

University of Dublin
Trinity College

School of Engineering
and
School of Computer Science and
Statistics

BAI Electronic Engineering (Stream C)
BAI Computer Engineering (Stream D)
BAI Electronic and Computer Engineering
(Stream CD)

Senior Sophister Handbook

2011-2012

Introduction

Welcome to the Senior Sophister specializations: Electronic, Electronic and Computer Engineering and Computer Engineering. As you will know by now, these are referred to as Stream C, Stream CD and Stream D respectively. The objective of the BAI degree offered by the Departments of Electronic and Electrical Engineering and Computer Science is to produce well-rounded graduates, having a strong grounding in analytical skills and flexibility to adapt to the advances in electronic technology, computer systems and communications systems. The Course organization and Module descriptions included in this Handbook have been checked but may be subject to change.

Academic Structure

The SS year is broken into two twelve-week semesters:

UNIVERSITY OF DUBLIN

ACADEMIC YEAR STRUCTURE
2011 / 2012

TRINITY COLLEGE

Cal. Wk	Dates 2011/12 (week beginning)	Outline Structure of Academic Year 2011/12	Notes
1	29-Aug-11		
2	05-Sep-11	Supplemental Examinations	Statutory Term (Michaelmas) begins
3	12-Sep-11	PG Registration	
4	19-Sep-11	UG New Entrant Registration/Freshers' Week	
5	26-Sep-11	Teaching Week 1 Registration continuing students	Michaelmas Lecture term begins
6	03-Oct-11	Teaching Week 2 Registration continuing students	
7	10-Oct-11	Teaching Week 3 Registration continuing students	
8	17-Oct-11	Teaching Week 4	
9	24-Oct-11	Teaching Week 5	
10	31-Oct-11	Teaching Week 6 (Monday, Public Holiday)	
11	07-Nov-11	Teaching Week 7 - Study Week	
12	14-Nov-11	Teaching Week 8	
13	21-Nov-11	Teaching Week 9	
14	28-Nov-11	Teaching Week 10	
15	05-Dec-11	Teaching Week 11	
16	12-Dec-11	Teaching Week 12	←Michaelmas term ends Friday 16 December 2011
17	19-Dec-11		
18	26-Dec-11	Christmas Period (College closed from 23 December 2011 to 3 January 2012)	
19	02-Jan-12		
20	09-Jan-12	Foundation Scholarship Examinations	Note: it may be necessary to hold some exams in the preceding week.
21	16-Jan-12	Teaching Week 1	Hilary Term begins
22	23-Jan-12	Teaching Week 2	
23	30-Jan-12	Teaching Week 3	
24	06-Feb-12	Teaching Week 4	
25	13-Feb-12	Teaching Week 5	
26	20-Feb-12	Teaching Week 6	
27	27-Feb-12	Teaching Week 7 - Study Week	
28	05-Mar-12	Teaching Week 8	
29	12-Mar-12	Teaching Week 9 (Saturday, St Patrick's Day)	
30	19-Mar-12	Teaching Week 10 (Monday, Public Holiday)	
31	26-Mar-12	Teaching Week 11	
32	02-Apr-12	Teaching Week 12 (Friday, Good Friday)	←Hilary Term ends Friday 6 April 2012. Good Friday
33	09-Apr-12	Revision (Monday, Easter Monday)	Trinity Term begins. Easter Monday 9 April 2012
34	16-Apr-12	Revision Trinity Week (Monday, Trinity Monday)	
35	23-Apr-12	Revision	
36	30-Apr-12	Annual Examinations 1	Annual Examination period: 4 weeks at present followed by 4 weeks for marking, examiners' meetings, publication of results, Courts of First Appeal and Academic Appeals.
37	07-May-12	Annual Examinations 2 (Monday, Public Holiday)	
38	14-May-12	Annual Examinations 3	
39	21-May-12	Annual Examinations 4	
40	28-May-12	Marking/Courts of Examiners/Results	
41	04-Jun-12	Marking/Courts of Examiners/Results (Monday, Public Holiday)	
42	11-Jun-12	Marking/Courts of Examiners/Results	
43	18-Jun-12	Marking/Courts of Examiners/Results/Courts of First Appeal	
44	25-Jun-12	Courts of First Appeal/Academic Appeals	
45 to 52	02-Jul-12 to 20-Aug-12	Postgraduate dissertations/theses /Research 1-8	Eight weeks between end of statutory (Trinity) term and commencement of statutory (Michaelmas) term. This period is also used for writing up Masters dissertations and research theses due for submission in September. ← Ends Friday 24 August 2012

Courses

Courses undertaken by the Senior Sophister students are:

Faculty Courses

4E1	Management for Engineers
4E2	Final Year Project

Semester

I
I & II

EEE Dept Courses

4C1	Integrated Systems Design
4C2	Microelectronic Circuits
4C3	Digital Control Systems
4C4	Telecommunications
4C5	Digital Signal Processing
4C6	Microelectronic Technology
4C7	Digital Communications
4C8	Digital Media Processing

Semester

I
II
II
I
I
I
II
II

CS Dept Courses

4D1a	Distributed Systems
4D1b	Advanced Microprocessor Systems
4D2a	Data Engineering
4D2b	Knowledge Engineering
4D3a	Computer Graphics I
4D3b	Computer Graphics II
4D4a	Computer Vision I
4D4b	Computer Vision II

Semester

I
II
I
II
I
II
I
II

Semester 1

Module Title	ECTS	Code
Mgt for Engineers	5	4E1
Individual Project	5	4E2a
Microelectronic Technology	5	4C6
Digital Signal Processing	5	4C5
Telecommunications	5	4C4
Integrated Systems Design	5	4C1
Distributed Systems	5	4D1a
Data Engineering	5	4D2a
Computer Graphics I	5	4D3a
Computer Vision I	5	4D4a

Semester 2

Module Title	ECTS	Code
Individual Project	10	4E2b
Microelectronic Circuits	5	4C2
Digital Media Processing	5	4C8
Digital Communications	5	4C7
Digital Control Systems	5	4C3
Adv. Microprocessor Systems	5	4D1b
Knowledge Engineering	5	4D2b
Computer Graphics II	5	4D3b
Computer Vision II	5	4D4b

C Stream		CD Stream				D Stream	
4E1 4E2		Students must do 4E1 and 4E2 totaling 20 ECTS and a total of 40 ECTS from the following list including a minimum of 10 ECTS from each Department with a distribution of 30 ECTS per semester				4E1 4E2	
EEE Subjects		EEE Subjects		CS Subjects		CS Subjects	
Sem 1	Sem 2	Sem 1	Sem 2	Sem 1	Sem 2	Sem 1	Sem 2
4C6	4C2	4C6	4C2	4D1a	4D1b	4D1a	4D1b
4C5	4C8	4C5	4C8	4D2a	4D2b	4D2a	4D2b
4C4	4C7	4C4	4C7	4D3a	4D3b	4D3a	4D3b
4C1	4C3	4C1	4C3	4D4a	4D4b	4D4a	4D4b

Collaboration and Individual Work

Engineering is about co-operation, but also individual effort. The everyday fruits of engineering, such as a jet aircraft or a suspension bridge or a micro chip or a DVD player, have been designed and built by teams of hundreds, even thousands, of engineers working together. These engineers exchange ideas and ultimately co-ordinate their efforts to achieve the overall project goal. However, each component of even the largest project is the result of one individual's engineering skill and imagination.

If you want to become a successful engineer, you must develop your own ability to analyse problems. This means that, while it is useful to work as a team initially, you must ultimately produce your own work. For example, in the case of a computing exercise, discuss the task with your classmates, swap ideas on how to solve the problem, but at the end of the day, implement your own solution. The examinations will test your ability rather than just your knowledge and the only way to develop your ability for engineering analysis is to complete the laboratory and tutorial exercises yourself.

In the academic world, the principal currency is *ideas*. As a consequence, you can see that *plagiarism* – i.e. passing off other people's ideas as your own – is tantamount to theft.

The College's policy on plagiarism is set out in the College's *General Regulations and Information* booklet, or Section H of the College Calendar.

BAI Examination Rules

Candidates undertake 60 credits during the Senior Sophister year of the degree programme. Each module has an individual rating of 5, 10 or 15 credits, the amount being dependent on the level of effort involved. It is the responsibility of each student to ensure that they are taking courses with ratings amounting to 60 credits. The weighted average mark achieved for the Senior Sophister year is calculated using these credit ratings.

Degrees in all streams are awarded based on an overall average mark calculated by combining the average mark achieved in the ANNUAL Junior Sophister examinations (20% towards overall average) and the ANNUAL Senior Sophister examinations (80% towards overall average).

Students who are deemed to have passed the year are awarded 60 credits and an Honors Degree with the grade based on the overall average mark achieved. In order to pass, students must:

- have achieved marks of at least 40% in individual Senior Sophister courses worth at least 50 ECTS credits **AND**
- have an overall average mark of at least 40%.

A pass degree is awarded in the case of students who fail the B.A.I. degree examinations at the first sitting but pass when they present themselves for re-examination. Students presenting for a Pass degree are assessed on their performance in the Senior Sophister year only.

The full set of overall grades is set out below;

<i>Description</i>	<i>Grade</i>	<i>Criterion</i>
First Class Honors	I	mark greater than or equal to 70%
Second Class Honors,	II.1	mark greater than or equal to 60% and less than 70%

First Division		
Second Class Honors,	II.2	mark greater than or equal to 50% and less than 60%
Second Division		
Third Class Honors	III	mark greater than or equal to 40% and less than 50%
Fail	F	the candidate has failed to satisfy the criteria listed above
Exclude	EX	the candidate has not made a serious attempt at the examinations <u>or</u> the candidate has not passed the year within eighteen months from that date on which they first became eligible <u>or</u> the candidate has at least one unexplained absence
Deferred	D	the candidate was absent with permission due to medical or other grounds and the result is incomplete
ERASMUS Awaiting Result	ER	Applies to International Exchange students, (eg. Unitech / CLUSTER)
Result Withheld	RW	it may be necessary for academic or administrative reasons to withhold a result (e.g. unpaid fees or fines)
Withdrawn	WD	the candidate has withdrawn from the course

For students entering College before the 2008/2009 academic year, a Pass degree may be awarded.

Individual module results

All individual module results are published anonymously by student number on the College notice boards, on the local School of Engineering website - <http://www.tcd.ie/Engineering/Results/> (students will need their College username and password) and on the College's Examinations Office website - <http://www.tcd.ie/Examinations/Results/>. Class rankings are also published on the Engineering notice boards.

Where a mark is not reported for a subject the following codes apply:

f	=	mark is less than 25%;
a	=	absent with permission – may take a SUPPLEMENTAL examination;
A	=	absent without permission or explanation – automatic exclusion;
mc	=	medical certificate supplied to and accepted by the Senior Lecturer;
cr	=	credit for subject;
gw	=	grade withheld (e.g. unpaid fees or fines).

Senior Sophister Fourth Year Courses

4C1 – Integrated Systems Design

COURSE TITLE: Integrated Systems Design		CODE: 4C1
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES: 3C7
LECTURER(S): Dr. Séamas McGettrick	TEACHING ASSISTANT:	TBA
TERMS: Semester 1	LECTURES/WEEK: 2	PRACTICALS/WEEK: 2
DURATION (WEEKS): 12	TOTAL: 22	TOTAL: 22
AIMS/OBJECTIVES <p>This course builds directly on the course 3C7 Digital Systems Design. It will introduce more advanced topics in Verilog-based FPGA design. The approach will be from a systems perspective, looking at the implication of design trade-offs for integrated digital systems typical in DSP and consumer applications.</p>		
SYLLABUS <ul style="list-style-type: none">• Finite state machines with datapath• Verilog HDL language• Automation of testbenches and design of golden vectors• Code coverage• Finite precision effects and choice of bit-width in fixed-point applications• Translating DSP system designed in Matlab onto an FPGA• Memory on FPGAs• Working with FPGA board peripherals• Realisation of all above concepts in hardware designs		
ASSOCIATED LABORATORY/PROJECT PROGRAMME <p>4 * 2 hour labs will be used to support the existing practical element of the course. There will be no extra assignments from these labs sessions.</p>		
RECOMMENDED TEXT(S) <ol style="list-style-type: none">1. Contemporary Logic Design, 2/E, Randy H. Katz, University of California, Berkeley, Gaetano Borriello, University of Washington (MM Mano and MD Ciletti, Digital Design, 4th edition, (Pearson) Prentice Hall, 2007. can be used if it was purchased in 2nd year but be aware that the examples are in VHDL, not in Verilog)2. Verilog HDL, 2/e Palnitkar (reference only)3. FPGA Prototyping By Verilog Examples: Xilinx Spartan-3 Version, Pong P. Chu (wiley) (reference only) <p>Supplementary Reading may be given during course.</p>		

LEARNING OUTCOMES

On completion of this course the student will be able to:

1. Build a synchronous DSP system in verilog and verify its performance
2. Build and test complex FSMs in Verilog
3. Automate testbenches for automatic pass/fail
4. Analyse finite precision effects in filters
5. Make design decisions for fixed point implementations given constraints
6. Analyse memory usage/requirements for FPGA
7. Target sequential designs to FPGA

TEACHING STRATEGIES

This is a highly practical course. There will be 2 “classic” style lectures per week. There will also be a two-hour practical session each week which will be a lecture-come-lab. The FPGA board used to support the practical sessions is the Spartan-3 starter board. The practical sessions will require the students to complete 3 significant assignments outside class hours (average 4-6 hours extra per week), spreading the load through the year. It is critical that the student keeps up with the practical work during the semester.

ASSESSMENT MODE(S)

The written examination will contribute 50% of the overall subject mark at the Annual and Supplemental examinations. The course practical work will contribute 50% of the overall grade.

Attendance at weekly practical sessions is **COMPULSORY**. No marks will be given for the corresponding assignment for unattended practical sessions.

Submission dates will be given for each related assignment.

Late assignments policy:

- Lose half of marks if up to 1 week late
- No marks if over 1 week late

4C2 –Microelectronic Circuits

COURSE TITLE: Microelectronic Circuits		CODE: 4C2
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES: None
LECTURER: Dr. Cédric Assambo	TEACHING ASSISTANT:	
TERMS: Semester 2	LECTURES/WEEK: 3	TOTAL: 33 H
DURATION (WEEKS): 12	TUTORIALS/WEEK: 1	TOTAL: 11 H
	PRACTICALS/SEMESTER: 2	TOTAL: 4 H
AIMS/OBJECTIVES CMOS is the technology driving the semiconductor industry and the digital revolution. It is the technology inside ICs (Integrated Circuits) allowing digital media like the iPod and mobile phones become sufficiently low-cost, small, and low-power to meet the demands of today's consumers. This course will allow the student to develop a strong understanding of the fundamentals of CMOS, how logic structures are built with CMOS and the challenges involved in the constant scaling to smaller technologies.		
SYLLABUS <ul style="list-style-type: none">• The MOSFET: physical principles of device operation; current voltage relationships, device models; second order effects.• Static Circuit Analysis: MOS inverters; the CMOS inverter transfer characteristic and its switching level; NAND and NOR gates; noise margin; transmission gate.• Dynamic Circuit Analysis: circuit lay-out, MOS transistor capacitances; inverter step response; gate delays; power dissipation.• Technology Scaling, 45 nanometers and beyond: Limitations and emerging technologies.• CMOS Logic Functions: generalised CMOS combinational logic; XOR and transmission gate logic; sequential logic elements, SRAM, DRAM.• CMOS Subsystem performance: RC gate delay models.		
ASSOCIATED LABORATORY/PROJECT PROGRAMME There will be associated laboratories which will be used to demonstrate the operation of some of the CMOS circuits studied in the lectures.		
RECOMMENDED TEXT(S) <ul style="list-style-type: none">• S. M. Kang and Y. Leblebici. CMOS Digital Integrated Circuits: Analysis and Design, McGraw-Hill, 1999.• Neil Weste, CMOS VLSI design: a circuits and systems perspective, Pearson Addison Wesley, 2005.• Jan M. Rabaey, Digital integrated circuits: a design perspective, Prentice-Hall , 2003.• Ken Martin, Digital integrated circuit design, Oxford University Press, 2000.• Richard Muller and Theodore Kamins, Device electronics for integrated circuits, John Wiley, 2003.		

LEARNING OUTCOMES

On successful completion of this course the student will be able to:

- Explain the electrical operation of the metal-oxide-semiconductor (MOS) field effect transistor.
- Analyse the fundamental static and dynamic performance of simple CMOS circuits noting design trade-offs.
- Implement various CMOS logic structures.
- Determine the limitations of current CMOS structures and alternatives.
- Extend knowledge of dynamic performance to more complex logic structures and systems.
- Demonstrate the fundamental principles of system design.

TEACHING STRATEGIES

The course is taught using a combination of lectures, tutorials and supporting laboratories. During the tutorials students will develop their problem solving skills by tackling problems based on the lecture material.

ASSESSMENT MODE(S)

The written examination will contribute 80% and continuous assessment will contribute 20% of the overall subject mark at the Annual examinations.

4C3 – Digital Control Systems

COURSE TITLE: Digital Control Systems		CODE: 4C3
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES:
LECTURER(S): Prof Frank Boland, Mr Dermot Geraghty	TEACHING ASSISTANT:	
TERMS: Semester 2	LECTURES/WEEK: 3	TUTORIALS/WEEK: 1
DURATION (WEEKS): 12	TOTAL: 33	TOTAL: 11
AIMS/OBJECTIVES This course introduces the technology of control systems and their applications in power electronic devices to control motors, such as AC, DC and Stepping Motors and IN process control. The aim is to equip the student with the ability to select and design suitable control systems. Aspects of the course include: Programmable Logic Controllers, Industrial PID controllers, Real Time control and Digital Control.		
SYLLABUS Review of Control System Theory including Root Locus and Frequency Response design methods. Process Modelling: Input-output models based on transfer functions and state-space models derived from engineering analysis. System identification. System Analysis: Basic behaviour of systems in open and closed loop in terms of stability, transient and steady-state responses Control System Design: Specification of the performance of controlled systems and the design of appropriate algorithms. Direct Digital Control, DDC, controllers and tuning. These topics will be illustrated by examples such as speed/position controllers as used in elevators, liquid-level control systems and robotic systems. Use of power electronic components to implement control systems. Introduction to Programmable Logic Controllers.		
ASSOCIATED LABORATORY/PROJECT PROGRAMME To be advised		
RECOMMENDED TEXT(S) G.F. Franklin, J.D. Powell, and A. Emami-Naeini, Feedback Control of Dynamic Systems, Addison-Wesley. G.F. Franklin, J.D. Powell, and M.L. Workman, Digital Control of Dynamic Systems, Addison-Wesley Supplementary Reading		

Course notes from a Chemical Engineering module on process control at the University of Texas at Austin, <http://www.che.utexas.edu/course/che360/index.html>

Course notes from an Electrical Engineering module on digital control at University of Saskatchewan, Canada, <http://www.engr.usask.ca/classes/EE/480/>

LEARNING OUTCOMES

On successful completion of this course, students will (be able to):

- understand the operation of AC and DC power control systems and Programmable Logic Controllers as commonly employed in industrial control systems
- describe the dynamical behaviour of controlled processes through use of time and transform domain descriptors
- explain the role and function of sampling and quantization in digital control loops
- explain criteria for stability and dynamic response constraints as applied to closed-loop control systems
- translate a word description of a controller specification into a formal description
- design compensators for closed loop control using industrial controllers
- carry out modelling and design of a digital controller using state-space methods

TEACHING STRATEGIES

The course is taught using a combination of lectures, laboratories and tutorials. The laboratory session introduces students to the apparatus of control systems including controlled rectifiers, such as the fully controlled full wave rectifying bridge. They are also introduced to speed control of a DC motor and implement P, PI and PID speed control of a DC motor. The transient behaviour of the control system is also investigated.

For the digital control section of the course the strategy is a mixture of lectures, problem solving tutorials and group based project work. During tutorials, students work on written analysis problems with the aid of the lecturer. The intention is to cover problems relating to the material covered in lectures during each week. The lecturer supervises each student as they work through problems, giving advice and assistance as required.

During the tutorial, the lecturer illustrates the essence of the solution to each problem once the class has made a realistic attempt. The project work is undertaken in groups of 3-4 students. The objective of the project is defined in descriptive terms and the group must decide on an appropriate approach and use Matlab as a tool for obtaining a practical solution.

ASSESSMENT MODE(S)

Assessment is based on an annual written examination and laboratory/practical work.

85% of the mark returned for 4C3 is derived from the annual examination held at the end of the academic year.

The remaining 15% of the marks are obtained from a laboratory and project programme.

4C4 – Telecommunications

COURSE TITLE: TELECOMMUNICATIONS		CODE: 4C4
LEVEL: Senior Sophister	CREDITS:	PREREQUISITES: 3C5
LECTURER(S): Dr. N. Marchetti		
TERMS: Semester 1	LECTURES/WEEK: 3	TUTORIALS/WEEK: 1
DURATION (WEEKS): 12	TOTAL: 33	TOTAL: 11
AIMS/OBJECTIVES <p>This course will expand on the third-year course (3C5) on Telecommunications, by applying communication principles to the analysis and design of wireless telecommunication networks. In this course, we will discuss the performance evaluation of networks through analytical, simulation and experimental methods. We will also study technologies and protocols for wireless and mobile networks, including wireless local area networks, cellular systems and ad hoc networks. Finally, we will study mobility in the Internet Protocol suite and application support for mobility. The course explores fundamental concepts in wireless communications through the study of current and emerging wireless network technologies.</p>		
SYLLABUS <ul style="list-style-type: none">▫ Wireless channel impairments and mitigation techniques, overview of wireless networks▫ Network performance evaluation▫ Introduction to queuing theory▫ Network traffic models: Poisson arrival process, self-similarity▫ Wireless local area networks: IEEE 802.11▫ Wireless personal area networks: IEEE 802.15, Zigbee and Bluetooth▫ Wide-area wireless and mobile networks: 2G/3G/LTE, IEEE 802.16▫ Cognitive radio and dynamic spectrum access▫ Mobile IP and transport-layer support for mobility▫ Mobile ad hoc networks and wireless sensor networks▫ Application-layer support for mobility and pervasive computing		
ASSOCIATED LABORATORY/PROJECT PROGRAMME <ul style="list-style-type: none">• Students will complete a project focusing on emerging topics in wireless communications and networks, such as dynamic access to spectrum and green communications.		
RECOMMENDED TEXT(S) <ol style="list-style-type: none">1) Jochen Schiller, "Mobile Communications," 2nd edition, Addison-Wesley, 2003.2) T. G. Robertazzi, "Computer Networks and Systems: Queuing Theory and Performance Evaluation," 3rd edition, Springer, 2000.		
LEARNING OUTCOMES <p>On completion of this course the student will be able to:</p> <ul style="list-style-type: none">▫ Design simulation experiments, develop traffic models, develop network models, and evaluate the performance of queues, local area networks, internetworks, and protocols;▫ Analyze the use of digital modulation and wireless communication methods in personal, local and wide area wireless and mobile networks;▫ Assess the operation of medium access protocols in contemporary wireless standards for personal area networks, local area networks, and metropolitan area networks, and discuss co-existence between different types of systems;▫ Explain and compare routing, service discovery, and management protocols used in mobile ad hoc networks and sensor networks;▫ Analyze proposed enhancements to the TCP/IP suite to support seamless mobility; and		

- Describe how applications make use of middleware and protocols to support mobile users.

TEACHING STRATEGIES**ASSESSMENT MODE(S)**

Continuous assessment will be adopted. The final marks for the course will be calculated according to:

- A project report and presentation (10%)
- Two in-class quizzes (15% each)
- End-of-year exam (60%)

4C5 – Digital Signal Processing

COURSE TITLE: Digital Signal Processing (4C5)		CODE: 4C5
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES: 3E3,3C1,3C5
LECTURER(S): Dr A. Quinn	TEACHING ASSISTANT:	
TERMS: Semester 1	LECTURES/WEEK: 3	TUTORIALS/WEEK: 1
DURATION (WEEKS): 12	TOTAL: 33	TOTAL: 11
<p>AIMS/OBJECTIVES</p> <p>The course provides a secure framework for designing and analyzing signal processing algorithms hosted by digital computers and by dedicated DSP hardware. Conceptual, arithmetic and practical issues are covered. Matlab—as the <i>lingua franca</i> for DSP software development—is the principal implementation pathway considered. Low-level software and hardware issues are only touched on. The course aims to deliver a framework for processing large classes of real signals, rather than dwelling on the local culture of specific physical, multimedia or other sources (though speech provides an illustrative context later in the course).</p> <p>It is a premise of the course that good DSP design must take account of the analogue context in which the DSP engine operates. Care is taken to address (i) the handling of the interface between the analogue and digital stages of a signal processing system, and (ii) the additivities available to the engineer who chooses to implement a design using DSP blocks with A/D and D/A interfaces, rather than just in analogue. Regarding (i), treatment of A/D is confined to uniform and idealized sampling, emphasizing the potential reversibility of the operation. Quantization is not formally covered, but its irreversibility is mentioned. Hence, the course perpetuates the tradition of fundamental DSP courses, in treating discrete-time—not digital—signals. Consideration of D/A is confined to reconstruction via zero-order hold. The practical importance of over-sampling is discussed.</p> <p>A keystone of the course is the analysis of real and even sequences. The student first encounters them during the design of generalized linear phase FIR filters via the windowing method, using a brick-wall gain specification, and the efficient realization of these filters is considered. Reciprocity both of the poles and zeros of these sequences is an important result, allowing them to be computed by autocorrelation under special assumptions. Ultimately, these assumptions lead to perhaps the most powerful result in the course, namely spectral factorization for regular random processes.</p> <p>Inversion of the bilateral z-transform is by partial fraction expansion, avoiding contour integration. Pole-zero placement is used as an informal design method for digital filters in the z-domain. The student is motivated to think creatively around the many degrees of freedom available, notably model order and the binary FIR/IIR choice. This work is developed with recourse to the SPTool GUI in Matlab. The student comes to appreciate how quick this approach is in prototyping digital filters, but also its limitations in satisfying detailed passband/stopband specifications. This motivates formal IIR filter design via bilinear transformation to the s-plane. Practical tasks in filtering and spectral shaping are addressed in the laboratory. In this way, the student is armed with practical knowledge of filter design, but also with the theoretical insights that allow any design to be assessed critically.</p> <p>DSP practice also informs the coverage of system realizations via MAC operations. The canonical structures are covered, with the aid of the Matlab roots and residuez functions, allowing high-order filters to be treated. Care is taken to convert signal flowgraph (SFG) characterizations into Matlab scripts, joining up the theory-design-implementation thread that distinguishes DSP as an engineering—not a mathematical—discipline. Computational load is assessed, particularly as a function of model order and FIR/IIR choice. Finite wordlength issues are touched on, but only qualitatively, as a means of encouraging classroom discussion of the additivities of a digital implementation (see point (ii) in paragraph two above).</p> <p>The coverage of information-bearing processes aims to bring the student to a point where they are equipped theoretically and practically with the DSP skills needed to handle real-world signals, such as the multimedia objects they will encounter later in 4C8. The autocorrelation sequence is revealed as a deterministic object</p>		

amenable to the methods developed earlier. The Wold representation—a unifying concept for the whole course—opens the way to advanced applications in coding, prediction and classification of random processes. The Wiener criterion is presented to the students as a means of estimating linear prediction coefficients for such applications. The students implement these ideas in the laboratory, in the context of all-pole modelling of unvoiced speech.

If time permits, sampling in the frequency domain will be addressed, in order to show the students how to implement linear convolution on a computer via frequency-domain operations. If time does not allow formal coverage of the DFT/FFT, then the ' $N_1 + N_2 - 1$ ' principle for successful DFT-based convolution will be covered qualitatively within a Matlab laboratory.

SYLLABUS

- **Sampling and reconstruction of analogue signals**
Review of analogue signal analysis: the energy spectral density

Ideal impulse sampling, periodic extension of the spectrum, aliasing and the sampling theorem

Reconstruction via zero-order hold: over-sampling
- **Discrete-time sequences**
Fundamental sequences and their unit sample sequence expansion

The discrete-time Fourier transform (DTFT) and its properties

Real and even sequences: (generalized) linear phase
- **The z-transform and its properties**
Rational function-type z-transform: partial fraction expansion

Linear convolution

Pole-zero characterization and spectral shaping

z-transform of real and even sequences: deterministic autocorrelation
- **Classes of discrete-time systems**
Linearity, time-invariance, causality, stability

The convolution theorem

FIR and IIR filters

Generalized linear phase (GLP) causal FIR filters

The DSP-enabled analogue signal processing system
- **Digital filter design**
Informal design via pole-zero placement

Formal GLP causal FIR filter design via the windowing method

Formal causal stable IIR filter design via the bilinear transformation

- **Realization of digital filters**
The difference equation and signal flowgraph characterizations

Cascade and parallel structures

Efficient realization of GLP causal FIR filters

Finite wordlength effects: Jordan's ordering and pairing rules

- **Discrete-time random processes**
The regular random process and its complex spectrum

Spectral factorization and the Wold representation

ARMA-class random processes and power spectra

The Wiener linear predictor

- **Frequency-domain sampling**
Time-aliasing in periodically extended sequences

The discrete Fourier transform (DFT)

Circular convolution

Linear convolution via the DFT and the overlap-add method

The radix-2 decimation-in-time fast Fourier transform (FFT) algorithm

ASSOCIATED LABORATORY/PROJECT PROGRAMME

There will be four Matlab-based laboratories held during the semester.

RECOMMENDED TEXT(S)

The main recommended text for the course is:

1. Proakis, J. G. and Manolakis, D.G., *Digital Signal Processing*, 4th ed., Prentice Hall, 2006.

Secondary recommended texts are as follows:

2. Oppenheim, A.V. and Schaffer, R.W., *Discrete-Time Signal Processing*, 3rd ed., Prentice Hall, 2009.
3. Ludeman, L.C., *Fundamentals of Digital Signal Processing*, Wiley, 1988.

LEARNING OUTCOMES

On successful completion of this module the student will be able to:

- Evaluate the spectrum after sampling and reconstruction of an analogue signal, either with or without discrete-time processing
- Evaluate the spectrum of parametric sequences
- Express certain real and even sequences via autocorrelation
- Classify discrete-time systems into key classes
- Design FIR and IIR filters via pole/zero placement in the z -domain
- Design GLP causal FIR filters satisfying brick-wall gain specifications
- Design IIR filters to meet detailed gain specifications, via the bilinear transformation
- Implement digital filters in Matlab observing computational constraints (pipeline and parallel flows)

- Represent a regular random process as the output of a system driven by noise
- Derive optimal all-pole models for stationary random process realizations, such as unvoiced speech, and use them for coding, prediction and classification
- Implement linear convolution of arbitrarily-long sequences on a digital computer, via the FFT

TEACHING STRATEGIES

The course is delivered through the medium of lectures and tutorials, with a 3:1 ratio between them. The technical and mathematical concepts will, at all times, be sited in the practical design and analysis concerns of the DSP engineer. Hence, formal mathematical procedures will be used only to answer these practical problems, rather than being the source of the problems themselves. The over-arching vision of the course is to minimize the number of distinct concepts required, but to understand deeply and exploit fully those concepts that are deployed. In this way, connectivity and coherence will be the features of the teaching approach. Mathematics anxieties – involving functions of complex variables, summing of series, manipulating the Dirac delta-function, etc. – will be addressed head-on. The course notes will be provided after each lecture, via scans uploaded to the course webpage.

DSP is a practical art, and Matlab will be central to the course as a means of ‘exporting’ theoretical results into the implementational domain. It will be used during lectures via the computer projection facilities in the theatres, and also, of course, during the four laboratories, which will be supported by an experienced TA. While hardware design will not form a part of this course, the key role of Matlab in the prototyping of DSP software routines will inform the course, and the student will emerge with experience of the Matlab signal processing toolbox and its GUI (SPTool).

The need for problem-solving experience is vital. This will be gained primarily through the weekly tutorials, but also via regular homework assignments, where, again, the student will be encouraged to use Matlab in developing their solutions. Reference solutions to all tutorials and assignments will be made available to the students subsequently via the course webpage.

ASSESSMENT MODE(S)

85% of the final mark is determined via the annual examination, and the remaining 15% via an end-of-semester quiz.

4C6 – Microelectronics Technology

COURSE TITLE: MICROELECTRONICS TECHNOLOGY		CODE: 4C6
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES: NONE
LECTURER(S): Tatiana Perova	TEACHING ASSISTANT:	1
TERMS: Semester 1	LECTURES/WEEK: 3	TUTORIALS/WEEK: 1
DURATION (WEEKS): 12	TOTAL: 33	TOTAL: 11
AIMS/OBJECTIVES		
<p>To obtain a good understanding of the basic processes used to fabricate an integrated circuit; to be able to apply this knowledge to the design and layout of metal-oxide-semiconductor (MOS) and bipolar circuits and to develop the skills necessary to design a processing schedule for a simple integrated circuit (IC).</p>		
SYLLABUS		
<ul style="list-style-type: none">• Processing techniques: The following techniques will be examined in some detail: crystal growth; diffusion; ion-implantation; oxidation; lithography; metallisation; plasma etching. There will be a guest speaker from industry to give an industrial perspective.• Spectroscopy: Introductory material on application of Fourier Transform Infrared (FTIR) and Raman spectroscopy to IC technology will be presented.• Devices/Circuits: Techniques for fabrication of the following devices, using the processing techniques described above, will be studied: Bipolar: discrete transistor; monolithic IC transistor; resistor; diode; capacitor; circuit layout for ECL NOR gate. MOS: PMOS, NMOS and CMOS technologies will be introduced and the problem of latch-up will be discussed.• Silicon-on-Insulator will be proposed as an alternative to bulk silicon.		
ASSOCIATED LABORATORY/PROJECT PROGRAMME		
<p>The course provides the students with an opportunity to fabricate a simple PMOS integrated circuit in Microelectronic Technology Laboratory. The device contains transistors, a resistor, a capacitor and two logic devices. During the laboratory course the participants will acquire and appreciation of the various stages in the fabrication of the device. This will include wafer cleaning, field oxidation, the photolithography process, window etching, diffusion, gate oxidation and metallization.</p> <p>The participants are also fully trained in the safe handling of chemicals. The final part of the course consists of a lecture on device testing and packaging in the lab.</p>		
RECOMMENDED TEXT(S)		
<ol style="list-style-type: none">1. <i>Introduction to Microelectronic Fabrication</i>, Vol. V, Richard C. Jaeger, Addison-Wesley, 1988.2. <i>Fundamentals of Semiconductor Fabrication</i>, G.S. May and S.M. Sze, Wiley, 2003.3. <i>Semiconductor Devices, Physics and Technology</i>, S.M. Sze, Wiley, 2002.4. <i>Silicon Processing for the VLSI Era</i>, Vol. 1, Process Technology, S. Wolf and R.N. Tauber, Lattice Press, 1986.		

LEARNING OUTCOMES

On completion of this course the student will have gained in-depth knowledge on:

- 1) the processing techniques used to fabricate an integrated circuit (IC);
- 2) how these processes are applied to the production of both bipolar and MOS ICs;
- 3) importance of the characterisation techniques and their use in IC technology.

TEACHING STRATEGIES

There will be 3 lectures plus 1 tutorial per week during Michaelmas Term for the course Microelectronic Circuits (4C6). Additionally, the students must complete a course of laboratory work which is compulsory. At the discretion of the course examiners, up to 5% of the available marks for 4C6 may be assigned for satisfactory completion of the laboratory course.

ASSESSMENT MODE(S)

There will be a single written exam. 15% of the overall mark is from a compulsory and laboratory work assignment for 4C6 during Michaelmas Term. The remainder of the mark is from the written exam and assignment from 4C6.

4C7– Digital Communications

COURSE TITLE: DIGITAL COMMUNICATIONS		CODE: 4C7
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES:
LECTURER(S): Mr Liam Dowling, Prof Luiz da Silva	TEACHING ASSISTANT:	
TERMS: Semester 2	LECTURES/WEEK: 3	TUTORIALS/WEEK: 1
DURATION (WEEKS): 12	TOTAL: 33	TOTAL: 11
AIMS/OBJECTIVES The objective of the 4C7 Digital Communications course is to develop the fundamental theory to enable students to design and analyse digital communication systems. The course builds on the material covered in the 3C1 Signals and Systems and 3C5 Telecommunications courses.		
SYLLABUS <ul style="list-style-type: none">• <i>Mathematical definition of a random process</i> Stationary processes. Mean, correlation, and covariance functions. Ergodic processes. Transmission of a random process through a linear time-invariant filter. Power spectral density. Gaussian Process. Narrowband noise.• <i>Baseband pulse transmission</i> Matched filter. Error rate due to noise. Intersymbol interference. Nyquist's criterion for distortionless baseband binary transmission. Correlative-level coding. Baseband M-ary PAM transmission.• <i>Signal space analysis</i> Conversion of the continuous AWGN channel into a vector channel. Likelihood functions. Coherent detection of signals in noise: Maximum likelihood decoding. Correlation receiver. Probability of error.• <i>Passband Digital Transmission</i> Binary phase-shift keying. Error probability of binary phase-shift keying. Generation and detection of coherent quadriphase-shift keying signals. Error probability of quadriphase-shift keying. Coherent frequency-shift keying.• <i>Error control coding</i> Linear block codes. Cyclic codes. Convolutional codes. Maximum likelihood decoding of convolutional codes. Trellis coded modulation.• <i>Advanced topics in digital communications</i> Spread spectrum. Multi-carrier modulation. Power and interference control. Topology control. Channel assignment algorithms.		
ASSOCIATED LABORATORY/PROJECT PROGRAMME		
RECOMMENDED TEXT(S) S. Haykin and M. Moher, <i>Communication Systems</i> , 5th ed., Wiley, 2009.		

LEARNING OUTCOMES

On successful completion of this module the student will be able to apply fundamental theory to the design and analysis of baseband and band-pass digital communication systems.

TEACHING STRATEGIES

This module is taught using a combination of lectures and problem solving tutorials.

4C8 – Digital Media Processing

COURSE TITLE Digital Media Systems		CODE: 4C8
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES: 3C1
LECTURER(S): Dr. David Corrigan	TEACHING ASSISTANT:	
TERMS: Semester 2	LECTURES/WEEK: 3	TUTORIALS/WEEK: 1
DURATION (WEEKS): 12	TOTAL: 33	TOTAL: 11
AIMS/OBJECTIVES <i>Digital Media Systems</i> builds on ideas in DSP as they apply to higher level system concepts in the application area of Digital Media Systems. Students begin with an introduction to digital video processing algorithms that form the core of important devices such as the iPod, SKY set top boxes and the DVD/BluRay player. Attention is also given to emerging technology in digital cinema postproduction where hardware/software co-design issues are increasingly becoming relevant for the Digital Intermediate Workflow. Core aspects of this technology will be discussed within the context of implementation on embedded devices e.g. ARM cores for the consumer market. At the end of this course students should have a basic knowledge of MPEGx, Motion Estimation, Film Pre-production and Visual Media storage systems/formats.		
RECOMMENDED TEXT(S) <ul style="list-style-type: none">• Digital Signal processing for Multimedia Systems, Parhi and Nishitani, Marcel Dekker 1999• Digital Video an Introduction to MPEG2, Haskell, Puri and Netravali, Chapman and Hall 1997• JPEG2000, Acharya and Tsai, Wiley, 2005• Two Dimensional Image Processing, J. Lim, Prentice Hall		
LEARNING OUTCOMES On completion of this course the student will be able to <ol style="list-style-type: none">1. Describe the elements of current media broadcast formats2. Describe the building blocks of emerging compression systems and select processing modules to enable efficient compression3. Design building blocks for image up/down sampling4. Implement a simple motion estimator in Matlab, and the essential building block in an embedded device, and assess its performance for real image sequences5. Diagnose problems leading to visual quality degradation in visual communications systems		
TEACHING STRATEGIES The strategy is a mixture of lectures, problem solving laboratories and interactive class demonstrations. During tutorials, students work on design problems with the aid of the lecturer. Material covered in classes preceding the tutorial are examined and questions relating to this are posed by the lecturer. Once a realistic attempt has been attempted by the class, the lecturer then proceeds to go through the solution piecemeal, encouraging the students to actively engage in questioning the proposed solutions and its merits based on material presented in class.		
ASSESSMENT MODE(S) The final exam counts for 75% of overall mark. 25% is given to continuous assessment made up of Laboratory work. Note that it is expected that the student should be able to complete the problems posed in laboratories during the timetabled hours.		

4D1a –Distributed Systems

COURSE TITLE: Distributed Systems		CODE: CS4D1A
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES:
LECTURER: Stephen Brooks	TEACHING ASSISTANT:	
TERMS: Michaelmas Term	LECTURE/WEEK: 3	TUTORIALS/WEEK: 1
DURATION (WEEKS): 12	TOTAL: 33	TOTAL: 11
AIMS/OBEJECTIVES		
<p>Distributed Systems is a one-term course that covers both practical and theoretical topics in the area of distributed systems. Common algorithms, models and techniques used in the areas of Distributed Systems are presented, along with a number of case studies. Throughout the course, students are encouraged to consider the suitability of an algorithm, model or technique in different circumstances and to compare different approaches to solving problems, the aim of the course being to enable students to design, construct and reason about a distributed system of moderate complexity. This course also places a strong emphasis on independent practical work, encouraging students to develop their communication, research and programming skills while tackling real-world design problems and using real-world technologies. An individual project provides a focus for discussion of practical issues of constructing distributed systems.</p>		
SYLLABUS		
<ul style="list-style-type: none">• Concurrency• Network Programming• RPC/Remote Object Technology and Middleware• Enterprise Infrastructure and Paradigms• P2P• Advanced Network Applications		
ASSOCIATED LABORATORY/PROJECT PROGRAMME		
<p>An individual project provides a focus for discussion of practical issues of constructing distributed systems, and that contributes towards the continuous assessment for the module.</p>		
RECOMMENDED TEXT(S)		
<p>Key sources include</p> <ul style="list-style-type: none">· "Distributed Operating Systems", Andrew S. Tanenbaum, Prentice-Hall, 1995.· "Engineering Distributed Objects", Wolfgang Emmerich, Wiley, 2000.· "Distributed Systems: Concepts and Design", George Coulouris, 3rd Edition, Addison-Wesley, 2001. <p>Extensive use will also be made of research papers and other material from the literature.</p>		

LEARNING OUTCOMES

When students have successfully completed this module they should be able to:

- Describe the basic characteristics, structure and operation of a distributed system;
- Identify appropriate architectural models for distributed problem scenarios;
- Design, construct, document and test distributed system solutions to realistic real-world problems;
- Reason about the performance tradeoffs of decentralised architectures;
- Make use of appropriate documentation and reference material.

TEACHING STRATEGIES

The teaching strategy is a mixture of lectures, problem-solving tutorials and substantial coursework exercises. There is an emphasis on practical design work and students will gain practical experience by completing a number of exploratory in lecture design problems and in individual project work.

ASSESSMENT MODE(S)

Assessment is by examination (80%) and continuous assessment (5%), and assessment of an individual project (15%).

Continuous assessment is composed of a number of marked tutorial exercises and programming assignments.

4D1b – Advanced Microprocessor Systems

COURSE TITLE: Advanced Microprocessor Systems		CODE: CS4D1B
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES:
LECTURER:	TEACHING ASSISTANT:	
TERMS: Hilary Term	LECTURE/WEEK: 3	TUTORIALS/WEEK: n/a
DURATION (WEEKS): 12	TOTAL: 33	TOTAL: N/A
AIMS/OBJECTIVES		
<p>This module builds on CS3D2 (Microprocessor Systems 2), focusing on those aspects of microprocessor architectures that relate to the memory hierarchy. Particular emphasis is placed on the performance of architectural features in practice. The major topics covered include, caches, multiprocessors, multiprocessor cache coherency and multiprocessor synchronization. The aim of the course is to give students an appreciation of the close relationship between microprocessor architectures and the performance achieved by the programs that are executed on those architectures. Students will gain practical experience designing and writing software to take advantage of underlying microprocessor architectural features and conducting experiments to evaluate performance.</p>		
SYLLABUS		
<ul style="list-style-type: none">• microprocessor cache organization and operation;• microprocessor cache performance;• review of virtual memory and its implementation in modern microprocessors• relationship between virtual memory and microprocessor caches• multiprocessor architectures;• multiprocessor cache coherency protocols;• multiprocessor synchronization;• programming techniques for multiprocessor systems.		
ASSOCIATED LABORATORY/PROJECT PROGRAMME		
<p>Approximately 11 hours of laboratory work with lecturer to support a number of practical projects that contribute towards the continuous assessment for the module.</p>		
RECOMMENDED TEXT(S)		
<ul style="list-style-type: none">• John L. Hennessey and David A. Patterson, “Computer Architecture: A Quantative Approach”.• David A. Patterson and John L. Hennessey, “Computer Organization and Design: The Hardware / Software Interface”.		

LEARNING OUTCOMES

On successful completion of this course, students will (be able to):

1. describe the organisation and operation of a microprocessor cache;
2. reason about the performance of a given memory hierarchy;
3. design a simulation that models the behaviour of a memory hierarchy and design suitable experiments to compare the performance of different cache configurations;
4. discuss the challenges that arise when using caches in a multiprocessor environment, describe several well-known approaches for implementing cache coherency and reason about their performance;
5. explain the advantages of using virtual memory, show how virtual addresses are mapped efficiently to physical addresses and explain the role of the operating system in implementing a virtual memory system;
6. describe and discuss several multiprocessor synchronisation algorithms and be able to reason about their performance;
7. design and implement simple programs for multiprocessor systems and quantitatively evaluate their performance.

TEACHING STRATEGIES

The teaching strategy is a mixture of lectures, problem-solving tutorials and substantial coursework exercises. Students make use of interactive web-based animations to explore the operation of caches and cache coherency protocols. There is an emphasis on practical laboratory work and students will gain practical experience by completing a number of laboratory projects.

ASSESSMENT MODE(S)

Assessment is by continuous assessment and examination. The examination will contribute 80% of a student's mark for the module, with continuous assessment contributing the remaining 20%. Continuous assessment is in the form of a number of projects and laboratory exercises.

4D2a – Data Engineering

Module Code	4D2a
Module Title	Information Systems and Data Engineering
Pre-requisites	None
ECTS	5
Chief Examiner	Prof Vinny Wade
Teaching Staff	Prof Vinny Wade
Delivery	Information systems management course runs over the entire 1 st Semester. In each week, there are three one-hour lectures. However, tutorials are scheduled in place of a lecture every two weeks. Alongside the lectures, a full online course on the application of database language SQL is delivered. This online course must be taken by all students taking this module. Project work which integrates the lectures, tutorials and online course accounts for 20% of the module's overall mark. Attendance at all lectures and tutorials is compulsory.
Aims	<p>This course is focused on modelling of information and database system technology. More specifically it focuses on state-of-the-art database technology, from both the user and systems perspectives.</p> <p>From a system engineering perspective, the course examines the concepts and algorithms for: Transaction processing, Concurrency control, Metadata Representation, Semantic Representation and Active Databases, Recovery, Database Security Policies, Integration of databases on the web, Semantic Web, Object Oriented, Object Relational Databases, Web based Databases.</p> <p>From an information designer's perspective, the course examines the theoretical model underpinning relational databases, functional dependency theory and normalisation (for information modelling), functional dependency modelling, Object Relational Modelling, implementation of databases and database applications using SQL PL (relational database query language) as well as PHP/SQL.</p> <p>Thus the course is intended to enable the students to design information models and implement these models in state of the art databases (relational and native web databases), as well as be able to analyse and evaluate approaches to information organisation, storage, transaction support and management.</p>
Learning Outcomes	<p>Having completed the course the student will be able to:</p> <ul style="list-style-type: none"> • Define and comprehend the theoretical and practical issues in management, retrieval, organisation, indexing techniques and storage of large quantities of data • Model data within an organisation or enterprise, and analyse data relationships for exploitation within database management systems • Program a Database management systems for database creation, information retrieval and database management • Comprehend how to web based database applications using PHP and SQL/PL • Analyse and assess various database concurrency protocols and algorithms to assess their performance and relative appropriateness in differing operating environments.

	<ul style="list-style-type: none"> • Evaluate existing concurrency control algorithms and appropriate data structures for data management. • Model Object Relational databases for OOPL / RDBMS integration • Apply meta-data modelling techniques for managing multimedia resources (XML, Ontologies)
Syllabus	<p>Specific topics addressed in this module include:</p> <ul style="list-style-type: none"> • Overview of database systems and users • Database system concepts and architecture • Relational Database management • Relational Model • Functional dependency theory and normalisation • On-line module on SQL (relational database query language) • SQL/PL (SQL as a programming language) • Transaction processing • Query Processing • Concurrency control • Modelling Semantics • Active Databases • Database Recovery • Security and authorisation • Object-oriented database systems • Software Engineering for database applications and database development • Integration of Web based technology and Database (PHP based web database applications) • Object Relational DBMSs • Web Databases • Semantic Web Architecture and Technologies (Ontologies) • Meta Data models for open content management
Assessment	<p>Assessment is by examination (80%) and continuous assessment (20%).</p> <p>Continuous assessment is focused around assignments and projects.</p>
Bibliography	<p>Recommended text:</p> <p>Fundamentals of Database Systems by Elmasri & Navathe (6th Edition) published by Addison Wesley;</p> <p>Database Systems 5th Edition, Connolly & Begg, published by Addison Wesley.</p> <p>Database Systems – The Complete Book, by Garcia-Molena, Ullman & Widom, published by Prentice Hall;</p>
Website	<p>The course work notes are available in the TCD Learning Management System. (TCD Moodle)</p>

4D2b – Knowledge Engineering

COURSE TITLE: Knowledge Engineering		CODE: 4D2b
LEVEL: Senior Sophister	CREDITS: 5	PREREQUISITES: 4D2a
LECTURER(S): Dr. Owen Conlan	TEACHING ASSISTANT:	
TERMS: Semester 2	LECTURES/WEEK: 2	TUTORIALS/WEEK: 1
DURATION (WEEKS): 12	TOTAL: 22	TOTAL: 11
AIMS/OBJECTIVES		
<p>Knowledge Engineering is a one semester course taken by Senior Sophister CD and D Stream students. It builds on the outcomes of Data Engineering by introducing knowledge engineering techniques such as information organisation and storage, ontological reasoning and artificial intelligence. addresses techniques and technologies for organizing, structuring and storing data. Specifically, this module focuses on the organization and sorting of files including topics such as hardware optimization, indexing and dynamic data structures. These topics are expanded to include advanced Knowledge Management approaches, such as Information Retrieval (IR) and Data Mining, and technologies, such as advanced XML and ontologies. The Artificial Intelligence aspect of this module investigates search and rule-based systems, case-based reasoning and the semantic web.</p> <p>The objectives this module are to give students an understanding of the organization and manipulation of knowledge and data using a variety of techniques.</p>		
SYLLABUS		
<ul style="list-style-type: none">• Record storage and primary file organization;• Index structures for file;• Fundamental Technologies and Techniques for DBMSs<ul style="list-style-type: none">○ Storage Hardware○ File Basics, File Organisation, File Sorting○ Hashing○ Single-Level Indexes, Multi-Level Indexes, B Trees, B+ Trees• Knowledge Management<ul style="list-style-type: none">○ Advanced XML<ul style="list-style-type: none">▪ XSLT; XPath; XQuery; XUpdate○ Ontologies<ul style="list-style-type: none">▪ OWL; Reasoning▪ Application in Semantic Web• Artificial Intelligence<ul style="list-style-type: none">○ Rule-based systems; Case-based reasoning; Bayesian Networks• Retrieving textual information:<ul style="list-style-type: none">○ IR vs database retrieval;○ Classic IR models (Boolean, vector space, probabilistic);○ Retrieval effectiveness - precision and recall.• Information retrieval on the web:<ul style="list-style-type: none">○ First generation search engines;○ The HITS algorithm;○ Google; Scamming Google;○ The next generation of search engines.		
ASSOCIATED LABORATORY/PROJECT PROGRAMME		
None		

RECOMMENDED TEXT(S)

1. Fundamentals of Database Systems by Elmasri & Navathe (5th Edition) published by Addison Wesley
2. Charles F. Goldfarb's XML Handbook by Charles F. Goldfarb, Paul Prescod (5th Edition) published by Prentice Hall

LEARNING OUTCOMES

On completion of this course the student will be able to:

1. Design and engineer a Database Management System (DBMS) with consideration given to hardware, information organization, hashing and indexing.
2. Understand the structure of, and apply advanced manipulation techniques to, XML documents.
3. Develop skills in managing knowledge using Ontological and Semantic Web technologies.
4. Understand and be able to apply different Artificial Intelligence techniques.
5. Understand and Compare different Information Retrieval techniques, specifically those used on the web.

TEACHING STRATEGIES

The course is taught using a combination of lectures and tutorials. During the tutorials students will develop their problem solving skills by tackling problems based on the lecture material.

ASSESSMENT MODE(S)

The written examination will contribute 100% of the overall subject mark at the Annual examination.

4D3a – Computer Graphics I

Module Code	CS4D3A
Module Title	Computer Graphics
Pre-requisites	C or C++ programming, freshman mathematics
ECTS	5
Chief Examiner	PROF CAROL O’SULLIVAN
Teaching Staff	PROF CAROL O’SULLIVAN
Delivery	Lectures/week: 2, Tutorials/week: 1 There are 10 labs covering aspects of 3D modeling, OpenGL, curves and surfaces. Attendance to all labs is compulsory.
Aims	<p>Computer Graphics is a final year option which may be taken by students of Engineering, Computer Science and Mathematics.</p> <p>The objective of this course is to equip the students with the fundamental understanding of the major elements of Computer Graphics and explore related areas including geometric modelling, rendering and animation. The main focus of the course is on the mathematics and algorithms used in the synthesis of computer graphics imagery and animation, and their practical application. Students are introduced to the standard architectures of modern graphical applications including details on the underlying hardware and low-level software components common to all such systems. The course is intended to enable students to bridge the gap between these low-level fundamental, components common to all computer applications, and the high-level abstract output in most interactive graphical applications.</p> <p>Students are also introduced to OpenGL, a modern high-level graphics API which is widely used for 3D Design and Visualisation, along with the industry standard modelling software, 3D Studio Max, and this software is used throughout the course to demonstrate concepts and to allow the students to develop their own 3D models, scenes and applications.</p>
Learning Outcomes	<p>When students have successfully completed this module they should be able to:</p> <ol style="list-style-type: none">1. Write graphical programs, using OpenGL or a similar graphics API, of moderate complexity;2. Select an object or scene representation, create a model using modelling software, and export this model for use in an interactive application;3. Discriminate between the different rendering choices for displaying objects, such as global or local illumination algorithms, and select the correct solution for the application area;4. Derive and solve the mathematical formulations that underpin the practical aspects of creating, animating and rendering objects and scenes.

Syllabus	<p>Specific topics addressed in this module include:</p> <ul style="list-style-type: none"> • An introduction to computer graphics; problem domain and applications; • Modelling - data sources and acquisition; modelling software; representation schemes; • Linear algebra - two and three dimensional transforms; geometric operations; hierarchical 3D transformations; • The computer graphics pipeline and the OpenGL API for 3D computer graphics; • Projection and viewing; window to viewport transformation; • Illumination models and rendering algorithms; colour, shading algorithms (Gouraud and Phong), local and global illumination;
Assessment	<p>Assessment is by examination and practical coursework. End-of-year examination contributes 80% of the final mark, whilst the total coursework contributes the remaining 20%. The practical exercises are conducted individually by each student and involve the development of a real-time, interactive application that involves navigating and manipulating a 3D environment.</p>
Bibliography	<p>No course text, although students are encouraged to use appropriate texts and reference documentation, where necessary. For example:</p> <ol style="list-style-type: none"> 1. <i>Introduction to Computer Graphics</i>, Foley, Van Dam, Feiner, Hughes and Phillips 2. <i>OpenGL Programming Guide: The Official Guide to Learning OpenGL</i>, (5th edition), Dave Shreiner, Mason Woo, Jackie Neider and Tom Davis
Website	<p>The course material is provided on the website: http://isg.cs.tcd.ie/cosulliv/teaching.html</p>

4D3b– Computer Graphics II

Module Code	CS4D3B
Module Title	Computer Graphics 2
Pre-requisites	C or C++ programming, freshman mathematics
ECTS	5
Chief Examiner	DR RACHEL MCDONNELL
Teaching Staff	DR RACHEL MCDONNELL
Delivery	Lectures/week: 2, Tutorials/week: 1 There are 10 labs covering aspects of rendering, and animation using OpenGL and 3DS Max. Attendance to all labs is compulsory.
Aims	<p>Computer Graphics 2 is a final year option which may be taken by students of Computer Engineering.</p> <p>The objective of this course is to equip the students with the fundamental understanding of the major elements of Computer Graphics and explore related areas including advanced rendering and animation. The main focus of the course is on the mathematics and algorithms used in the synthesis of computer graphics imagery and animation, and their practical application. Students are introduced to the standard architectures of modern graphical applications including details on the underlying hardware and low-level software components common to all such systems. The course is intended to enable students to bridge the gap between these low-level fundamental, components common to all computer applications, and the high-level abstract output in most interactive graphical applications.</p> <p>Students are also introduced to OpenGL, a modern high-level graphics API which is widely used for 3D Design and Visualisation, along with the industry standard modelling software, 3D Studio Max, and this software is used throughout the course to demonstrate concepts and to allow the students to develop their own 3D models, scenes and applications.</p>
Learning Outcomes	<p>When students have successfully completed this module they should be able to:</p> <ol style="list-style-type: none">5. Write graphical programs, using OpenGL or a similar graphics API, of moderate complexity;6. Create an animation of a 3D object, and discriminate between the options available such as scripted vs. procedural, stylised vs. physically-based etc.;7. Derive and solve the mathematical formulations that underpin the practical aspects of creating, animating and rendering objects and scenes.

Syllabus	<p>Specific topics addressed in this module include:</p> <ul style="list-style-type: none"> • Ray tracing - ray casting, Whitted illumination model, ray-traced shadows, ray-sphere/plane/polygon intersection algorithms; reflection and refraction, anti-aliasing/super-sampling strategies. • Radiosity – computing form factors, solving the linear system, meshing adaptively. • Animation Basics - principles; classes of animation; keyframing and interpolation; ordinary differential equations. • Advanced Animation – physically based animation, particles, facial animation.
Assessment	<p>Assessment is by examination and practical coursework. End-of-year examination contributes 80% of the final mark, whilst the total coursework contributes the remaining 20%. The practical exercises are conducted individually by each student and involve the development of a real-time, interactive application that involves navigating and manipulating a 3D environment.</p>
Bibliography	<p>No course text, although students are encouraged to use appropriate texts and reference documentation, where necessary. For example:</p> <ol style="list-style-type: none"> 3. <i>Introduction to Computer Graphics</i>, Foley, Van Dam, Feiner, Hughes and Phillips 4. <i>OpenGL Programming Guide: The Official Guide to Learning OpenGL</i>, (5th edition), Dave Shreiner, Mason Woo, Jackie Neider and Tom Davis
Website	<p>The course material is provided on the website: http://isg.cs.tcd.ie/mcdonner/teaching.html</p>

4D4a – Computer Vision I

Module Code	CS4D4a
Module Title	Vision Systems I
Pre-requisites	A working knowledge of C++
ECTS	5
Chief Examiner	Dr. Kenneth Dawson-Howe
Teaching Staff	Dr. Kenneth Dawson-Howe
Delivery	The teaching strategy employed on this course is a mixture of lectures and problem-solving tutorials/laboratories. The laboratory assignments allow the students to appreciate the difficulties of actually realising real solutions using computer vision. The tutorials allow the students to develop a better understanding of the material and to practise the design of appropriate solutions. Students make use of OpenCV , an open source, computer vision library, to experiment with many of the computer vision techniques, and to implement their assignments.
Aims	The aim of this module is to give students a firm understanding of the theory underlying the processing and interpretation of visual information and the ability to apply that understanding to ubiquitous computing and entertainment related problems. It provides them with an opportunity to apply their problem-solving skills to an area which, while it is firmly part of computer science/engineering, draws strongly from other disciplines (physics, optics, psychology). The course is based around problems so that the technology is always presented in context and during some tutorials students work in groups to design solutions to real world problems using the techniques that they have been taught. In addition, the course has a significant practical component so that students can appreciate how difficult it can be to apply the technology.
Learning Outcomes	When students have successfully completed this module they should be able to: <ul style="list-style-type: none"> • design solutions to real-world problems using computer vision. • develop working computer vision systems using C++. • critically appraise computer vision techniques. • explain, compare and contrast computer vision techniques.
Syllabus	<p style="text-align: center;"><u>Specific topics addressed in this module include:</u></p> <ul style="list-style-type: none"> • image digitisation and colour; • binary image processing including mathematical morphology, connected components analysis, and motion analysis; • geometric image transforms; • noise and smoothing; • edge based processing including edge detection, contour extraction and representation; • recognition techniques including template matching, statistical pattern recognition, and the Hough transform; • texture; • region based processing;
Assessment	The labs and assignments account for 20% of the final mark and the exam

	80%. Students must answer 2 out of 3 exam questions.
Bibliography	<ol style="list-style-type: none">1. Image Processing, Analysis and Machine Vision. Milan Sonka, Vaclav Hlavac & Roger Boyle, Thompson, Third Edition 2008.2. Learning OpenCV, Gary Bradski & Adrian Kaehler, O'Reilly, 2008.
Website	http://www.scss.tcd.ie/CourseModules/CS4053/

4D4b– Computer Vision II

Module Code	CS4D4b
Module Title	3D Vision Systems
Pre-requisites	A working knowledge of C++ Completed CS4D4a
ECTS	5
Chief Examiner	Mr. Peter Redmond
Teaching Staff	Mr. Peter Redmond
Delivery	The teaching strategy employed on this course is a mixture of lectures and problem-solving tutorials/laboratories. The laboratory assignments allow the students to appreciate the difficulties of actually realising real solutions using computer vision. The tutorials allow the students to develop a better understanding of the material and to practise the design of appropriate solutions. Students make use of OpenCV , an open source, computer vision library, to experiment with many of the 3D computer vision techniques, and to implement their assignments.
Aims	The aim of this module is to give students a firm understanding of the theory underlying the processing and interpretation of visual information and the ability to apply that understanding to ubiquitous computing and entertainment related problems. It provides them with an opportunity to apply their problem-solving skills to an area which, while it is firmly part of computer science/engineering, draws strongly from other disciplines (physics, optics, psychology). The course is based around problems so that the technology is always presented in context and during some tutorials students work in groups to design solutions to real world problems using the techniques that they have been taught. In addition, the course has a significant practical component so that students can appreciate how difficult it can be to apply the technology.
Learning Outcomes	When students have successfully completed this module they should be able to: <ul style="list-style-type: none"> • design solutions to real-world problems using 3D computer vision. • develop working 3D computer vision systems using C++. • critically appraise 3D computer vision techniques. • explain, compare and contrast 3D computer vision techniques.
Syllabus	<p style="text-align: center;"><u>Specific topics addressed in this module include:</u></p> <ul style="list-style-type: none"> • 3D vision including camera calibration, stereopsis and the fundamental matrix; • Structure from motion • Visual SLAM • Feature Extraction and description • Feature Classification and Selection • Optical flow • Applications of 3D
Assessment	The labs and assignments account for 20% of the final mark and the exam 80%. Students must answer 2 out of 3 exam questions.
Bibliography	3. Image Processing, Analysis and Machine Vision. Milan Sonka,

	<p>Vaclav Hlavac & Roger Boyle, Thompson, Third Edition 2008.</p> <ol style="list-style-type: none">4. Learning OpenCV, Gary Bradski & Adrian Kaehler, O'Reilly, 2008.5. Computer Vision, A modern approach. David Forsyth and Jean Ponce, 2003
Website	http://www.scss.tcd.ie/CourseModules/CS4053/

4E1 – Management for Engineers

Course Organisation

This subject is taught in the first semester

Semester 1 – weeks 1 to 6

Lecturer: Dr. Peter Redmond, School of Computer Science and Statistics

Course Description and Aims

This course aims to introduce students to the concepts and tools of project management. We will use a project management simulation SimProject www.fissure.com to develop the practical skills of project management..

Course Outline

Week	Lecture Learning Outcomes
1	Course Outline and introduction to Project management concepts Project definition and organization
2	Project Planning tools
3	Project Feasibility and Evaluation
4	Risk , Resources and Costs
5	Team Dynamics and Organizational behavior
6	Alterative Models of Project Management: IT, Innovation, New Product Development
Coursework	
End Week 6	Team Report on SimProject sowing perofmace reports – submit via Turnitin Individual Reflective Diary on SimProject – submit via Turnitin
End Week 7 (reading week)	Individual Case Study on Project Management - submit via Turnitin

Assessment

This part of the course will be assessed entirely by coursework. Plagiarism will be taken extremely seriously and all assessments must be submitted via the turnitin plagiarism detection system.

Project Management Case Study	60%
Group Report on SimProject	20% - each team member gets the same mark.

Individual Reflective diary on SimProject	20%
---	-----

References

Primary Texts

1. [Project Management](#) - Clifford F. Gray & Erik W. Larson; McGraw-Hill, 4th Edition, ISBN 978-0-07-128751-7

Supplementary Texts

1. [Software Project Survival Guide \(Pro -- Best Practices\)](#)- Steve McConnell MICROSOFT PRESS (1 Oct 1997)
2. The Innovators Dilemma – Clayton M Christensen; Collins Business Essentials 2003
3. Winning at New Products: Accelerating the Process from Idea to Launch, by Robert G. Cooper

4E1 – Part II

COURSE TITLE: Management for Engineers		CODE: 4E1 Part 2
Electronic & Electrical Engineering		
LEVEL: Senior Sophister	CREDITS: 5 (pts 1+2)	PREREQUISITES: None
LECTURER: Dr. P. Scanlon Dept Electronic & Electrical Eng	TEACHING ASSISTANT:	
TERMS: Semester 1	LECTURES/WEEK: 2	TUTORIALS/WEEK:
DURATION (WEEKS): 5	TOTAL: 15 HRS	TOTAL:
AIMS/OBJECTIVES		
<p>The focus in this part of the course is on technology and society. The course uses the broad fields of signal/media processing, biomedical engineering and communications engineering as the bases for exploring issues relating to the design and development of technology in these fields. These are broad areas of research in the Dept. of EEE. In particular business, economic, social, ethical, political and other factors that affect the adoption/growth/decline of a technology are studied.</p>		
SYLLABUS		
<p>The course content cannot be described in a traditional itemized manner. However the course is well structured consisting of a number of well-defined core lectures which are supplemented by guest lectures that focus on topics of interest. Topics will include:</p> <ul style="list-style-type: none"> • an overview of some emerging biomedical, information and communications technologies; • case studies of technologies whose social impact is debatable; • a focus on some aspects technology that is more future-oriented (e.g. medical enhancement technology or e-health); • ethics in engineering – with special emphasis on social responsibility and engineering professionalism in technology development; • the circular relationship between society and technology – i.e. how technology affects society and how society drives technological advances. <p>It is anticipated that, as in previous years, there will be a number of guest lecturers. These are chosen from different walks of life to discuss the various topics of relevance from their perspectives. Previous guests have included: Denis O'Brien (entrepreneur), Sean O'Sullivan (Industrialist), Eamonn Ryan (Green Party), Simon Coveney (Fine Gael); Alexis Donnelly (TCD Lecturer), Richard Horton (ComReg), Brendan Tuohy (Dept. Communications); Jonah Brucker-Cohen (Art technologist), Katherine Moriwaki (Art technologist) and an Engineers Ireland representative.</p>		

LEARNING OUTCOMES

Upon completion of this course, students will be able to:

- analyse current technological trends from business, economic, social, ethical and political viewpoints;
- debate and discuss the implications of technology trends and choices;
- challenge and question the value and role of various technologies and developments;
- speculate and predict future trends in the communications, information technology and biomedical fields using arguments based not purely on technical issues;
- source material on technological issues that go beyond the technical details of the systems involved;
- express technical issues in terms suited to a wider non-technical audience.

TEACHING STRATEGIES

The teaching strategy for this course concentrates on trying to get students to think ‘outside the box’ and to express their thoughts in a comprehensive, clear and concise manner. It is important for them to realise that engineering decisions have consequences and impacts that go beyond technology implications. This is achieved by introducing the students to a wide range of material, through core lectures and invited guest speakers as well as by getting the students to participate in discussion, research and presentation where possible. Emphasis is placed on students forming and giving their opinions and on questioning issues and ideas. The overall philosophy of the course is to encourage as much interaction and debate as possible.

ASSESSMENT MODE(S)

Assessment is carried out completely by continuous assessment including various forms of assignments. Continuous assessment occurs throughout the course where the students are given specific tasks or assignments to which marks are attributed.

The distribution of marks between the various components of continuous assessment will be indicated by the principal lecturer throughout the course.

4E2 – Final Year Project – Electrical & Electronic Engineering Dept

COURSE TITLE: Final Year Project		CODE: 4E2
Electronic & Electrical Engineering		
LEVEL: Senior Sophister	CREDITS: 15	PREREQUISITES: None
CO-ORDINATOR: Prof. R. Reilly	TEACHING ASSISTANT:	
TERMS: Semester 1 and 2	LECTURES/WEEK:	TUTORIALS/WEEK:
DURATION (WEEKS): 22	TOTAL:	TOTAL:
AIMS/OBJECTIVES		
<p>As part of the fulfilment of the final year course of the BAI, the students in C and CD streams are required to carry out an individual engineering project. To this end, each student is assigned a project topic and supervisor who will guide the course of the project throughout the academic year.</p>		
SYLLABUS		
<p>Projects are allocated in areas of research expertise and interest of members of the academic staff in the Department. The project content is decided by the supervisor for each individual project. Project assignments are made to students on the basis of the choice forms filled out at the end of the Junior Sophister year. The nature and content of the project is then discussed by supervisor and student in the first week of Semester 1.</p>		
LEARNING OUTCOMES		
<p>On completion of their project work the student will be able to:</p>		
<ol style="list-style-type: none">1. Derive, apply and adapt solutions from the discipline specific knowledge gained in lectures and coursework, to a real world problem solving context.2. Experience independent enquiry and investigation of a practical engineering problem, application or topic.3. Assess and criticise information, methods and results for a defined engineering purpose.4. Identify and formulate technical problems in such a manner as to make them amenable to solution.5. Design a system, component or process to meet a specified goal6. Analyse and interpret results from experiments conducted during the course of the design process in order to modify improve or explain the functionality of the system, component or process being created.7. Communicate effectively in technical and scientific writing, and to present scientific/technical ideas concisely to a technical audience that may not be expert in the specific domain of the presentation.8. Formulate the design of systems in terms of a schedule of intermediate goals that manifest in subsystems.9. Manage workflow and task scheduling within the constraints of resources and time given specific design goals and deadlines.10. Use industry standard hardware and/or software tools and codes of practice for all aspects of design including analysis and presentation.		

11. Examine and discuss the impact of the project design or theme on society.

TEACHING STRATEGIES

There are no formal timetabled hours associated with the project but students are expected to spend the time it takes to make reasonable progress and to keep in regular contact with their supervisors. It is recommended that students make a formal arrangement with their supervisors to meet on a weekly basis, preferably at a regular appointed time which suits their lecture schedule.

ASSESSMENT MODE(S)

The Final Year Project is assessed by means of three submissions which will be marked by both the project supervisor and an assigned second reader. The combined submissions are marked out of a total of 150 marks. The submissions are as follows:

Project Summary: 10 marks

A 5-page summary outlining the nature of the project and the work carried out to date as well as a plan of work for its completion must be submitted at the end of week 7 of the first Semester.

Poster Session: 40 marks

It is intended that students complete work on their project by the end of week 6 of the second semester. During week 7 of this semester, when there are no scheduled lectures or tutorials students will prepare a poster on their project. A template will be provided for this purpose. Students display an A0 sized poster outlining the aims, technical content and accomplishments of their projects at a special session attended by supervisors, second readers and the external examiner. They give a short oral presentation at their poster and are then briefly questioned on it. This is then marked in confidence by the supervisor and second reader. This session will be scheduled close to week 8 of the second semester. The poster must be submitted electronically at the end of week 7 of the second semester.

Typed Report: 100 marks

A full typed report on the project is submitted, which should not exceed 50 pages. This report should be properly structured and typed in accordance with instructions given. The report should contain an introduction outlining in detail the aims and objectives of, as well as some background information on the project. The bulk of the report should discuss in detail the main technical work carried out by the student with appropriate results, explanations and deductions. A final conclusion should comment on the overall outcome of the project. This is then marked by the supervisor and second reader according to established guidelines. A copy of the project marking scheme used by the Department of Electronic & Electrical Engineering is included in the handbook. Two soft-bound copies of the report must be submitted by a date near the end of the second semester.

Submission Dates:

Exact dates for submission of, or attendance at, each these elements will be provided throughout the course of the year by the final year project co-ordinator.

Department of Electronic and Electrical Engineering

Summary - BAI FINAL YEAR PROJECT MARKING SHEET 2010/2011

STUDENT'S NAME:

TITLE OF PROJECT

READER'S NAME

ROLE: Supervisor _____ Second Reader _____ Third Reader _____

Interim Summary Mark: _____/10

Poster Session

1) **Oral Presentation**
.....Mark...../15

2) **Technical Content**
.....Mark...../15

3) **Visual Clarity**
.....Mark...../10

Total Poster Mark: /40

Project Report

- 1) **Introduction/Background**
..... Mark:/10
- 2) **Technical Content**
..... Mark:/40
- 3) **Conclusion**
..... Mark:/10
- 4) **Presentation**
..... Mark:/20
- 5) **Overall Achievement**
..... Mark:/20

Total Report
Mark _____/100

Reader's Marks:
Interim Summary _____/10
Poster Session _____/40
Project Report _____/100

Final Agreed Marks:
Interim Summary _____/10
Poster Session _____/40
Project Report _____/100

TOTAL _____/150

TOTAL _____/150

Signed: _____ **Date:** _____

4E2 – Final Year Project – Computer Science Dept

COURSE TITLE: Final Year Project		CODE: 4E2
Computer Science Department		
LEVEL: Senior Sophister	CREDITS: 15	PREREQUISITES: None
COORDINATOR: Dr. Hugh Gibbons	TEACHING ASSISTANT:	
TERMS: Semester 1 and 2	LECTURES/WEEK:	TUTORIALS/WEEK:
DURATION (WEEKS): 20	TOTAL:	TOTAL:
AIMS/OBJECTIVES As part of the requirements of the final year, students in Computer Engineering carry out an individual engineering project. Each student is assigned a project topic and supervisor who will guide the course of the project throughout the academic year. There are no formal timetabled hours associated with the project but students are expected to spend the time it takes to make reasonable progress and to keep in regular contact with their supervisors. It is recommended that students make a formal arrangement with their supervisors to meet on a weekly basis, preferably at a regular appointed time. Students may opt to have their projects supervised by lecturers in either the Department of Electronic and Electrical Engineering or in the Department of Computer Science, provided the chosen topic is judged suitable by the relevant departmental coordinators. In the Department of Computer Science, lecturers advertise projects and it is then up to the students to approach the relevant lecturer to make the necessary arrangements - students may also approach individual lecturers with their own project proposals.		
LEARNING OUTCOMES On completion of this course the student will be able to: <ul style="list-style-type: none">• derive, apply and adapt solutions from the discipline specific knowledge gained in lectures and coursework, to a real world problem solving context;• experience independent enquiry and investigation of a practical engineering problem, application or topic;• assess and criticise information, methods and results for a defined engineering purpose;• identify and formulate technical problems in such a manner as to make them amenable to solution;• design a system, component or process to meet a specified goal;• analyse and interpret results from experiments conducted during the course of the design process in order to modify improve or explain the functionality of the system, component or process being created;• communicate effectively in technical and scientific writing, and to present scientific/technical concisely to a technical audience that may not be expert in the specific domain of the presentation;• formulate the design of systems in terms of a schedule of intermediate goals that manifest in subsystems;• manage workflow and task scheduling within the constraints of resources and time given specific design goals and deadlines;• use industry standard hardware and/or software tools and codes of practice for all aspects of design including analysis and presentation;• examine and discuss the impact of the project design or theme on people and the environment.		
TEACHING STRATEGIES There are no formal timetabled hours associated with the project but students are expected to spend the time it takes to make reasonable progress and to keep in regular contact with their supervisors. It is recommended that students make a formal arrangement with their supervisors to meet on a weekly basis, preferably at a regular appointed time.		
ASSESSMENT MODE(S) The project is assessed on the basis of the final year report. The project supervisor and the second reader attend a project demonstration held early in the third (Trinity) term. The demonstration is informal and is not marked.		