	F27SB2 Software Development 2 or equivalent	
Aims:	<ul style="list-style-type: none"> ◆ To introduce core algorithms and data structures used in a wide range of applications in Computer Science ◆ To further develop skills in algorithm and data structure design, and the development of medium sized programs 	
Syllabus:	<ul style="list-style-type: none"> ◆ Algorithm and data structure topics including: advanced trees, string processing, graphs, hash tables ◆ Algorithm/data structure choice, design and deployment 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Ability to analyse and hence choose suitable algorithms and data structures for a given problem ◆ To design and implement medium sized programs based on a range of standard algorithms and data structures and making appropriate use of libraries ◆ Understanding the distinction between abstract Algebraic Data Type (ADT) properties and concrete ADT realisations ◆ Appreciation of need for integration of multiple ADTs in substantial programs ◆ Appreciation of efficiencies/reassurances from ADT reuse 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ To be able to critically analyse and hence choose suitable algorithms and data structures for a given problem ◆ To be able to convey the advantages and disadvantages of alternative data structures and algorithms ◆ To develop practical problem-solving skills in the context of programming ◆ To be able to plan & execute a substantial software 	
Assessment Methods:	Assessment: Examination: (weighting – 60%) Coursework: (weighting – 40%)	Re-assessment: Examination: (weighting – 100%)

Course Code: F28PL	Course Title: Programming Languages	Course Co-ordinator: Jamie Gabbay
Pre-requisites:	Software Development 2 (F27SB), Introduction to Computer Systems (F27CS)	
Aims:	<p>To gain understanding of different language paradigms</p> <p>To gain understanding of defining concepts of programming languages</p> <p>To develop skills in programming in languages from key paradigms</p>	
Syllabus:	<ul style="list-style-type: none"> ◆ Overviews of language history, definition (lexicon, syntax, semantics), implementation (compiler, interpreter, virtual machine) ◆ Overviews of language paradigms: e.g. imperative (high-level, system, low-level), declarative (functional, logic), concurrency/parallelism ◆ Overviews of programming language concepts: variable, lvalue & rvalue, assignment (sharing/copying), data abstraction (sequential, structured, recursive, shared/distributed), type mechanisms (weak/strong, static/dynamic, ad-hoc/parametric polymorphism), declaration (scope, extent), control abstraction (sequence, choice, repetition, block, procedure, labels/jumps, exceptions, processes), expression abstraction (functions), parameter mechanisms (value, reference) , evaluation mechanisms (strict/lazy, ordered/unordered, concurrent) ◆ An introduction to programming in languages from key paradigms e.g. <ul style="list-style-type: none"> ○ scripting: e.g. Python ○ declarative/functional: e.g. SML ○ declarative/logic: e.g. Prolog 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Understanding of distinguishing characteristics of language paradigms ◆ Understanding of relationships between languages ◆ Understanding of generic language concepts ◆ Ability to program in languages from key paradigms ◆ Ability to use tool sets for these languages. 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Understanding of how to choose an appropriate language for different problem domains 	
Assessment Methods:	<p>Assessment:</p> <p>Examination: (weighting – 70%)</p> <p>Coursework: (weighting – 30%)</p>	<p>Re-assessment:</p> <p>Examination: (weighting – 100%)</p>

BSc Computer Systems

Course Descriptors

Year 2, Semester 2

Course Code: F28SD	Course Title: Software Design	Course Co-ordinator: Patricia Vargas/tbc
Pre-requisites:	F27SA – Software Development 1 OR equivalent	
Aims:	<ul style="list-style-type: none"> ◆ An introduction to a range of processes and methods that promote the design of high quality software systems. ◆ A perspective of where design sits within the development life-cycle 	
Syllabus:	<ul style="list-style-type: none"> ◆ Software process models; ◆ Architectural styles; ◆ Design methods and associated notations – including function-oriented, object-oriented and component-based design; design patterns; ◆ Software development life-cycle issues, with particular focus on requirements engineering, validation and verification (testing). 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ To demonstrate a critical understanding of software process models and design methods. ◆ To be able to develop design solutions using a range of structured notations. ◆ To demonstrate a critical understanding of the context in which software design takes place, in particular requirements engineering and the activities of validation and verification (testing). 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Take responsibility for own work and exhibit critical reflection on development process. (PDP) ◆ To be able to use appropriate methods and standards for practice and documentation in software engineering and information systems. ◆ Demonstrate evidence based approaches to problem solving. ◆ Use a range of numerical and graphical skills in evaluating and communicating ideas, as well as measuring progress toward achieving goals 	
Assessment Methods:	<p>Assessment:</p> <p>Examination: (weighting – 60%)</p> <p>Coursework: (weighting – 40%)</p>	<p>Re-assessment:</p> <p>Examination: (weighting – 100%)</p>

Course Code: F28DM	Course Title: Database Management Systems	Course Co-ordinator: Alasdair Gray
Pre-requisites:	F27WD Web design & Databases or equivalent	
Aims:	To familiarise students with the principles of database management systems, to enable them to design and implement databases for specific applications and to integrate databases with application programs.	
Syllabus:	<p>Database Design: data requirements, entity relationship diagrams, relational data model, integrity constraints, key constraints, types, integrity maintenance</p> <p>Relational Queries: SQL, Boolean combinations of queries, aggregation, duplicate elimination, nested queries, negation, views, insertions, deletions, updates, command level interfaces, integration with programming application</p> <p>Query execution and optimisation: data storage principles, file organisation, indexing, indexes in commercial DBMSs, relational algebra, query execution plans, cost estimation of plans, interpretation of plans, physical database design</p> <p>Concurrency: transactions, schedules, serialisability, concurrency control protocols, locking, two-phase-locking, time stamp based concurrency control.</p> <p>Emerging Database Trends: data warehousing, distributed databases, and alternative database models such as XML, document, object, and graph stores</p>	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Broad knowledge and understanding of the concepts and formalisms of database design ◆ Detailed knowledge of the building blocks and meaning of relational database queries ◆ Critical understanding of the principles of query evaluation and concurrency control underlying database applications ◆ Practice in the collection of data requirements and the design of conceptual database schemas ◆ Evaluation of emerging database trends and ability to understand their benefits 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Practice in working on a development project in small groups (PDP) ◆ Practice in defining the subject and scope of a development project (PDP) ◆ Deconstructing a problem and synthesizing a solution (PDP) ◆ Time management (PDP). 	
Assessment Methods:	Assessment: Examination: (weighting – 70%) Coursework: (weighting – 30%)	Assessment: Examination: (weighting – 100%)

Course Code: F28HS	Course Title: Hardware-Software Interface	Course Co-ordinator: Greg Michaelson/Hans-Wolfgang Loidl
Pre-requisites:	F28HS Introduction to Computer Systems	
Aims:	<ul style="list-style-type: none"> ◆ To gain an understanding of low-level, hardware-oriented and systems programming. ◆ To develop skills in resource-conscious programming. ◆ To develop programming skills in such languages. 	
Syllabus:	<ul style="list-style-type: none"> ◆ Low-level, assembler programming ◆ Low-level, C programming ◆ Advanced computer architecture issues impacting software performance (caches, multi-cores, etc) ◆ Operating system interfaces for low-level software ◆ Operating system concepts such as device handling, interrupts, BIOS etc ◆ Embedded systems programming ◆ Resource-conscious programming techniques (memory, performance; programming techniques, tools, monitoring) 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Critical understanding of computer architecture concepts and their performance implication for low-level software. ◆ Detailed theoretical and practical understanding of hardware and operating system concepts, interfacing to low-level software. ◆ Ability to develop efficient, resource-conscious code, interfacing to hardware components. ◆ Practical skills in low-level, systems programming, with effective resource management. 	
Learning Outcomes:: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <p>Ability to articulate system-level operations and to identify performance implications of given systems</p>	
Assessment Methods:	<p>Assessment:</p> <p>Examination: (weighting – 60%)</p> <p>Coursework: (weighting – 40%)</p>	<p>Re-assessment:</p> <p>Examination: (weighting – 100%)</p>

Course Code: F17SC	Course Title: Discrete Mathematics	Course Co-ordinator: Anatoly Konechny
Pre-requisites:	None	
Aims:	The goal of the course is to explain the basic techniques of discrete mathematics which are used in computer science.	
Syllabus:	<ul style="list-style-type: none"> ◆ Set Theory and Combinatorics: Set algebra, The Inclusion-Exclusion Principle, Binomial Theorem, Elementary counting methods, Mathematical Induction. (6 lectures) ◆ Graph Theory 1: Introduction to graphs, Basic graph terminology. Adjacency Matrices, Paths and connectivity, Connected components, Shortest path problems in weighted graphs, Dijkstra's Algorithm. (5 lectures) ◆ Graph Theory 2: Trees and spanning trees, Breadth-first search, Kruskal's and Prim's Algorithms for a minimal spanning tree, Euler and Hamilton paths, Fleury's Algorithm for constructing Euler circuits, Prefix Codes and Huffmann Coding. (6 lectures) ◆ Recurrence Relations: Solving problems by iteration, First and second order recurrence relations, Recurrences in Algorithms (3 lectures) ◆ Matrices and Linear Transformations: Linear equations and elementary row operations, Elementary matrices and Gaussian elimination, Echelon matrices, Matrices as space transformations, Eigenvectors and eigenvalues, Diagonalization, The rank theorem. (8 lectures) ◆ Probability Theory: Probability Space, Conditional Probability, Independence and Bayes' Theorem, Random Variables and Distributions, Expected Value, Variance, Examples of applications to algorithms (5 lectures). 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Know the basic terminology of set theory, graph theory, liner algebra and probability theory ◆ Understand how formal mathematical objects like sets, graphs, matrices, recurrence relations arise in computer science related problems ◆ Be able to solve elementary counting problems, solve systems of linear equations, apply graph algorithms, solve simple recurrence relations, be able to compute probabilities ◆ Appreciate the power of mathematical formalisation, facilitated by the use of precise definitions and notations, in solving practical problems. ◆ Appreciate the value of careful, quantitative reasoning in analysing problems related to computer science and to recognise that the outcome of such reasoning can defy naïve intuition 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Understand and use abstract mathematical concepts ◆ Use logical reasoning to prove theorems ◆ Communicate mathematical orally and in writing 	
Assessment Methods:	Assessment: Examination: (weighting – 70%) Coursework: (weighting – 30%)	Re-assessment: Examination: (weighting – 100%)

Course Code: F28CD	Course Title: Creative Design Project	Course Co-ordinator: Jenny Coady/tbc
Pre-requisites:	F27IS Interactive Systems, F27DB Introduction to Databases	
Aims:	<ul style="list-style-type: none"> ◆ To consolidate multimedia design, prototyping and implementation skills through a realistic project. ◆ To develop an appreciation and experience of group processes 	
Syllabus:	<p>Introduction to multimedia scenarios and prototyping</p> <p>Introduction to group processes</p> <p>Conduct multimedia project in structured stages: develop scenario, construct/evaluate prototype, iterative refinement to implementation, documentation, evaluation</p>	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ To integrate multimedia design and programming knowledge ◆ To develop proficiencies in multimedia application design 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ To gain practical experience of design, integration and coordination in the context of a realistic project ◆ To acquire group work skills (PDP) ◆ Developing personal responsibility for managing projects and updating module co-ordinator on progress (PDP) ◆ Understanding different perspectives on a project (PDP) 	
Assessment Methods:	Assessment: Coursework: (weighting – 100%)	Re-assessment: Individual Project

Course Code: C18OP	Course Title: Operations Management	Course Co-ordinator: Abhijeet Ghadge
Pre-requisites:	None	
Aims:	<ul style="list-style-type: none"> ◆ To acquire knowledge about managing operations & resources in organisations of various sizes; private, public & not-for-profit. ◆ To explore basic strategic & managerial perspectives & frameworks of operating systems. ◆ To develop an understanding of the many internal & external factors that impact on the development of effective operating systems. ◆ To recognise & appreciate the role of technology & its impact on operations management. ◆ To appreciate the impact decisions made by operations managers have on an organisation's competitive performance. ◆ To evaluate the role of the Operations Manager. ◆ To explore the international dimension of Operations Management. 	
Syllabus:	<ul style="list-style-type: none"> ◆ Operations management and strategy ◆ Social, environmental and economic performance ◆ Product and service design ◆ Process design ◆ Facilities location, layout and flow ◆ Supply network design ◆ Capacity management ◆ Inventory management ◆ Planning and control ◆ Lean synchronisation ◆ Quality and operations improvement 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Describe & explain the operations management concepts and techniques which can be used to support management decision making. ◆ Understand and use new developments in operations management thinking. ◆ Solve real operational problems by the application of theoretical and analytical operational models. ◆ Develop the ability to research a particular dimension of operations management. ◆ Develop analytical & evaluation skills. 	
Learning Outcomes: Personal Abilities:	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Demonstrate an interest and awareness of current developments in operations management. ◆ Explain the subjective, ever-changing and uncertain nature of the environment within which operations managers operate. ◆ Work independently and as part of a group. ◆ Develop time and project management skills. ◆ Communicate and present ideas effectively by written and verbal means. 	
Assessment Methods:	Assessment: Examination: (weighting – 60%) Coursework: (weighting – 40%)	Re-assessment: Examination: (weighting – 100%)

BSc Computer Systems

Course Descriptors

Year 3, Semester 1

Course Code: F29SO	Course Title: Software Engineering	Course Co-ordinator: Helen Hastie
Pre-requisites:	F28SD Software Design, F28DM Database Management, F28IN Interaction Design	
Aims:	<ul style="list-style-type: none"> ◆ To equip students with knowledge and skills for the effective management of a group project which encompasses the software development lifecycle ◆ To enable students to reinforce their knowledge and skills gained in software processes, internet technology, database management and interaction design ◆ To build students understanding, knowledge and skills in teamwork, software development in groups, and project planning. ◆ To enable students to develop a broader understanding of the interrelationship of development life-cycles and a critical capability in the selection of tools and methods to support project planning, systems analysis, requirements capture, and system specification. 	
Syllabus:	<ul style="list-style-type: none"> ◆ Review and extension of the components studied in earlier years which contribute to the group project ◆ Software project management including working in groups, project planning and costing, risk assessment ◆ Use of Industry-level Standards for software development and documentation, covering aspects such as change control and requirements traceability ◆ Further study of software development tools, especially version control 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ A broad and integrated understanding and knowledge of the various development and programming paradigms, software development life-cycles, teamwork and project planning ◆ Detailed theoretical and practical knowledge of the use of methodologies for requirements capture, iterative design, resource capture and management, deployment and evaluation of systems, at a basic level ◆ Practice in the use of object-oriented programming, databases, scripting and markup languages applied to a substantial project 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Identification, critical analysis and evaluation of the development of a software system (PDP) ◆ Practice in working in a group, negotiating requirements, reaching a consensus, taking responsibility for own work, taking part in a presentation, and working with others to a deadline (PDP) ◆ Appreciation of the interrelationship of knowledge domains 	
Assessment Methods:	Assessment: Group Project: (weighting – 50%) Examination: (weighting – 40%) Coursework: (weighting – 10%) Synoptic with F29PD Professional Development	Re-assessment: Coursework: (weighting – 100%)

Course Code: F29DC	Course Title: Data Communications and Networking	Course Co-ordinator: Peter King/ Idress Ibrahim
Pre-requisites:	F28WP Web Programming	
Aims:	<ul style="list-style-type: none"> ◆ Introduction to data communications and computer networking ◆ Understanding of the structure of Internet ◆ Understanding of concepts of connection oriented and connectionless communication, and principles of data communication protocols 	
Syllabus:	<ul style="list-style-type: none"> ◆ Internet history and organisation, OSI and Internet reference models. ◆ Link level communications, data transparency, error detection, sliding window protocols ◆ Network layer protocols, IP, ICMP, ◆ network routing, ◆ routers ◆ Transport protocols, TCP, UDP ◆ Congestion control ◆ Higher level protocols e.g. HTTP (simple example only) 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Subject-Specific Skills</i></p> <ul style="list-style-type: none"> ◆ Understanding of data communication protocols, ◆ Appreciation of necessity for formal specification and verification of protocols. ◆ Appreciation of complexity of network infrastructure and sensitivity to parameter choices 	
Learning Outcomes: Personal Abilities	<p><i>Cognitive skills, Core skills and Professional Awareness</i></p> <p><i>Professional Development</i></p> <ul style="list-style-type: none"> ◆ Appreciation of role of standards in networking ◆ Appreciation of precision and need for validation in specification of data communication protocols <p><i>Practical Expertise</i></p> <ul style="list-style-type: none"> ◆ Ability to analyse and explain basic issues relating to communication and networking technologies ◆ Practice in ICT, numeracy and report writing, team working 	
Assessment Methods:	Assessment: Examination: (weighting– 75%) Coursework: (weighting – 25%)	Re-assessment: Examination: (weighting – 100%)

Course Code: F29KM	Course Title: Knowledge Management	Course Co-ordinator: Jenny Coady
Pre-requisites:	None	
Aims:	<ul style="list-style-type: none"> ◆ To provide students with an overview of information and knowledge management in organisations ◆ To critically evaluate a range of methods used to develop strategies for information and knowledge management ◆ To examine the role that knowledge and users play in the learning organisation ◆ To critically evaluate the value of knowledge and IT for competitive advantage 	
Syllabus:	<ul style="list-style-type: none"> ◆ Information and Knowledge Management in Organisations ◆ Principles of Knowledge Management: information mapping and information audits ◆ Knowledge elicitation and representation ◆ Information strategy development ◆ Knowledge and IT for competitive advantage and as a corporate resource ◆ The learning organisation ◆ Planning for Knowledge Management within an organisation and the ethical issues which arise 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Differentiating between Data, Information and Knowledge ◆ Understand and evaluate theories and practices of knowledge management in organisations ◆ Critically evaluate the value of knowledge and IT for competitive advantage ◆ Compare and contrast methods to develop strategies and planning for knowledge management within organisations ◆ Examine the rise of the concept of the Learning organisation and how it can aid in competitive advantage 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Evaluating policies and strategies ◆ Planning for large scale organisations ◆ Ability to manage directed reading with self research (PDP) ◆ Report writing and demonstrating argument development (PDP) ◆ Use of technology as a means of learning, contributing and discussing (PDP) 	
Assessment Methods:	<p>Assessment:</p> <p>Examination: (weighting – 60%)</p> <p>Coursework: (weighting – 40%)</p>	<p>Re-assessment:</p> <p>Examination: (weighting – 100%)</p>

Course Code: F29AI	Course Title: Artificial Intelligence & Intelligent Agents	Course Co-ordinator: Ron Petrick/Patricia Vargas
Pre-requisites:	Elementary knowledge of logic at the level of undergraduate Computer Science. Knowledge of high-level programming language concepts.	
Aims:	<ul style="list-style-type: none"> ◆ To introduce the fundamental concepts and techniques of AI, including planning, search and knowledge representation ◆ To introduce the scope, subfields and applications of AI, topics to be taken from a list including natural language processing, expert systems, robots and autonomous agents, machine learning and neural networks, and vision. ◆ To develop skills in AI programming in an appropriate language 	
Syllabus:	<ul style="list-style-type: none"> ◆ Search algorithms (depth first search, breadth first search, uniform cost search, A* search) ◆ constraint satisfaction problems ◆ games (min-max, alpha-beta pruning) ◆ logic, resolution, introductory logic programming ◆ knowledge representation – logic, rules, frames ◆ goal and data-driven reasoning ◆ practical rule-based programming ◆ Overview of main fields of AI (Vision, Learning, Knowledge Engineering) ◆ In depth view of one field of AI (e.g. Planning, Natural language) ◆ Autonomous agents ◆ Applications of AI ◆ AI programming 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Critical understanding of traditional AI problem solving and knowledge representation methods ◆ Use of knowledge representation techniques (such as predicate logic and frames). ◆ Critical understanding of different systematic and heuristic search techniques ◆ Practice in expressing problems in terms of state-space search ◆ Broad knowledge and understanding of the subfields and applications of AI, such as computer vision, machine learning and expert systems. ◆ Detailed knowledge of one subfield of AI (e.g. natural language processing, planning) and ability to apply its formalisms and representations to small problems ◆ Detailed understanding of different approaches to autonomous agent and robot architectures, and the ability to critically evaluate their advantages and disadvantages in different contexts. ◆ Practice in the implementation of simple AI systems using a suitable language 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Identification, representation and solution of problems ◆ Time management and resource organisation ◆ Research skills and report writing ◆ Practice in the use of ICT, numeracy and presentation skills 	
Assessment Methods:	Assessment: Examination: (weighting – 70%) Coursework: (weighting – 30%)	Re-assessment: Examination: (weighting – 100%)

BSc Computer Systems

Course Descriptors

Year 3, Semester 2

Course Code: F29PD	Course Title: Professional Development	Course Co-ordinator: Nick Taylor
Pre-requisites:	F29SO Software Engineering (Synoptic)	
Aims:	<ul style="list-style-type: none"> ◆ To instil a professional and ethical attitude toward the application of computer technology ◆ To introduce methods for the rational resolution of ethical problems ◆ To provide an appreciation of the relevant professional and legal requirements concerning computer-based systems ◆ To ensure an awareness of, and encourage deliberation about, the social implications of information technology 	
Syllabus:	<p>Professionalism - British Computer Society.</p> <p>Rules & Regulations -Codes & Standards; Computer Law; Ethical Decision Making.</p> <p>Risks & Threats - Computer Crime; Viruses.</p> <p>Privacy & Security – Databases; Biometrics.</p> <p>Dependence & Change - Safety-Critical Systems; Technology & Society.</p> <p>Brave New Worlds - Co-operative Computing; eLife.</p>	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <p>Broad and integrated knowledge and understanding of:</p> <ul style="list-style-type: none"> ◆ British Computer Society Codes - Conduct; Practice ◆ ISO & BSI Standards - Safety; Quality; Security ◆ Statute Law - Contracts, Torts, Restitution; Data Protection; Freedom of Information, Intellectual Property; Computer Misuse ◆ Ethics - Frameworks; Decision Making ◆ <p>Critical understanding and detailed knowledge of:</p> <ul style="list-style-type: none"> ◆ Development life-cycle of a software system ◆ Bi-directional influence between technological and societal trends ◆ Current concerns over the application of computer technology ◆ Current and potential remedies to abuse of computer technology 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Practice in personal decision making and introspection ◆ Identification and analysis of justification of personal choices to others ◆ Critical analysis of rational reasoning, consequential reasoning and debate ◆ Practice and reflective analysis of communication skills using a variety of media ◆ Practice in working in a group, negotiating requirements, reaching a consensus, and working with others to a deadline 	
Assessment Methods:	<p>Assessment:</p> <p>Group Project: (weighting – 50%)</p> <p>Examination: (weighting – 40%)</p> <p>Coursework: (weighting – 10%)</p> <p>Synoptic with F29SO Software Engineering</p>	<p>Re-assessment:</p> <p>Coursework: (weighting – 100%)</p>

Course Code: F29OC	Course Title: Operating Systems & Concurrency	Course Co-ordinator: Sven Bodo-Scholz
Pre-requisites:	None	
Aims:	<ul style="list-style-type: none"> ◆ For the Operating system part: To provide an introduction to operating systems, their basic principles and shell programming. ◆ For the Concurrency part: To introduce the theory and practice of concurrent hardware and software systems 	
Syllabus:	<ul style="list-style-type: none"> ◆ For the Operating system part: overview on operating systems concepts and structures, processes, threads, classical inter-process communication problems, memory management ◆ For the Concurrency part: Concurrency, Parallelism, Pthreads. Parallelism Pattern: Pipelining, Data-Parallelism, Nested Data-Parallelism, Flattening, Task-Parallelism, Data-Flow. 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <p>For the Operating systems part:</p> <ul style="list-style-type: none"> ◆ Understanding of the concepts and structures present in modern operating systems. <p>For the Concurrency part:</p> <ul style="list-style-type: none"> ◆ Broad and integrated knowledge and understanding of concurrency concepts, techniques and problems ◆ Critical understanding of predominant concurrency pattern and their implementation on modern architectures ◆ Hands-on experience 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Critically evaluate the problematic and concepts related to operating systems. ◆ Analysis of the different possible solutions to leveraging concurrency for parallel execution. 	
Assessment Methods:	<p>Assessment:</p> <p>Examination: (weighting– 60%)</p> <p>Coursework: (weighting – 40%)</p>	<p>Re-assessment:</p> <p>Examination: (weighting – 100%)</p>

Course Code: F29LP	Course Title: Language Processors	Course Co-ordinator: Jamie Gabbay/Gudmund Grov
Pre-requisites:	F28PL Programming Languages	
Aims:	<ul style="list-style-type: none"> ◆ To develop skills in programming language definition ◆ To develop skills in programming language implementation 	
Syllabus:	<p>Language definition:</p> <ul style="list-style-type: none"> ◆ lexicon: non-terminal & terminal symbols, type 3 grammars, regular expressions, finite state automata, Moore & Mealey machines, left & right recursion ◆ concrete syntax: type 2 grammars & BNF, factoring grammars, converting left to right recursion, LL(K) & LR(K) grammars, push down automata, parsing ◆ abstract syntax: abstract syntax from concrete syntax ◆ static semantics: types ◆ dynamic semantics: overviews of axiomatic, denotational & operational semantics; introduction to structural operational semantics (SOS) for declarations, expressions, statements <p>Language implementation:</p> <ul style="list-style-type: none"> ◆ lexical analysis ◆ syntax analysis: recursive descent for LL(1) ◆ abstract syntax tree (AST) construction ◆ AST pretty printing ◆ overview of interpreter/abstract machine ◆ AST interpretation ◆ static analysis of AST: e.g. type checking, identifier resolution ◆ code generation e.g.: register/memory allocation, structured constructs as test/jump, structured data as structured memory sequences, stack discipline for blocks/subroutines, parameter passing ◆ run time environment: e.g. heap allocation, garbage collection, debugging support ◆ compiler construction tools e.g. Lex, Yacc, JavaCC <p>Based on simple imperative language</p>	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Ability to construct & manipulate lexical & syntactic definitions ◆ Understanding of relationship between grammars & automata ◆ Ability to construct lexical & syntactic analysers from definitions ◆ Familiarity with static & dynamic semantics, and associated formalisms ◆ Ability to generate & traverse ASTs ◆ Ability to construct AST-based analysers, interpreters & code generators for simple languages ◆ Understanding of run-time environments ◆ Ability to use compiler construction tools 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Appreciation of problem solving as language definition & implementation ◆ Ability to apply appropriate language oriented formalisms, tools & techniques in solving problems ◆ Skills in engineering complex software artefacts 	
Assessment Methods:	Assessment: Examination: (weighting – 60%) Coursework: (weighting – 40%)	Re-assessment: Examination: (weighting – 100%)

Course Code: F29SS	Course Title: Sociotechnical & Soft Systems	Course Co-ordinator: Jenny Coady
Pre-requisites:	None	
Aims:	<p>The course aims to give students the opportunity to develop an understanding and an ability to apply Checkland and Wilson's Soft Systems Methodology (SSM). They will be introduced to systems thinking as a means of analysing the whole context of an information system. The use of techniques such as rich pictures and other diagrammatical notations will be used to allow analysis to incorporate all stakeholders. Students will learn to adopt a critical approach of evaluating socio-technical systems based on an understanding of their role as a reflective fractioned. Examine the underlying frameworks, understand the issues that arise when characterizing problems and practical use of these skills will be developed through exercises based on case studies.</p>	
Syllabus:	<ul style="list-style-type: none"> ◆ The problem situation unstructured ◆ The problem situation expressed ◆ Root definitions of relevant systems ◆ Making and testing conceptual models ◆ Comparing conceptual models with reality ◆ Identify feasible and desirable changes ◆ Action to improve the problem situation Systems thinking ◆ Socio-technical systems methods ◆ Modelling frameworks and issues ◆ People centred solutions ◆ Checkland's theories and thinking 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Critically analyse system problems from a holistic perspective. Understand and evaluate the theory behind systems thinking and Checkland's theories ◆ Analysing problems using systems thinking ◆ Critically evaluate socio-technical systems methodologies ◆ Compare and contrast the frameworks and methods used within the field ◆ Propose and reflect on solutions to problems. ◆ Determine the effectiveness of conceptual models in capturing reality ◆ Examine the rise of people centred solutions in the modern organisation 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Identify stakeholders and their interests in solving a problem and evaluating methods and frameworks for solving problems ◆ Employ a range of modelling techniques to capture and communicate key aspects of a system ◆ Inform and guide the solution of problems within and improvements to systems ◆ Understanding the people / cultural aspects of the Information Systems field ◆ Ability to use directed reading, and critically evaluate articles and develop learning through case studies ◆ Report writing and demonstrating argument development ◆ Use of VLE as a means of learning, contributing and discussing 	
Assessment Methods:	Assessment: Examination: (weighting – 50%) Coursework: (weighting – 50%)	Re-assessment: Examination: (weighting – 100%)

BSc Computer Systems

Course Descriptors

Year 4, Semester 1

More information about the Dissertation can be found on Vision under F20PA

Not all final year optional courses may run in a given year

Course Code: F20PA	Course Title: Research Methods & Requirements Engineering	Course Co-ordinator: Alasdair Gray
Pre-requisites:	None	
Aims:	Development of project research method and requirement analysis skills.	
Syllabus:	Requirements analysis of software development project Researching current state of art in this area Library resources and their use, Web and online database searching	
Learning Outcomes: Subject Mastery	<i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i> <ul style="list-style-type: none"> ◆ Understanding of research or development based problem related to a substantial software development topic ◆ Requirements specification and background research skills for it ◆ Ability to plan a significant project of research, investigation or development 	
Learning Outcomes: Personal Abilities	<i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i> <ul style="list-style-type: none"> ◆ Ability to research and undertake critical review and evaluation of data and supplied literature ◆ Project planning skills ◆ Written communication skills ◆ Time management 	
Assessment Methods:	Assessment: Coursework: (weighting – 100%) Synoptic with F20PB & F20PC	Re-assessment: None

Course Code: F20CL	Course Title: Computing in the Classroom	Course Co-ordinator: Tessa Berg/Fiona McNeill
Pre-requisites:		
Aims:	<p>This course runs under the Undergraduate Ambassadors Scheme (http://www.uas.ac.uk/) and provides an opportunity for students to act as ambassadors for their disciplines.</p> <p>Aims are:</p> <ul style="list-style-type: none"> ◆ To develop a range of skills in the student and to offer an early taste of teaching to those interested in pursuing it as a career. ◆ To help students gain confidence in communicating their subject, develop strong organisational and interpersonal skills, and understand how to address the needs of individuals. ◆ To learn to devise and develop projects and teaching methods appropriate to engage the relevant age group they are working with. ◆ To help inspire a new generation of prospective undergraduates by providing role models for school pupils. ◆ To help teachers convey the excitement of their subject to their pupils by showing them the long-term applications of school studies. ◆ To help teachers by providing an assistant who can work with and support pupils at any point on the ability spectrum. 	
Syllabus:	<p>This format is standard within the Undergraduate Ambassadors :</p> <ul style="list-style-type: none"> ◆ Initial training to provide the student with an introduction to working with children and conduct in the school environment ◆ Competitive interview system to ensure students' suitability for the course. ◆ Student will be matched with an appropriate school and a specific teacher in the local area. ◆ Student will spend the equivalent of half a day a week in the school every week for a semester. ◆ No formal lectures. ◆ A supporting tutorial for one hour once a week for students to share experiences. ◆ Initial contact with the teacher and pupils will be as a classroom assistant - observing the teacher and offering practical support. ◆ The teacher will assign the student teaching tasks which will vary depending on the teacher's needs and the ability of the student. ◆ The student will design and deliver a special project on the basis of discussion with the teacher and their own assessment of what will interest the particular pupils they are working with. ◆ End of unit presentation of special project 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Communicate and present computing topics to others. ◆ Develop a better understanding of, appreciation of, and confidence in computing through teaching it to others. ◆ Gain a broad understanding of many of the key aspects of teaching computing in schools 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <p>To experience working in a challenging and unpredictable environment. Be able to assess and devise appropriate ways to communicate a difficult principle or concept</p>	

	<p>To master the following specific and transferable skills:</p> <ul style="list-style-type: none"> - Understanding the needs of individuals. - Employ effective interpersonal skills when dealing with colleagues. - To understand and support teaching staff responsibilities and conduct oneself accordingly. - To be able to improvise and adapt to the responses of students in a classroom. - To give (and take) feedback. - To make effective use of organisational, prioritisation and negotiating skills. - To handle difficult and potentially disruptive situations. - To be able to use public speaking and communication skills, both one-to-one and when speaking to an audience - Team-working. - To understand and be able to make use of standard teaching methods. - To prepare, use and reflect on the effectiveness of lesson plans and teaching materials. <p>To be able to reflect on the effectiveness of methods of teaching and learning both personally and in review with tutors.</p>
Assessment Methods:	<p>Assessment:</p> <p>Coursework (weighting – 100%)</p> <p>Course will be spread over 2 semesters so students will have to work more on their project in Semester 1 to compensate for spending some time on it in Semester 2 (equivalent to half course)</p> <p>Re-assessment:</p> <p>None</p>

Course Code: F20CN	Course Title: Computer Network Security	Course Co-ordinator: Hamish Taylor/Mike Just
Pre-requisites:	F29DC Data Communications and Networking and F27SG Software Development or their equivalent	
Aims:	<ul style="list-style-type: none"> ◆ Impart specialist understanding of key concepts, issues, theories, principles and techniques of computer network security. ◆ Provide detailed understanding and practical experience with key services and tools used for computer network security purposes. ◆ Give practical experience of analysing requirements, designing, implementing and validating security solutions for computer network applications. 	
Syllabus:	<ul style="list-style-type: none"> ◆ Security concepts and definitions, basics of cryptography (concepts, definitions, steganography), symmetric cryptography (historical, hash functions, MACs, block and stream encryption), asymmetric cryptography (basic number theory, RSA, DH, digital signatures), cryptographic key management, operating system security (concepts, memory management, buffer overflow, race conditions, file/disk encryption), security-enhanced Linux, authentication & access control (biometrics, passwords, role- and capability-based), as well as some Linux-based security tools (e.g., GPG, openssl). ◆ Computer network security concepts, X.800 model - attacks, mechanisms, services. Computer network layers, reference model, TCP/IP and its vulnerabilities. Network service fundamentals - sockets, services, threads and their protection. Digests – MD5, symmetric ciphers, JCE. Digital signatures, public key certificates. X.509 certificates, certificate authorities and hierarchical trust models. Secure key exchange – Diffie Hellman, SSL/TLS, SSH. Applets, Java security model and tools, signing applets with CA keys. PGP public keys, OpenPGP, web of trust, network security tools - VPNs, Firewalls, Intrusion Detection Systems, malware scanners. 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Detailed and reflective understanding of the concepts, issues, principles and theories of computer network security ◆ Theoretical and practical knowledge of a range of computer network security technologies as well as network security tools and services ◆ Practical experience of analysing, designing, implementing and validating solutions to computer network security challenges using common network security tools. 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Ability to deal with complex issues and make informed judgements about network security in the absence of complete or consistent data. ◆ Exercise autonomy and initiative in addressing computer network security challenges. ◆ Showing initiative and team working skills in shared computer network security application development. (PDP) ◆ Demonstrate reflection on network security issues. (PDP) 	
Assessment Methods:	Assessment: Examination: (Weighting 60%) Coursework: (weighting – 40%)	Re-assessment: None

Course Code: F20IF	Course Title: Information Systems Methodologies	Course Co-ordinator: Jenny Coady
Pre-requisites:	F29SO – Software Engineering & F29PD – Professional Development <i>or equivalent</i>	
Aims:	This course explores a range of issues concerning advanced contemporary methodological approaches to information systems development. The aim is to enable students to develop critical faculties and techniques in relation to the selection and application of these methodological approaches.	
Syllabus:	<p>There is a growing requirement in industry for engineers and scientists with good and appropriate analytical skills when considering the development and evolution of systems, in particular information systems. This module develops further the knowledge and skills students should have already gained in the Information Systems and Software Engineering modules in topics such as:</p> <ul style="list-style-type: none"> ◆ General Systems Principles; ◆ Systems Classification and Taxonomy Models; ◆ Information Systems Life Cycle and Functions; ◆ Paradigmatic Approach to Methodology Classification; ◆ Framework for Analysis and Comparison of Methodologies (NIMSAD & Fitzgerald's); ◆ Process Improvement Models. 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <p>This course develops further the knowledge and skills students should have already gained in the Information Systems and Software Engineering courses. It will enable students to:</p> <ul style="list-style-type: none"> ◆ Determine alternative approaches to gathering requirements and systems development ◆ Compare methodologies for use in organisations using a standardised Framework ◆ Rationalise systems development to prepare a more relevant system ◆ Assimilate their knowledge and understanding of the ways in which Information Systems are developed, including a range of established techniques of enquiry or research methodologies 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Critical reading and reviewing works in the field ◆ Evaluating Methods under an agreed Framework ◆ Structuring an argument (PDP) ◆ Use of VLE as a means of learning 	
Assessment Methods:	Assessment: Examination: (weighting – 60%) Coursework: (weighting – 40%)	Re-assessment: None

Course Code: F20MC	Course Title: Mobile Communications and Programming	Course Co-ordinator: Peter King
Pre-requisites:	Knowledge of network communications and object oriented programming	
Aims:	<ul style="list-style-type: none"> ◆ To introduce students to the particular problems of building networks which include mobile computing devices and to explain how they may be overcome using current technology ◆ To introduce students to the issues surrounding ad hoc networking and give an understanding of how these can be addressed ◆ To introduce students to programmable mobile and handheld devices ◆ To develop students' skills in developing applications for mobile and handheld devices 	
Syllabus:	Fixed node IP routing - routing techniques for conventional wired networks Mobile IP routing - routing for wireless mobiles to IP Ad hoc networks and routing Security protocols - identification and authorisation, infra structure security Small device characteristics - screen size, memory, power consumption, input mechanisms Current devices - tablet PC, mobile phone, PDA Application development environments - Java APIs, C# and .NET	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ To understand and apply the principles of secure, effective communication over networks including mobile elements. ◆ To be able to explain the operation of current and proposed protocols for communication over networks which include mobile elements ◆ To understand the issues introduced by ad-hoc networking. ◆ To have knowledge of common ad-hoc routing protocols ◆ To explain evaluate current and proposed mobile devices ◆ To design applications for mobile devices including use of wireless communications where appropriate. ◆ To program such applications using current application development environments 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ To be able to select and apply suitable techniques of analysis in assessing the effectiveness of a technical solution ◆ To be able to critically review the issues of security and privacy relating to networking ◆ To be able to write good technical documents in support of problem solving within the domains of mobile networking and of mobile and handheld device solutions. 	
Assessment Methods:	Assessment: Examination: (weighting– 80%) Coursework: (weighting – 20%)	Re-assessment: None

Course Code: F20RS	Course Title: Rigorous Methods for Software Engineering	Course Co-ordinator: Andrew Ireland/Lilia Georgieva
Pre-requisites:	F28SD Software Design <i>or equivalent.</i>	
Aims:	To provide knowledge and understanding of tools and techniques which support rigorous software engineering	
Syllabus:	<ul style="list-style-type: none"> ◆ The course addresses the challenges of engineering and modelling safe and secure software systems. ◆ It covers a range of rigorous processes and formal methods that support the development of high integrity software systems. ◆ From modelling and reasoning about designs through to code, students will experience a range of state-of-the-art static analysis tools and techniques. ◆ While theory based, the course has a strong practical element, drawing upon industrial case study material where appropriate. 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Knowledge and understanding of a range of rigorous processes and formal methods that support the development of high integrity software systems. ◆ Critical understanding of the relationship between code level annotations and high-level formal specifications. ◆ Knowledge of the mechanisms that underlie advanced modelling and static analysis techniques. ◆ To be able to demonstrate a critical understanding of the relationship between code level annotations and flow analysis techniques. ◆ To be able to demonstrate a critical understanding of program proof and how it can be used to provide strong formal correctness guarantees. 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Conceptualize and define new abstract problems within the context of automated software development. ◆ Make informed judgements in situations in the absence of complete or consistent data. ◆ Exercise autonomy, initiative and creativity in the application of software engineering techniques. ◆ Demonstrate critical reflection. (PDP) ◆ Communicate with professional level peers, senior colleagues and specialists. (PDP) 	
Assessment Methods:	<p>Assessment:</p> <p>Examination: (weighting– 60%)</p> <p>Coursework: (weighting – 40%)</p>	<p>Re-assessment:</p> <p>None</p>

Course Code: F20SC	Course Title: Industrial Programming	Course Co-ordinator: Hans-Wolfgang Loidl
Pre-requisites:	Programming skills in a language such as C or Java	
Aims:	<ul style="list-style-type: none"> ◆ To develop proficiency in contemporary industrial programming languages and platforms ◆ To enable the elaboration and combination of system components in different languages ◆ To enable an agile and flexible response to changes in industrial practices ◆ To enable participation by industrial practitioners to provide context and applicability 	
Syllabus:	<ul style="list-style-type: none"> ◆ Programming in a modern general purpose language e.g. C#, C++11 ◆ Programming for concurrency using state-of-the-art libraries and language extensions ◆ Rapid prototyping in a major scripting language with associated libraries and frameworks e.g. Python, PHP, Ruby, Lua ◆ Coverage of advanced language features where languages have been met in earlier courses ◆ Foresight of emerging programming language technologies ◆ Practical experience with standard environments (Unix, Windows), virtual machines (.NET) and tools (e.g. compilers, debuggers, libraries, shell) 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Basic appreciation of role of different programming paradigms in programming/managing systems ◆ Understanding of core characteristics of contemporary operating systems ◆ Knowledge of key abstractions across programming languages ◆ Technical proficiency in advanced language techniques in different programming paradigms 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Ability to choose/deploy/combine appropriate languages, architectures and tools ◆ Ability to employ an agile approach to software development 	
Assessment Methods:	Assessment: Coursework: (weighting – 100%)	Re-assessment: None

Course Code: F20GA	Course Title: 3D Graphics & Animation	Course Co-ordinator: Stefano Padilla
Pre-requisites:	Software Development 3 (F27SG) or equivalent (object-orientated programming).	
Aims:	Investigate Computer Graphics theory and develop programming skills in 2D/3D Graphics and Animation.	
Syllabus:	<ul style="list-style-type: none"> ◆ Computer Graphics concepts and practical introduction to Graphics Programming. ◆ Vertices, triangles, meshes, display lists and models. ◆ Hierarchical modelling and scene graphs understanding and representation. ◆ 2D and 3D transformations, homogeneous co-ordinates, matrices multiplication. ◆ Model, world, camera, scene and projection spaces. ◆ Instancing and tessellation. ◆ Materials, texture mapping and shading. ◆ Lighting and global illumination models. ◆ Shadows, occlusions and reflections. ◆ Real-time and offline rendering pipelines. ◆ Basics of procedural and physical animations. ◆ Animation systems and concepts. ◆ Animation skeletons, poses, clips, skinning and blending. ◆ Tools, environments, coding practices and industrial applications. ◆ Course summary and review. 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Understanding and categorising Computer Graphics concepts. ◆ Ability to identify core 3D elements and structures. ◆ Investigation of scene graphs, hierarchical models, and spaces. ◆ Understanding and research into transformations, modelling, and projection concepts. ◆ Ability to contrast different material, lighting and shadowing models. ◆ Develop and implement animation concepts and systems. ◆ Ability to design, understand and implement a small-scale rendering and animation system. 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Ability to think, plan, and construct in three dimensions. ◆ Representation of, planning for, and solution of problems. ◆ Ability to plan, design and implement a rendering and animation system. ◆ Skills integrating graphics and animation in various industries. 	
Assessment Methods:	Assessment: Class Test: (weighting – 40%) Coursework: (weighting – 60%)	Re-assessment: None

BSc Computer Systems

Course Descriptors

Year 4, Semester 2

Course Code: F20PB	Course Title: Project: Design & Implementation	Course Co-ordinator: Alasdair Gray
Pre-requisites:	None	
Aims:	Development of project design and implementation skills	
Syllabus:	<ul style="list-style-type: none"> ◆ Software and/or experimental design and its documentation ◆ Relevant commercial practice in applied design of software 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Software design and implementation skills 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Time management ◆ Project Management 	
Assessment Methods:	Assessment: Coursework (weighting – 100%) Synoptic with F20PA & F20PC	Re-assessment: None

Course Code: F20PC	Course Title: Project: Testing & Presentation	Course Co-ordinator: Alasdair Gray
Pre-requisites:	None	
Aims:	Development of knowledge and skills for testing and evaluating a software project	
Syllabus:	<ul style="list-style-type: none"> ◆ Testing of Software ◆ Evaluation of Software ◆ Report Writing 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Testing and evaluation of software development projects ◆ Documenting Software projects 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Awareness and experience of methods and tools for validation and verification in professional practice ◆ Practical skills in testing and evaluation ◆ Documentation skills 	
Assessment Methods:	Assessment: Coursework (weighting – 100%) Synoptic with F20PA & F20PB	Re-assessment: None

Course Code: F20AD	Course Title: Advanced Interaction Design	Course Co-ordinators: Lynn Baillie/tbc
Pre-requisites:	<i>F28IN Interaction Design or equivalent</i>	
Aims:	<p>The course aims to give students the opportunity to develop:</p> <ul style="list-style-type: none"> ◆ A detailed and critical knowledge of requirements gathering, design and evaluation techniques in interaction design. ◆ An awareness of current research and emerging issues in the field of interaction design. ◆ A range of specialised skills, and research methods involved in working with users. 	
Syllabus:	<p>Current and emerging topics in Interaction Design including: user demographics, patterns in technology adoption, interaction design lifecycles, user interface design patterns, prototyping methods, a wide range of qualitative and quantitative data gathering and analysis techniques, accessibility, and a range of research case studies covering cutting edge issues in the field</p>	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <p>Students will develop skills in the following areas:</p> <ul style="list-style-type: none"> ◆ Review, critically analyse, evaluate, and synthesise of previous research projects in the field of interaction design ◆ Identify and propose innovative solutions in response to analysis of users' requirements. ◆ Make informed judgements about appropriate methodologies for developing and evaluating technologies suitable for user demographics and background experience. 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <p>Students will develop skills in the following areas:</p> <ul style="list-style-type: none"> ◆ Use discipline appropriate software for data analysis, prototyping and learning. ◆ Present, analyse and interpret numerical and graphical data gathered as part of evaluation studies. ◆ Communicate effectively to knowledgeable audiences by preparing formal and informal presentations and written reports. ◆ Exercise autonomy and initiative by planning and managing their own work; develop strategies for independently solving problems and taking the initiative. ◆ Take responsibility for their own and other's work by contributing effectively and conscientiously to the work of a group, actively maintaining good working relationships with group members, and leading the direction of the group where appropriate. ◆ Reflect on roles and responsibilities by critically reflecting on their own and others' roles and responsibilities. ◆ Develop an awareness of professional and ethical issues including working with human subjects and wider issues relating to technology in society 	
Assessment Methods:	Assessment: Exam: (weighting – 50%) Coursework: (weighting – 50%)	Re-assessment: None

Course Code: F20BD	Course Title: Big Data Management	Course Co-ordinator: Albert Burger/Alasdair Gray
Pre-requisites:	Academic knowledge of fundamentals of databases and logic.	
Aims:	<ul style="list-style-type: none"> ◆ Review principle abstractions, methods and techniques for the management of large and complex data sets ("Big Data"). ◆ Develop an understanding of the foundations and tools of the Semantic Web. ◆ Enable students to appreciate critically a range of data integration solutions. 	
Syllabus:	<p>Complex data sets: RDF, triple stores, SPARQL, Big Data vs Smart Data vs Broad Data, NoSQL, indexing data.</p> <p>Semantic Web Foundations: RDFS, OWL, Ontologies, Reasoning, Protégé.</p> <p>Data Integration: Linked Data, Mash-ups, Ontology mapping, Data Provenance.</p>	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Knowledge and understanding of a range of data representation and data management techniques for big data sets. ◆ Critical understanding of the role of semantic web technologies in the context of big data management. ◆ Knowledge of the mechanisms that underlie data integration techniques. ◆ To be able to demonstrate a critical understanding of appropriateness and effectiveness of different techniques. 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Conceptualize and define new abstract problems within the context of complex data sets. ◆ Make informed judgements about the applicability of semantic web solutions to big data questions. ◆ Exercise autonomy, initiative and creativity in the application of data integration techniques. ◆ Demonstrate critical reflection. (PDP) ◆ Communicate with professional level peers, senior colleagues and specialists. (PDP) 	
Assessment Methods:	Assessment: Examination: (weighting – 70%) Coursework: (weighting – 30%)	Re-assessment: None

Course Code: F20DE	Course Title: Digital Knowledge Economy	Course Co-ordinator: Jessica Chen-Burger
Pre-requisites:	Fundamentals of logic, grasp of computational thinking	
Aims:	<ul style="list-style-type: none"> ◆ To provide an overview of advanced topics in Digital and Knowledge Economy, including current developments and future trends in developed economies resulting from deploying new technologies and utilising emerging knowledge. ◆ To discuss e-Business, as a new breed of modern business model that leverages technical advancements to create economic growth. ◆ To provide a high level description of business and technological issues related to Digital and Knowledge Economy. ◆ To introduce technologies and methodologies so as to provide a deep understanding of the Digital and Knowledge Economy, including business, organisational, knowledge and technology based issues. ◆ To impart rigorous technical modelling and analytical methodologies for working with complex problems in this area. ◆ To facilitate the dialogue between business and computing personnel, and translate business requirements to computing ones and vice versa. ◆ To impart deep understanding of the motivation and rationale behind the conversations between business and IT, as well as other relevant technologies and future trends - so that students can recommend them and/or participate in the decision making process for future planning. 	
Syllabus:	<ul style="list-style-type: none"> ◆ Introduction to Digital and Knowledge Economy <ul style="list-style-type: none"> ○ Introduction to Digital and Knowledge Economy ○ Its relevance to e-Business ◆ Topics in Digital Economy <ul style="list-style-type: none"> ○ An overview of technologies and tools for e-Business ○ What is a business model? What are the different types of business models? ○ What are the relationships between business models and innovative/disruptive technologies? ○ Current development and future trends in Digital and Knowledge Economy ○ Relevant technology offerings, e.g. Bitcoin, IBM's cloud computing platform ◆ Knowledge based technologies in Knowledge Economy <ul style="list-style-type: none"> ○ introduction to knowledge management, knowledge modelling technologies, including ontologies ○ Introduction to logic, Intelligent Systems and related technologies, including semantic web based technologies ○ Case studies of Intelligent Systems and Future trends ◆ Supply Chain Management and its relation to Digital Economy <ul style="list-style-type: none"> ○ What is SCM? What are the standard practices in SCM, e.g. SCOR? ○ Introduction to process modelling, business operations and SCM. ○ What is global SCM? Case studies, e.g. IKEA's global SCM; Current and future trends ◆ Business Intelligence: Fundamentals issues and technologies 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Understanding of key issues in Digital and Knowledge Economy. ◆ Understanding of ontologies, conceptual and knowledge modelling technologies, in terms of design, critical evaluation and suitable practical uses. ◆ Understanding of issues in intelligent systems, supply chain management and business intelligence and the roles technologies may play. ◆ Understanding of issues and the motivation and rationale of business and technical problems in Digital and Knowledge Economy. 	

	<ul style="list-style-type: none"> ◆ Ability to select and construct conceptual models, including ontologies, and can create appropriate evaluation criteria to assess them. ◆ Ability to take self-initiatives to review relevant literature independently in Digital and Knowledge Economy. 	
Learning Outcomes: Personal Abilities	<i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i> <ul style="list-style-type: none"> ◆ Analytical skills in conceptual modelling methods, including ontologies, process and knowledge modelling, for business problems. ◆ Ability to make well-informed evidence-based arguments towards supporting or rejecting technologies to solve business problems. ◆ Ability to deal with complex issues and make informed judgements, e.g. about ontologies, knowledge modelling, intelligent and business systems in the absence of complete or consistent data. ◆ Exercise autonomy and initiative in addressing digital and knowledge economy challenges. ◆ Demonstrate reflection on digital and knowledge economy. ◆ Ability to judge technology hypes and develop personal opinions on future trends. 	
Assessment Methods:	Assessment: Exam: (weighting – 70%) Coursework: (weighting – 30%)	Re-assessment: None

Course Code: F20EC	Course Title: E-Commerce Technology	Course Co-ordinator: Santiago Chumbe/ Jessica Chen-Burger
Pre-requisites:	None	
Aims:	<ul style="list-style-type: none"> ◆ To impart an understanding of e-Commerce technology and of how Information Systems play a fundamental role in shaping e-Commerce; ◆ To put e-Commerce technologies in a structural framework, show how they support e-Commerce operations, provide technical know-how for implementing e-Commerce platforms and analytical skills to examine the technical aspects of e-Commerce; ◆ To show how the B2B, B2C, C2B, C2C and P2P types of e-Commerce work; ◆ To introduce practical aspects in implementing and managing e-commerce websites; ◆ To provide a description of technological challenges and innovations in e-Commerce; ◆ To impart an understanding of the integration and the interoperability aspects of e-Commerce in the whole business system. 	
Syllabus:	<ul style="list-style-type: none"> ◆ Introduction to e-Commerce and overview of its technology ◆ Web related e-Commerce technologies: <ul style="list-style-type: none"> ○ E-Commerce workflow and transactions ○ Web architecture and E-Commerce development challenges ○ Development of web applications (agile development, MVC pattern and web frameworks) ○ Service Oriented Architecture (SOA), web services and cloud computing ○ Databases in E-Commerce: SQL and NoSQL alternatives ○ Data mining, augmented reality and product ontologies (GoodRelations) ◆ Implementation and management of e-Commerce technologies <ul style="list-style-type: none"> ○ Security in E-Commerce: External and internal attacks and, data protection at rest and in transit ○ Mobile commerce ○ E-Commerce business, marketing and revenue models ○ Recommender systems and personalization ○ Cloud computing deployment models for e-Commerce ○ E-Commerce interoperability processes as part of a business system ○ Merchant services and payment transactions 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Recognise e-Commerce as a special type of business; while at the same time can relate its underlying business logic to that of conventional business operations, e.g. in different aspects of business systems ◆ Has in-depth knowledge and understanding of the different technologies that play crucial roles in enabling and enhancing business systems to meet e-Commerce requirements ◆ Critical understanding of the evolving methodological issues pertaining to e-Commerce system development ◆ In-depth understanding of Web technologies that enable e-Commerce applications ◆ Insights into real-world problems that are encountered when developing e-Commerce applications ◆ Analysis and development skills and knowledge of web technologies to create e-Commerce sites. 	

Learning Outcomes:: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Can deal with complex issues and make informed judgements about the use of Web based Technologies in e-Commerce ◆ Exercise substantial autonomy, initiative and creativity in the application of Web-based tools and services for the development of e-Commerce applications ◆ Can communicate with peers, more senior colleagues and specialists and demonstrate critical reflection (PDP) ◆ Can independently carry out research based on literatures in the field of e-Commerce Technology ◆ Can evaluate e-Commerce technologies and propose solutions to business scenarios based on a case study 	
Assessment Methods:	Assessment: Examination: (weighting – 60%) Coursework: (weighting – 40%)	Re-assessment: None

Course Code: F20GP	Course Title: Computer Games Programming	Course Co-ordinator: Ruth Aylett/Stefano Padilla
Pre-requisites:	C++ programming skills	
Aims:	To develop programming skills and techniques specific to the area of 2D and 3D computer games	
Syllabus:	<ul style="list-style-type: none"> ◆ Computer Games Design Concepts (Genres, Narrative and Fun). ◆ Elements of Game Design (Formal, Dramatic and System Dynamics). ◆ Character and World Design. ◆ Design Programming Patterns (Input, loops, structures, objects and optimisation). ◆ Games Creation Concepts (Conceptualisation, Prototyping, Playtesting). ◆ Game-state, simulator, renderer, (hierarchical) controllers. ◆ Tools, environments and coding practices – e.g. graphics, C++ and engines. ◆ 2D and 3D game programming techniques. ◆ Physically-based modelling, particle systems, flocking. ◆ Obstacle avoidance and path planning. ◆ Group movement. ◆ Learning and adaptation in games. ◆ Action and behaviour selection. ◆ Procedural Generation. ◆ Course summary and review. 	
Learning Outcomes: Subject Mastery	<p><i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i></p> <ul style="list-style-type: none"> ◆ Understanding of game theory and computer game history, genres and impact ◆ Understanding of game design concepts, elements and characters. ◆ Understanding of available tools and their application ◆ Knowledge of algorithms for path planning and navigation ◆ Knowledge of physically-based modelling in games and selection of techniques ◆ Knowledge of AI techniques in games and selection of techniques ◆ Ability to understand, design and implement a small-scale game using 2D and 3D tools ◆ Practical skills in graphics and AI programming in the computer games context 	
Learning Outcomes: Personal Abilities	<p><i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i></p> <ul style="list-style-type: none"> ◆ Representation of, planning for, and solution of problems. ◆ Ability to plan, design, prototype and communicate a game. ◆ Ability to think and plan in three dimensions. ◆ Team working skills. 	
Assessment Methods:	Assessment: Exam: (weighting – 50%) Coursework: (weighting – 50%)	Re-assessment: None

Course Code: F20NA	Course Title: Network Applications	Course Co-ordinator: Hamish Taylor
Pre-requisites:	Either F28WP Web Programming, F27SG Software Development 3 and F29DC Data Communications and Networking or reasonable software development skills in Java and knowledge of fundamentals of data communications and the web	
Aims:	<ul style="list-style-type: none"> ◆ To impart a range of knowledge and understanding of the theories, principles and protocols underlying the primary network applications on the Internet ◆ To develop the ability to appreciate professionally the range of network application technologies and standards ◆ To develop skills in a range of the principal network technologies, to impart the main design and practical issues faced in their application, and confer the ability to select and apply relevant techniques for a given network application development problem. ◆ To have students creatively develop in teams a substantial network application involving web and application server technologies to an original design of their own 	
Syllabus:	Network services – service styles and models, Internet, DNS, sockets, implementing services; e-mail - MIME, SMTP, POP, IMAP; web protocols - URIs, HTTP versions and characteristics; web content - HTML, XML, XHTML, HTML 5, forms, tables, embedded objects; CSS style properties; JavaScript – object prototypes, standalone and web client programming; DOM versions, CSS 3, DHTML, AngularJS; web server programming in Java and JavaScript - REST web services, CGI, servlets, JSP, Node.js, web frameworks; asynchronous use of HTTP - AJAX, JSON, JSONP; textual conferencing - IM, IRC, web chat via short and long polling, HTTP streaming, applet sockets and web sockets; web sessions – URL rewriting, web storage, cookies; HTTP authentication.	
Learning Outcomes: Subject Mastery	<i>Understanding, Knowledge and Cognitive Skills; Scholarship, Enquiry and Research (Research-Informed Learning)</i> <ul style="list-style-type: none"> ◆ Detailed and specialist knowledge and understanding of the theories, techniques and principles underlying the design of network applications and the range of their application ◆ Theoretical and practical knowledge of the major network application types including email, web and chat applications and services ◆ Reflective awareness of protocols and standards underlying key network applications especially the web and of enabling technologies for network applications such as sockets, DNS, XML, DOM ◆ Ability to design and develop useful network applications including web, email and chat software using apt technologies and languages: HTML, XML, JavaScript, CSS, CGI, servlets, JSP, Node.js, REST web services, web sockets etc. 	
Learning Outcomes: Personal Abilities	<i>Industrial, Commercial & Professional Practice; Autonomy, Accountability & Working with Others; Communication, Numeracy & ICT</i> <ul style="list-style-type: none"> ◆ Skills in selecting, applying and evaluating apt technologies in a professional way given a problem requiring network interaction ◆ Ability to build on initial skills and knowledge by independent research using online resources ◆ Showing initiative, creativity and team working skills in shared network application development 	
Assessment Methods:	Assessment: Examination: (weighting – 60%) Coursework: (weighting – 40%)	Re-assessment: None