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)

Junior Sophister Handbook

2011-2012

Introduction

Welcome to the Junior 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 the flexibility to adapt to the advances in electronic technology, computer systems and communications systems.

Academic Structure

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

UNIVERSITY OF DUBLIN | TRINITY COLLEGE

ACADEMIC YEAR STRUCTURE
2011 / 2012

Cal. Wk	Dates 2011/12 (week beginning)	Outline Structure of Academic Year 2011/12	Notes
1	29-Aug-11	Supplemental Examinations	**Statutory Term (Michaelmas) begins**
2	05-Sep-11		
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	Christmas Period (College closed from 23 December 2011 to 3 January 2012)	
18	26-Dec-11		
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	**← Statutory (Trinity) Term ends Friday 29 June 2012**
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**

4 July 2011

Courses

Courses undertaken by the Junior Sophister students are:

Faculty Courses		**Semester**
3E1	Engineering Mathematics V	I
3E1a	Engineering Analysis	I
3E4	Management for Engineers	II

EEE Dept Courses		**Semester**
3C1	Signals and Systems	I
3C2	Digital Circuits	I
3C3	Analogue Circuits	II
3C4	Applied Electromagnetism	II
3C5	Telecommunications	II
3C6(a&b)	Electronic Design Projects	I & II
3C7	Digital Systems Design	II
3E3	Probabilistic Methods	I

CS Dept Courses		**Semester**
3D1	Microprocessor Systems I	I
3D2	Microprocessor Systems II	II
3D3	Computer Networks	II
3D4	Operating Systems & Concurrent Systems	II
3D5(a&b)	Software Design and Implementation Project	I & II
CS2022	Computer Architecture II	II
ST2004	Applied Probability	I

C Stream

Semester 1

Module Title	**ECTS**	**Code**
Maths V / Engineering Analysis	5	3E1a
Probabilistic Methods	5	3E3
Signals and Systems	5	3C1
Digital Circuits	5	3C2
Microprocessors I	5	3D1
Design Project I	5	3C6a

Semester 2

Module Title	**ECTS**	**Code**
Mgt for Engineers	5	3E4
Digital Systems Design	5	3C7
Telecommunications	5	3C5
Analogue Circuits	5	3C3
Applied Electromagnetism	5	3C4
Design Project II	5	3C6b

D Stream

Semester 1

Module Title	**ECTS**	**Code**
Maths V / Engineering Analysis	5	3E1a
Applied Probability	5	ST2004
Signals and Systems	5	3C1
Digital Circuits	5	3C2
Microprocessors I	5	3D1
Design Project I	5	3D5a

Semester 2

Module Title	ECTS	Code
Mgt for Engineers	5	3E4
Computer Architecture II	5	CS2022
Computer Networks	5	3D3
Operating Systems & Concurrent Systems	5	3D4
Microprocessors II	5	3D2
Design Project II	5	3D5b

CD Stream

Semester 1

Module Title	ECTS	Code
Maths V / Engineering Analysis	5	3E1a
Probabilistic Methods OR	5	3E3
Applied Probability		ST2004
Signals and Systems	5	3C1
Digital Circuits	5	3C2
Microprocessors I	5	3D1
Design Project I	5	3D5a

Semester 2

Module Title	ECTS	Code
Mgt for Engineers	5	3E4
Digital Systems Design OR	5	3C7
Computer Architecture II		CS2022
Telecommunications	5	3C5
Computer Networks	5	3D3
Microprocessors II	5	3D2
Design Project II	5	3C6b

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 in 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 or fraud.

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.

Contribution of Junior Sophister Year to BAI Degree

Students should note that the overall average mark obtained at the Junior Sophister Annual Examinations will contribute 20% of the overall result of the BAI degree and grade obtained.

BAI Examination Rules

Candidates undertake 60 credits during each of the four years of the degree programme. Each module has an individual rating of 5, 10, 15 or 20 credits, the amount dependent on the level of effort involved. It is the responsibility of each student to ensure that they are undertaking exactly 60 credits per year.

Students who pass the ANNUAL examinations are awarded 60 and an Honors grade for the year. This grade is based on the weighted average achieved, calculated using the credit ratings. In order to pass the ANNUAL examinations, students must:

- have achieved at least 40% in individual modules worth at least 50 ECTS credits **AND**
- have an overall average mark of at least 40% **AND**
- have **EITHER**

not more than 10 module credits with marks of at least 35% and less than 40%

OR

not more than 5 module credits with marks of at least 30% and less than 40%.

Students who have failed the ANNUAL examination are required to take a SUPPLEMENTAL examination in all modules in which they have not satisfied the examiners, as specified on the published examination results.

In order to pass the SUPPLEMENTAL examinations, students must:

- have an overall combined average mark in the ANNUAL and SUPPLEMENTAL examinations taken of not less than 40% **AND**
- have not more than 5 ECTS module credits with marks of at least 35% and less than 40%.

It should be noted that students are not permitted to repeat courses that are assessed by coursework only for supplemental examinations. It will not be possible to repeat missed or failed laboratories or assignments either and marks awarded for these will be carried forward to the supplemental examination results.

Students who pass the SUPPLEMENTAL examinations obtain an overall **PASS** grade for the year. Overall supplemental marks for all modules are calculated in the same manner using the same weightings as for the annual examinations and include continuous assessment/laboratory marks.

The full set of overall grades is set out below;

Description	***Grade***	***Criterion***
First Class Honors	**I**	mark greater than or equal to 70%
Second Class Honors, First Division	**II.1**	mark greater than or equal to 60% and less than 70%
Second Class Honors, Second Division	**II.2**	mark greater than or equal to 50% and less than 60%
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 or the candidate has not passed the year within eighteen months from that date on which they first became eligible or 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

		Applies to Erasmus / International Exchange students
ERASMUS Awaiting Result	**ER**	
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
Repeat year	**R**	the candidates is given permission to repeat the year IN FULL (applies at SUPPLEMENTAL examinations ONLY)
Pass	**P**	the candidate may rise to the next year of the degree programme (applies at SUPPLEMENTAL examinations ONLY)

After the Court of Examiners' meeting, ANNUAL and SUPPLEMENTAL examination results are published anonymously in student number order.

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/

Where a mark is not reported for a module the following codes apply where appropriate:

f	=	mark is less than 25%;
a	=	absent with permission/explained absence – 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 e.g. candidate is exempt on the basis of their performance in the Foundation Scholarship examination;
gw	=	grade withheld (e.g. unpaid fees or fines).
p	=	credit for subject passed on previous occasion.

Repeating the year

Candidates must repeat the year IN FULL which includes all continuous assessment requirements and laboratory experiments.

GUIDELINES AND REGULATIONS FOR B.A.I. STUDENTS UNDERTAKING ERASMUS YEAR OUT PROGRAMMES

This document provides guidelines and regulations for students who spend their Junior Sophister year of the B.A.I. programme at an approved foreign host University. Agreements are currently in place with INSA Lyon (France) and Karlsruhe University (Germany).

Students must obtain permission to spend their JS year at another University from the International Student Coordinator of the Department responsible for the B.A.I. stream in which they intend to specialise. These applications will then be reviewed by the Director of Teaching and Learning (Undergraduate) and/or the Head of School for final approval and decisions will then be communicated to the International Office who will inform students. At present, these Departmental Erasmus Coordinators are as follows:

- *Civil, Structural and Environmental Engineering:* Dr Sara Pavia (pavias@tcd.ie)
- *Computer Science:* Dr Carl Vogel (carl.vogel@scss.tcd.ie)
- *Electronic and Electrical Engineering*: Dr Anthony Quinn (anthony.quinn@tcd.ie)
- *Mechanical and Manufacturing Engineering*: Professor Henry Rice (hrice@tcd.ie)

Students must obtain at least a II.1 in their SF year in order to be given permission to spend their JS year abroad and must have appropriate language competency for their host University.

Each student must undertake modules that have a combined rating of at least 45 ECTS of which at least 40 ECTS must be in approved technically based engineering modules. Each student must get their module choices approved by their Departmental Erasmus Coordinator.

Students should be aware that some host universities (typically in Germany) do not return marks using a centralised administrative system. In such cases, students need to take responsibility for obtaining their marks for each module directly from their lecturers on official College letterhead. These must then be returned to their Departmental Erasmus Coordinator as quickly as possible.

Students must complete the year at the host university and have no entitlement to take supplemental exams at TCD. Students should be aware that some host universities do not have supplemental exams or may not allow students to sit supplementals if their attendance or performance has been poor - students are advised to monitor the course information at their host University frequently.

Assessment of modules taken in the overseas university will be weighted in the calculation of the final degree results as if the modules had been taken in this university.

MARKING SCHEMES

Firstly, the grades obtained are converted into TCD equivalents as follows:

INSA (Lyon)

ECTS mark returned	TCD equivalent
A	80%
B	65%
C	60%

D	**55%**
E	**45%**
Fx	**30%**
F	**20%**

Karlsruhe University

The marks obtained from Karlsruhe are based on the German system which grades subjects from 1.0 (very good) to 5.0 (NOT adequate). Grades are converted into TCD equivalents using the following formula:

$$TCD_{MARK} = (5 - Karlsruhe_{MARK}) / 4 * 100$$

Pass Criteria

In order to pass the JS year, students must:

- acquire 45 credits for modules at the host University, of which 40 credits must be in approved technical engineering modules;
- each student must submit an interim and a final report on their experience to their International Student Coordinator to acquire an additional 15 credits giving a total of 60 credits for the year.

Junior Sophister Third Year Courses

3C1 – Signals and Systems

COURSE TITLE: SIGNALS AND SYSTEMS		**CODE: 3C1**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): DR. DAVID CORRIGAN		
TERMS: Semester 1 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

Signals and Systems is a one semester course taken by Junior Sophister C, CD and D Stream students. It provides a foundation for Signal Processing and Communications Engineering topics covered later in the undergraduate curriculum.

The course contains elements of Signal Processing as a foundation for courses in DSP and Communications later on in the curriculum. *Signal Processing* is the study of the process of information extraction from signals such as images, audio, text or measurement data. The course introduces the student to methods for manipulating signals such as filters for high frequency noise removal. It introduces frequency analysis of signals and filters as well as the notion of system stability. Both analogue and digital processing is considered. The course also contains an introduction to Control Systems that formalises the notion of feedback in dynamic systems. It shows how the use of feedback loops can be used to regulate the output of a dynamic system in the presence of both reference and disturbance inputs. An example is a cruise control system in cars, where a dynamic system is used to regulate the speed of the car which can be affected by gusts of winds, changes in gradient etc.

SYLLABUS

Systems Analysis

- Use of block diagrams, Differential Equation Models, What is a
- Linear Time invariant System?
- Impulse response, convolution, step response, Laplace Transforms,
- Transfer functions.
- Poles, Zeros, Stability
- Frequency response, Steady state response, lowpass and highpass filtering.

Signal Analysis and Digital Signal Processing

- Fourier series, Fourier Transform, Parseval's Theorem
- Sampling theorem, Difference Equations and The z-Transform
- Low pass filtering, low pass filters, Basics of FIR, IIR Filters
- Stability and Applications of Digital Filters

Introduction to Control Systems

- The control system – (i) Feedback, major and minor feedback loops. Reference and Disturbance inputs.
- Steady State Error Analysis - (i) System type, (ii) Reference Inputs for analysis. (iii) Error Analysis using the Final Value Theorem.
- Step Response Criteria – relationship between responses and 2^{nd} Order pole locations.
- Introduction to root locus and compensators.

ASSOCIATED LABORATORY/PROJECT PROGRAMME

Lab S1: Harmonic Signal Analysis with the Fourier Transform.

Lab S2: Interaction of Signals with Systems and Primitive Speech Recognition

Note: Properly structured laboratory reports must be written up after each laboratory and submitted for marking.

RECOMMENDED TEXT(S)

- SIGNALS AND SYSTEMS, Oppenheim and Willsky, Prentice Hall,
- ELECTRONIC SIGNALS AND SYSTEMS, Paul A. Lyn, Macmillan Education
- MODERN CONTROL SYSTEMS, Dorf and Bishop, Addison Wesley
- SYSTEM ANALYSIS AND SIGNAL PROCESSING, Philip Denbigh, Addison-Wesley

LEARNING OUTCOMES

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

- Analyse systems in order to calculate, estimate and classify their impulse, step, frequency response and evaluate their stability
- Analyse signals in order to calculate their frequency spectra, and estimate, classify, assess the effect of a system on signals in terms of frequency content and time domain effects
- Apply difference equations and the Z-Transform in calculating the output of a digital system given any digital input
- Analyse the effect of the gain parameter of a closed-loop system on the stability of the system and the steady state error.

TEACHING STRATEGIES

The course is taught using a combination of lectures, tutorials and two supporting laboratories. During the tutorials students are guided through the solution of problems based on the lecture material. Students are expected to complete the majority of the tutorials outside of the scheduled contact hours.

ASSESSMENT MODE(S)

The written examination will contribute 85% and the laboratories will contribute 15% of the overall subject mark at the Annual and Supplemental examinations.

Note: Laboratories not completed during the teaching semesters cannot be repeated for supplemental examinations and existing marks will be carried forward to the supplementals.

3C2 – Digital Circuits

COURSE TITLE: Digital Circuits		**CODE: 3C2**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Dr. M. J. Burke	**TEACHING ASSISTANT:**	
TERMS: Semester 1 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

Digital Circuits is a one semester course taken by Junior Sophister C, CD and D Stream students. It provides a thorough foundation in digital circuits as applied to modern logic device families. The course aims to provide students with knowledge of the operational principles and practical limitations of digital circuits at device and circuit level, as well as instructing them in the analysis and design of these circuits. All of the principles and techniques learned are applicable to the design of digital circuits on a wider scale. During the course, students will develop the analytical and synthesis skills needed to design digital circuits for electronic equipment intended for any modern application area. In particular C Stream Electronic Engineering students will use these skills later in further integrated circuit design courses, while CD Stream Electronic and Computer Engineering and D Stream Computer Engineering students gain the insight needed to appreciate how the design of digital circuits influences and ultimately limits the performance of computers at gate and system level.

SYLLABUS

Semiconductor Materials: revision of fundamental laws; carrier transport phenomena; current flow mechanisms; the p-n junction; barrier potential; the ideal diode equation; bipolar diode as a logic element.

Bipolar Junction Transistor: physical principles of operation; device characteristics and parameters.

Bipolar Transistor Inverter: operation of the BJT transistor as a switch; simple inverter circuit; static and dynamic performance characteristics; effects of loading.

BJT Inverter Applications: the design of simple bipolar transistor circuits to act as buffers, drivers and interfaces in a range of applications.

TTL Logic Family: logic characteristics and performance; operating principles of standard 7400 series gates; circuit analysis and power consumption evaluation.

MOS Field Effect Transistor: physical principles of operation; device characteristics and parameters.

MOS Transistor Inverter: operation of the MOS transistor as switch; simple inverter circuit; static and dynamic performance characteristics; effects of loading.

CMOS Logic Family: Simple Logic functions in CMOS

ASSOCIATED LABORATORY/PROJECT PROGRAMME

Lab D1: Bipolar Junction Transistor: the assessment of transistor switching characteristics.

Lab D2: MOS C–V Characterisation: the characterisation of the CMOS technology process.

Note: Properly structured laboratory reports must be written up after each laboratory and submitted for marking.General instructions for laboratories will be issued.

RECOMMENDED TEXT(S)

1. Streetman B.G. & Banerjee S., *Solid State Electronic Devices*, 5th ed., Prentice-Hall, 2000.

2. Hodges D. A. & Jackson H. G., *Analysis & Design of Digital Integrated Circuits*, 2nd ed. McGraw-Hill; 1988.

3. Kang S. & Leblebici Y., *CMOS Digital Integrated Circuits*, McGraw-Hil; 1996.

LEARNING OUTCOMES

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

1. Explain the operation of the bipolar junction and MOS field effect transistors and associated logic gates.
2. Analyse simple single-transistor switching circuits to determine their performance criteria and limitations.
3. Analyse simple single transistor switching circuits to determine static and dynamic performance parameters and related figures of merit.
4. Design simple transistor circuits for applications such as: LED drivers, logic level shifting, current buffering, relay switching etc. working from a performance specification.
5. To carry out circuit analysis experiments using CAD tools such as PSpice in a systematic and disciplined manner.

TEACHING STRATEGIES

The course is taught using a combination of lectures, tutorials and two 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 85% and the laboratories will contribute 15% of the overall subject mark at the Annual and Supplemental examinations.

Note: Laboratories not completed during the teaching semesters cannot be repeated for supplemental examinations and annual marks will be carried forward to the supplemental examinations

3C3 – Analogue Circuits

COURSE TITLE: Analogue Circuits		**CODE: 3C3**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Dr E C Lalor	**TEACHING ASSISTANT:**	
TERMS: Semester 2 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

Analogue Electronics is a one semester course taken by Junior Sophister C Stream students. It provides a thorough foundation in analogue circuits as applied to systems used in generating, amplifying and in general processing signals which are continuous functions of time. The course aims to provide students with knowledge of operational principles and practical limitations of analogue circuits at device and circuit level, as well as instructing them in the analysis and design of these circuits. All of the principles and techniques learned are applicable to the design of analogue systems on a wider scale. During the course, students will develop the analytical and synthesis skills needed to design analogue circuits for electronic equipment intended for any modern application area. Electronic Engineering students will use these skills later in further integrated circuit design courses.

SYLLABUS

- BJT Amplifier Principles: Biasing, Small signal analysis (hybrid-π), Single-stage BJT amplifiers, Common Emitter, Common Collector
- Single-Stage Integrated-Circuit Amplifiers: Comparison of MOSFET and BJT, Common Source, Common Base, Darlington Configuration
- Differential and Multistage Amplifiers: MOS Differential Pair, Common-Mode Rejection Ratio, BJT Differential Pair, Nonideal Characteristics of Differential Amplifier, Two-stage CMOS Op Amp.
- Feedback: Negative Feedback, Feedback Amplifiers
- Operational Amplifiers: 741 Op-Amp circuit, DC and small-signal analysis of the 741, Gain, Frequency Response and Slew Rate of the 741.
- Data Converters: D/A and A/D Converters, Sigma-Delta Modulation
- Oscillators: Basic principles, Op Amp-RC Oscillator Circuits
- Output Stages and Power Amplifiers: Class A, Class B and Class AB Output Stages, IC Power Amplifiers

ASSOCIATED LABORATORY/PROJECT PROGRAMME

Lab A1: Active Filters: PSpice simulation of an operational amplifier based analogue active filter

Lab A1: Wien Bridge Oscillator: the assessment of oscillation

characteristics, purity of sine-wave, stability, frequency etc.

Note: Properly structured laboratory reports must be written up after each laboratory and submitted for marking.

RECOMMENDED TEXT(S)

1. Microelectronic Circuits (5^{th} Ed) Sedra & Smith, Oxford University Press, 2004

2. Microelectronic Circuit Design (2^{nd} Ed) Jaeger & Blalock, McGraw Hill, 2004

3. Electronic Devices: Conventional Current Version (8^{th} Ed) Floyd, Pearson, 2008

LEARNING OUTCOMES

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

6. Explain the operation of the BJT and MOSFET amplifiers.
7. Analyze simple linear amplifiers to determine their performance criteria and limitations.
8. Explain Op-Amp ideal and practical characteristics
9. Explain the Principles of Oscillation.
10. Design amplifiers and oscillators based on the performance criteria.
11. Explain the operation of D/A and A/D converters.

TEACHING STRATEGIES

The course is taught using a combination of lectures, tutorials and two 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 85% and the laboratories will contribute 15% of the overall subject mark at the Annual and Supplemental examinations.

Note: Laboratories not completed during the teaching semesters cannot be repeated for supplemental examinations and existing marks will be carried forward to the supplementals.

3C4 – Applied Electromagnetism

COURSE TITLE: Applied Electromagnetism		**CODE: 3C4**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Prof. W. T. Coffey	**TEACHING ASSISTANT:**	
TERMS: Semester 2 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

Electromagnetism is a one semester course taken by Junior Sophister C stream students. The course first deals with the physical principles of electromagnetism and electromagnetic waves.

SYLLABUS

Starting from Coulomb's law of force between charges, this course develops the basic equations of electrostatics including Laplace's and Poisson's equations for the electrostatic potentials in space.
Following this, the situation where the charges are allowed to move is considered and students then discuss the continuity equation for the current.
Next the concept of electromotive force is developed. Magnetism is then introduced via the Biot-Savart law, and Amperes theorem is proven. Following this, the mathematical statements of Faraday's and Lenz's laws are given and it is explained how Maxwell modified Amperes theorem to account for the observed current in a capacitive circuit. Thence the six Maxwell equations in both differential and integral form are given. Whence we proceed to Poyntings's theorem and give an elementary account of the Hertzian dipole.
Experiments which attempted to demonstrate the existence of the aether are recounted and an elementary account of the Lorentz transformations and special relativity is given. Maxwell's equations in a medium are briefly treated and the analogy with a transmission line is explained. The telegraph equation for the propagation of the voltage and current along a transmission line is derived and discussed in detail and the concepts of characteristic impedance and propagation coefficient are comprehensively treated.
The foregoing concepts are then used to provide a brief introduction to resonant cavities and waveguides. Finally, we discuss Planck's black body radiation law along with the photoelectric effect, the de Broglie hypothesis. These are then used to illustrate energy quantisation in matter via the Schrödinger equation for a particle in a 1-D box and how the solution of this equation for a periodic potential leads to the band structure of semiconductor materials.

ASSOCIATED LABORATORY/PROJECT PROGRAMME

Lab E1: Transmission Lines: The impedance and velocity of propagation are determined.

Lab E2: Introduction to Microwaves:

Note: Properly structured laboratory reports must be written up after each laboratory and submitted for marking.

RECOMMENDED TEXT(S)

1. Fundamentals of Electric Waves, H. H. Skilling, McGraw Hill, New York, 1948 (Reprinted by Krieger, Malabar, Florida 1974)
2. Classical Electricity and Magnetism, W. Panofsky and M. B. Phillips, Addison Wesley, New York, 1962
3. Electromagnetic Fields and Waves, Paul Lorrain, Dale P. Corson, and Francois Lorraine, 3rd Ed. W. H. Freeman and Co. New York, 1987
4. The Mathematical Theory of Electricity and Magnetism, J. H. Jeans, Cambridge University Press London, 4th Ed. 1966
5. Field and Wave Electromagnetics, David K. Cheng, 2nd Ed. Addison Wesley, New York, 1989

6. Electrodynamics: An Introduction Including Quantum Effects, H. J. W. Mueller-Kirsten, World Scientific, Singapore, 2004

LEARNING OUTCOMES

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

1. understand the basic equations of electrostatics and magnetostatics
2. understand the essential modifications made to these equations by Maxwell in order to obtain a wave theory of light
3. understand why the concept of an aether is unnecessary
4. understand the basic principles of electromagnetic radiation, i.e. the Hertzian dipole
5. have a working knowledge of Maxwell's equations in a medium and waves on a transmission line
6. describe the operation and characteristics of simple optoelectronic devices.

TEACHING STRATEGIES

The course is taught using a combination of lectures, tutorials and two supporting laboratories.

Course Webpages:

http://www.mee.tcd.ie/~wcoffey/teaching

http://www.tcd.ie/engineering/undergraduate

ASSESSMENT MODE(S)

The written examination will contribute 85% and the laboratories will contribute 15% of the overall subject mark at the Annual and Supplemental examinations.

Note: Laboratories not completed during the teaching semesters cannot be repeated for supplemental examinations and existing marks will be carried forward to the supplementals.

3C5 – Telecommunications

COURSE TITLE: Telecommunications		**CODE: 3C5**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): **Dr Nicola Marchetti**	**TEACHING ASSISTANT:**	
TERMS: Semester 2 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

The aim of the course is to introduce students to the key aspects of a digital radio system and the fundamental principles that underpin these systems. It is intended that students would become familiar with aspects of compression, coding and modulation (i.e. the key building blocks of a radio) as well aspects of information theory that give insights on performance of the radio.

SYLLABUS

Introduction to Random Processes (moments and autocorrelation, power spectral density and related theorems, white noise and its autocorrelation and spectrum)
Introduction to Telecommunication Systems (basics of major telecom systems, e.g. fiber optics and Internet)
Introduction to wireless systems (basics of radio and ideas from radio wave propagation)
Introduction to Information Theory
Lossless Compression (e.g. Huffman and LZW)
Coding Techniques (e.g. Syndrome Coding, Hamming Coding)
Analogue Modulation (eg. AM and FM)
Digital Modulation (e.g. PSK and QAM) and BER performance
Performance tradeoffs of the radio - Power, Bandwidth and Noise Performance

ASSOCIATED LABORATORY/PROJECT PROGRAMME

T1 Analogue Modulation

T2 Digital Modulation

RECOMMENDED TEXT(S)

1. B. Sklar, Digital Communications, Fundamentals and Applications, Prentice Hall 2001
2. T. Cover, Elements of Information Theory, Wiley, 2006
3. S. Haykin, Communication Systems, Wiley 2001
4. D. Mackay, Information Theory, Inference, and Learning Algorithms Published by Cambridge University Press (2003) (Published for Free Online)

LEARNING OUTCOMES

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

1. Understand key concepts of random processes
2. Understand the basics of telecommunication systems, and in particular of wireless systems

3. Understand key concepts in information theory (e.g. entropy, capacity of channels)

4. Apply different compression techniques to data and explain advantages and disadvantages of the different options.

5. Apply different coding techniques to data and demonstrate the types of errors which can be corrected.

6. Describe and explain a number of analogue modulation schemes and calculate bandwidth and power consumption of the different schemes.

7. Describe and explain a number of digital modulation schemes and calculate BER performance under different conditions.

8. Explain the trade-offs and choices that can be made in the design of digital radios.

TEACHING STRATEGIES

The course is taught using a combination of lectures and tutorials.

ASSESSMENT MODE(S)

The written examination will contribute 85% and the laboratories will contribute 15% of the overall subject mark at the Annual and Supplemental examinations.

Note: Laboratories not completed during the teaching semesters cannot be repeated for supplemental examinations and existing marks will be carried forward to the supplementals.

3C6a – Electronic Design Project: Analogue

COURSE TITLE: Electronic Design Project: Analogue		**CODE: 3C6a**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Dr. Ed Lalor.	**TEACHING SUPPORT:**	**MR. R. DEMPSEY**
TERMS: Semester 1 **DURATION (WEEKS): 11**	**LABORATORIES: 10** **TOTAL: 10**	**TUTORIALS :0** **TOTAL: 0**

AIMS/OBJECTIVES

The analogue design project aims to develop the students' practical knowledge of the design, simulation, implementation, and testing of analogue electronic circuits. Students learn to work in a group of usually four persons, who must manage the project, divide up the workload between themselves by applying a 'divide-and-conquer' approach to tackling a design task. The project involves constructing useful circuit modules from basic components while simulating, testing, and documenting each section.

SYLLABUS

- Using the transistor as a simple common emitter amplifier.
- Simulating small amplifier circuits with PSpice.
- How to design a PCB – a printed circuit board.
- Using an i.c. LM386 audio amplifier to amplify the output from a personal stereo or MP3 player.
- One or two proposed larger projects are:
- Constructing an instrumentation amplifier for an ECG signal.
- Sampling an audio signal using A-to-D conversion

RECOMMENDED TEXT(S)

1. Storey, Neil, *Electronics A Systems Approach.* , (4th ed.) Pearson.
2. Horowitz, P.& Hill, W. ***The Art of Electronics*** (2nd ed.). C.U.P.
3. Floyd, Thomas., *Electronic Devices (*5th ed. 1996), (8th ed. 2008)

LEARNING OUTCOMES

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

- Design and plan a project involving analogue electronics
- Construct a hardware solution for an analogue electronics problem
- Sketch a schematic diagram of the circuit with chosen component values

- Select a definite test strategy to check each stage of the design
- Carry out a test and verification procedure, recording appropriate results
- Write a structured and comprehensive technical report on the project
- Work as part of a team

TEACHING STRATEGIES

Students learn progressively more difficult design methods and tools as the weeks progress. The project is launched from introductory laboratory exercises with transistors and which include both construction of circuits and PSpice simulations. Support is on hand from the demonstrator and a technical officer throughout the project.

The essential steps in the design procedure are explained and illustrated. In the first few weeks, the students are heavily assisted to design, simulate, construct, solder together and test a simple introductory audio amplifier. They see the important role of calculations in circuit design and the problems associated with testing the design by making use of PSpice. Students are then required to research, and construct a more complex amplifier e.g, to amplify an ECG signal. This will require knowledge of instrumentation amplifiers and how they are configured from three op-amps. The performance of the simpler circuits will show their limitations and reveal the problems that need to be overcome.

ASSESSMENT MODE(S)

The written report will constitute 60% and the practical in-lab work will contribute 40% of the overall project mark.

Note: While this is a group project, each student must submit an individual report.

3C6b – Electronic Design Project: Digital

COURSE TITLE: Electronic Design Project: Digital		**CODE: 3C6b**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Dr. M. J. Burke	**TEACHING SUPPORT:**	**MR. R. DEMPSEY**
TERMS: Semester 2 **DURATION (WEEKS): 12**	**LABORATORIES: 10** **TOTAL: 10**	**TUTORIALS :0** **TOTAL: 0**

AIMS/OBJECTIVES

The main purposes of the project are:

- To develop the student's practical knowledge of digital logic gates, displays, and the use of microprocessors, such as the PIC and Arduino
- To gain further experience in the design, simulation, implementation, and testing of digital circuits
- To develop the ability to work on a project as a member of a team.

SYLLABUS

- Fundamental building blocks of digital circuits from gates to system level devices.
- Frequently used important blocks like decoders, multiplexors, flip-flops, shift registers, counters and timers.
- Use of block diagrams, circuit schematics with PSpice, circuit simulation and testing.
- Use of programmable logic (PIC) to simplify or modify designs and reduce cost.
- Analysis and design of combinational and synchronous digital systems.
- Maintaining good engineering documentation.

RECOMMENDED TEXTS

1: Katz, R. & Boriello, G., *Contemporary Logic Design*, (2nd ed. 2005) Pearson Education.

2. Hodges D. A. & Jackson H. G., *Analysis & Design of Digital Integrated Circuits*, (2nd ed. 1988) McGraw-Hill;

LEARNING OUTCOMES

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

- Design and plan a project involving digital electronics
- Construct a hardware solution for a digital electronics problem
- Sketch a block diagram of the circuit along with user interfaces
- Select a definite test strategy to check each stage of the design
- Obtain and describe timing waveforms
- Write a structured comprehensive technical report on the project
- Work as part of a team

TEACHING STRATEGIES

The hardware construction of two real working circuits is required – one introductory project, and one far more challenging circuit. For example, building a decoder and display for the time-code MSF signal broadcast from Cumbria. The project is launched from introductory laboratory exercises with CMOS ICs which include both construction and Spice simulations. Support is on hand from the demonstrator and a technical officer throughout the project.

ASSESSMENT MODE(S)

The written report will constitute 60% and the in-lab practical work will contribute 40% of the overall project mark.

Note: While this is a group project, each student must submit an individual report.

3C7 – Digital Systems Design

COURSE TITLE: Integrated Systems Design		**CODE: 3C7**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Dr. Naomi Harte	**TEACHING ASSISTANT:**	
TERMS: Semester 1 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 2** **TOTAL: 22**	**PRACTICALS/WEEK: 2** **TOTAL: 22**

AIMS/OBJECTIVES

This course introduces digital systems design with a focus on FPGA design. The course will build on the basics of digital circuits and logic from 2nd and 3rd year, with an in-depth study of combinatorial and sequential hardware systems and the use of finite state machines in the design of sequential systems. The students will learn how a Hardware Description Language (HDL) is used to describe and implement hardware. The emphasis is not on the details and syntax of the language, but rather how the language infers hardware. They will see how to simulate and test that hardware and optimise their designs. They will learn about the use of FPGAs in digital design and the full FPGA design flow. This will be presented in the context of other design platforms such as ASIC, DSP or GPU and the trade-offs involved.

SYLLABUS

Students will need to re-familiarise themselves with computer arithmetic from 2nd year.
Topics studied in 3C7:

- In-depth study of combinatorial and sequential logic and finite state machines.
- Digital design flows and design trade-offs
- FPGA structure and design flow
- Verilog HDL language
- Modelsim simulation environment
- Testbench construction
- Realisation of all above concepts in hardware designs

ASSOCIATED LABORATORY/PROJECT PROGRAMME

None

RECOMMENDED TEXT(S)

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. FPGA Prototyping By Verilog Examples: Xilinx Spartan-3 Version, Pong P. Chu (Wiley)
3. Verilog HDL, 2/e Palnitkar (reference only)

Supplementary Reading may be specified during the course.

LEARNING OUTCOMES

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

1. Discriminate between combinatorial and sequential circuits

2. Design state machines to control complex systems

3. Define and describe digital design flows for system design and recognise the trade-offs involved in different approaches

4. Write synthesisable Verilog code

5. Write a Verilog testbench to test Verilog modules

6. Analyse code coverage of a Verilog testbench

7. Target a Verilog design to an FPGA board

8. Analyse and debug Verilog modules

9. Build a synchronous DSP system in Verilog and verify its performance

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-cum-lab, where the lecturer will talk about the content of the session and the student will “learn by doing”. The FPGA board used to support the practical sessions is the Spartan-3 starter board. The practical sessions will require the students to complete the weekly assignment outside class hours (average 4-6 hours extra per assignment, 9 assignments in total), 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. The practical element of the course **CANNOT** be repeated. If the subject if failed at the annual examinations, the mark for the practical component will be carried forward to the supplemental exams.

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.

<u>Late assignments policy</u>:

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

3E3 – Probabilistic Methods

COURSE TITLE: Probability Modelling		**CODE: 3E3**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: SF**
LECTURER(S): Dr. A. Quinn	**TEACHING ASSISTANT:**	
TERMS: Semester 1 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

This course addresses the practical role of probability in electronic and computer engineering, and springs from the fundamental need to cope with uncertainty in engineering practice, notably in the analysis of experiments, and the interpretation and processing of data. The keystone of the course will be a philosophical one. The relationship between uncertainty and information will be explored from the start, with the probability calculus acting as a consistent framework for quantifying and manipulating belief in uncertain propositions (the Bayesian perspective). In this way, the foundation of the course will be a unified one, with information, uncertainty, errors, observation and noise all embraced within a Bayesian notion of probability.

The course will not separate this theory from the engineering practice it is designed to serve. Telecommunications will provide key examples: the digital source, the additive noise channel, the binary symmetric channel, etc. The analysis of queueing and arrival phenomena will also be discussed, as will the practical context of radiation and fluorescence phenomena such as underlie FLIM imagery. By delivering the key insight that relates probability and information, the course sets the scene for Shannon's quantification of information (information theory), which follows in the second semester (3C5).

A second principal aim is to reconcile probability modelling with sampling statistics. Typically a black-spot in the formation of the engineering student, statistical inference will be re-cast as a problem of probability modelling, using the device of the empirical distribution. Two themes of practical engineering interest will be discussed, i.e. the quantification of (i) measurement error, and (ii) reliability of measured systems.

Graphical models will appeal to engineering students as an intuitive way of expressing conditional independence between multiple random variables. Correlation will be explained as the vital resource for predicting uncertain phenomena, setting a secure but intuitive foundation for random processes in the second semester (3C5), and topics such as linear prediction in the final year (4C5).

3E3 will be taught in parallel with 3C1 (Signals and Systems), and these will support the delivery of Telecommunications (3C5) in the 2nd semester of the Junior Sophister year. In turn, they will provide the basis for delivery of Senior Sophister courses such as DSP (4C5) and digital communications (4C7).

In summary, this course on probability is taught from the perspective of the ICT practitioner. It is an invitation to the student to confront uncertainty as the fundamental phenomenon/resource in ICT systems, and to appreciate probability as a framework for the design, analysis and optimization of such systems.

SYLLABUS

- **Review of Propositional Logic**
  Uncertain experiments in electrical and computer engineering

  Sample space and events

  Necessity, sufficiency, mutual exclusivity

- **The Foundation of Probability Modelling**
  The axioms of probability and the probability triple

  Independence and conditional probability

  Key relationships: chain rule, theorem of total probability, Bayes' rule

- **Sequential Experiments**
  Independent sequential experiments: binomial and multinomial probability laws (application in repetition coding of digital sources)

  Homogeneous Markov chains

- **Univariate Random variables**
  The key probability functions (cdf, pdf, pmf)

  Key discrete probability models (Bernoulli→Binomial→Poisson)

  Key continuous probability models (rectangular, exponential, m-Erlang, normal)

  Functions of random variables: application in quantization

  Expectation

- **Multiple random variables**
  Marginal and conditional distributions

  Discrete-continuous case: finite mixture models

  The bivariate normal distribution

  Correlation

  Introduction to graphical models

- **Sampling Statistics**
  The empirical distribution and its moments (sampling statistics)

  Quantification of error and quality based on sampling

RECOMMENDED TEXT(S)

The main recommended text for the course is:

1. Leon-Garcia, A., *Probability, Statistics, and Random Processes for Electrical Engineering,* 3rd ed., Prentice Hall, 2008.

Secondary recommended texts are as follows:

2. Bertsekas, D.P. and Tsitsiklis, J.N., *Introduction to Probability,* 2nd ed., Athena Scientific Press, 2008.
3. Applebaum, D., *Probability and Information,* 2nd ed., Camb Univ Press, 2008.

LEARNING OUTCOMES

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

- Quantify beliefs in uncertain propositions related to key ICT contexts, such as coding, noisy communication, quantization and queueing
- Distinguish the essential notions of independence and dependence, and relate the latter to the idea of prediction
- Apply and analyze the key parametric probability models (distributions) governing uncertainty in these ICT contexts
- Evaluate measures of location, spread and dependence (moments) for these distributions
- Convert random experimental data (samples) into quantified beliefs via the empirical distribution, and summarize them via sampling statistics

TEACHING STRATEGIES

There is a 3:1 ratio between lectures and tutorials. The notes are provided after each lecture, via scans uploaded to the webpage. Problem-solving experience is vital, and gained primarily through the weekly tutorials, but also via regular homework assignments, with solutions provided on the webpage. Students are reminded that attendance at all timetabled activities is compulsory.

ASSESSMENT MODE(S)

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

Electronic Engineering Labs

Laboratory Programme Coordinator:

Ms. Bernadette Clerkin, Chief Technical Officer 1, Department of Electronic & Electrical Engineering, Printing House.

Introduction:

The programme of Electronic Engineering Laboratories is intended to complement and enhance the material covered in lectures for the wide range of subjects in the Junior Sophister year. With the exception of 3C7 Digital Systems Design which has its own integrated laboratory programme, each technical subject has two associated laboratories. Marks awarded for these laboratories will contribute to the overall mark for the particular subject at Annual and Supplemental Examinations. This is normally at the rate of 15% of the subject mark but this may vary for some subjects and is stipulated in the subject module sheet in the handbook. Each laboratory will require a properly structured report to be written up and submitted by each individual student which will then be marked by the laboratory demonstrator and returned to the student.

Attendance:

Attendance at the laboratories is compulsory and will be monitored throughout the year. Any report submitted by a student who has not attended the corresponding laboratory will not be marked. If a laboratory is missed due to illness or participation in an official College activity this should be certified and arrangements will be made where possible for the laboratory to be undertaken at a later stage. Casual or unexplained absences will not be facilitated. Please also note that laboratories not completed during the teaching semesters cannot be repeated during the summer vacation for supplemental examinations and existing marks will be carried forward to the supplemental results.

Reports:

You are required to write up a properly structured report on each laboratory undertaken. You may also be requested by the demonstrator to save or print out some electronic files from computer simulations as part of the submission. The report may be typed or handwritten. If it is handwritten it must be clearly legible to the demonstrator. The structure of the report should include:

Name: The student’s name and ID number.

Title: The code and name of the laboratory.

Date: Date on which laboratory was undertaken.

Aims: The specific intentions and objectives of the laboratory

Experimental Set-up: Details of the equipment used and the experimental set-up. If the laboratory is a simulation type the name and function of the software packages used should be given.

Procedure: An account of the steps involved in carrying out the experiment. A summarised version of the more detailed instructions given in the laboratory handout will suffice.

Results: A clear and accurate record of the results obtained. This should include tables of experimental data, numerical parameters, printouts of simulation waveforms or other appropriate forms of results. It should be possible from the results for a reader to get a complete understanding of the outcome of the laboratory.

Discussion: A detailed analysis and criticism of the results obtained. You should discuss the accuracy of the results, any limitations and their significance. You should relate them to the material covered in the lectures where possible. You should indicate what you have learned from the laboratory that is important in your discipline.

Conclusion: You should consider the importance and implications of the experiment you have carried out in the wider context of Electronic Engineering. You should give your opinions on what is good or bad practice concerning the topic covered by the laboratory and any professional ethical issues you feel are important.

Submission: You should submit your report for a particular experiment to Ms. Teresa Lawlor, the Executive Officer in the Departmental Office in the Printing House, within one week after completion of the relevant laboratory or as instructed by the laboratory demonstrator. The report must be 'signed in' by the student and countersigned by the Executive Officer in the register maintained for the purpose. A report of up to one week late will obtain half marks and a report will not be accepted after two weeks. If a student attends a laboratory but does not submit a report, no marks will be awarded for that laboratory. If a student submits a report but has not attended a laboratory, no marks will be awarded for that laboratory.

Note: Please keep a copy of your report for your records

3D1 – Microprocessor Systems I

COURSE TITLE: Microprocessor Systems 1		**CODE: 3D1**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER: Dr. John Waldron	**TEACHING ASSISTANT:**	
TERMS: Semester 1 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**
	PRACTICALS/WEEK: 2	**TOTAL: 14**

AIMS/OBJECTIVES

Microprocessor Systems 1 is a one-semester course taken by third year Electronic, Electronic/Computer and Computer Engineering students. This module provides students with an introduction to the basic structure, properties and operation of microprocessor systems. By developing and executing simple assembly language programs, the module aims to give students an understanding of how programs execute on a microprocessor system. The module also encourages students to consider the relationship between high level programming language constructs and their execution as sequences of instructions. Students will also be given opportunities to develop their problem solving, programming and written communication skills by designing solutions to programming problems, implementing those solutions, first in the form of high level programming constructs and then as assembly language programs, which must be documented and tested.

SYLLABUS

Specific topics addressed in this module include:

Number systems,
Memory and data representation
Binary arithmetic and logical operations
Floating-point representations and arithmetic
Basic computer architecture
Assembly language and machine code
Flow control
Memory load/store operations and addressing modes

ASSOCIATED LABORATORY/PROJECT PROGRAMME

7 LAB sessions: the practicals, conducted by each student individually, encourage the design, writing and testing of programmes and the development of the skills needed in actual practice.

BIBLIOGRAPHY

Recommended text: William Hohl, ARM Assembly Language: Fundamentals and Techniques, CRC Press, 2009.

Additional recommended texts:

Andrew Sloss, Dominic Symes and Chris Wright, ARM System Developer's Guide: Designing and Optimizing System Software, Morgan Kaufmann, 2004. nd

Steve Furber, ARM System-on-Chip Architecture, 2 edition, AddisonWesley Professional, 2000. Peter Knaggs, Stephen Welsh,

ARM: Assembly Language Programming, Bournemouth University, 2004.

WEBSITE

http://www.scss.tcd.ie/John.Waldron/3d1/3d1.html

LEARNING OUTCOMES

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

Describe the basic characteristics, structure and operation of a microprocessor system;

Translate between simple high-level programming language constructs and their assembly language equivalents;

Design, construct, document and test small-scale assembly language programs to solve simple problems;

Reason about the cost of executing instructions and the efficiency of simple programs;

Make use of appropriate documentation and reference material.

.

TEACHING STRATEGIES

The teaching strategy is a mixture of lectures, problem-solving tutorials and hands-on practical laboratory sessions. The format of lectures is conventional, however, a great deal of informal interaction is normal, and students can expect to participate in question-and-answer and problem solving sessions. For the first four weeks of the course, students are taught the general principles of low-level architecture and programming. Tutorials held during this time review basic skills such as binary and hexadecimal notation and algorithm design. Students are challenged to build programs based on a partial knowledge of the computer's instruction set. Practical sessions, starting around the fourth week, require the students to design, write, evaluate and debug their programs on special-purpose development systems. More advanced topics introduced during lectures become the subject of practical sessions through the rest of the semester.

ASSESSMENT MODE(S)

Assessment of this course is by formal written examination and by assessment of the practical laboratory sessions. Practical sessions attract a mark of up to 20% of the end-of-year mark, and the examination makes up the remaining 80% or more.

3D2 – Microprocessor Systems II

<table>
<tr><td colspan="2">COURSE TITLE: Microprocessor Systems 2</td><td>CODE: 3D2</td></tr>
<tr><td>LEVEL: Junior Sophister</td><td>CREDITS: 5</td><td>PREREQUISITES: None</td></tr>
<tr><td>LECTURER(S): Dr. Mike Brady</td><td colspan="2">TEACHING ASSISTANT:</td></tr>
<tr><td>TERMS: Semester 2
DURATION (WEEKS): 12</td><td>LECTURES/WEEK: 3
TOTAL: 33</td><td>TUTORIALS/WEEK: 1
TOTAL: 11</td></tr>
</table>

COURSE DESCRIPTION/AIMS

This course, which naturally follows 3D1, is divided into two parts (1) the design, construction and debugging of a simple microprocessor system and (2) the architecture of pipelined RISC CPUs.

In the first part of 3D2, students receive a theoretical introduction to the fundamental elements of a modern microprocessor system which they then have to put into practise by constructing a working microprocessor system comprising 68008 CPU, EPROM, RAM, ACIAs, glue logic and a basic monitor program. The course aims are (i) to give students enough knowledge so that they can design, construct and get their own microprocessor system working (ii) to make students realise that such systems are deterministic and do not work by "magic", (iii) that an attention to detail is important when dealing with hardware and software at this level, (iv) that logically based strategies need to be applied to locate faults whether they are at a design or constructional level and (v) to have exposure to working as a member of a group.

In the second part of 3D2, students receive an introduction to the architecture of RISC CPUs, instruction level pipelining and how data, load and control hazards can be resolved effectively. Students make use of an interactive web based animation of a DLX/MIPS CPU to explore these topics.

COURSE CONTENT

Part 1

- Review of totem-pole, tri-state & open-collector outputs.
- Logic design using programmable logic.
- Hardware review of the 68008 microprocessor.
- Interfacing the 68008 with memory and 6800 peripherals devices.
- Monitor implementation.
- Interrupt driven I/O.
- Design, construction and debugging of a simple 68008 based microprocessor system.

Part 2

- RISC vs CISC.
- RISC-1 design criteria and architecture.
- Register windows and delayed jumps.
- Instruction level pipelining.
- The DLX/MIPS pipeline.
- Resolving data, load and control hazards.

RECOMMENDED TEXTS

- Microprocessor Systems Design - Alan Clements
- Computer Architecture - a Quantitative Approach - John Hennessey & David Patterson
- High Performance Computer Architecture - Harold S. Stone

LEARNING OUTCOMES

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

1. Design simple logic circuits using programmable logic.
2. Design, construct and debug a simple microprocessor system.
3. Apply logically based strategies for locating faults.
4. Have experience working together as a group, keeping accurate documentation and writing an engineering report.
5. Explain the RISC design philosophy and translate high level language programs onto a RISC instruction set.
6. Explain the key concepts behind instruction level pipelining and know how to apply a number of techniques to overcome data, load and control hazards.

TEACHING STRATEGIES

The teaching strategy is a mixture of lectures, problem solving tutorials and six 3 hour practicals. A key component is the “hands-on” group project to construct a simple microprocessor system from lecture notes and a kit of parts using wire-wrapping. Students have monitored weekly targets which they should try and meet. Students can avail of help from the course lecturer, teaching assistants and from each other during the 3 hour weekly practical. Students learn how use oscilloscopes and logic analysers to collect evidence in order to formulate logical strategies for fault location and correction. This is the most interesting and challenging part of the course.

Students make use of an interactive web based animation of a DLX/MIPS CPU to explore the DLX/MIPS pipeline and how data, load and control hazards are resolved.

ASSESSMENT MODE(S)

The group project account for 20% of the final mark and the exam 80%. Students must answer 5 out of 6 exam questions (four from part 1 and two from part 2).

The group project is marked from their weekly progress reports and a group project report. All members of the same group normally receive identical marks.

3D3 – Computer Networks

COURSE TITLE: Computer Networks		**Code: 3D3**
LEVEL: Junior Sophister	**CREDITS: 5**	**Prerequisites: None**
LECTURER(S): Dr. Hitesh Tewari	**TEACHING ASSISTANT:**	
TERMS: Semester 2 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

This course introduces students to computer networks and concentrates on building a firm foundation for understanding Data Communications and Computer Networks. It is based around the OSI Reference Model which deals with the major issues in the bottom four (Physical, Data Link, Network and Transport) layers of the model. Students are also introduced to the areas of Network Security and Mobile Communications. This course provides the student with fundamental knowledge of the various aspects of computer networking and enables students to appreciate recent developments in the area.

SYLLABUS

Introduction to computer networks;
Physical layer issues;
The datalink layer;
Local area networks;
TCP/IP suite of protocols;
Network security;
Mobile networking

RECOMMENDED TEXT(S)

1. Computer Networks (4th edition), Andrew Tanenbaum, Prentice Hall
2. Computer Networking and the Internet (5^{th} edition),Fred Halsall, Addison Wesley
3. Data Communications and Networking (4th edition), Behrouz Forouzan, McGraw Hill
4. TCP/IP Protocol Suite (3rd edition), Behrouz Forouzan, McGraw Hill

LEARNING OUTCOMES

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

1. analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies;
2. have a basic knowledge of the use of cryptography and network security;
3. specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols;
4. analyze, specify and design the topological and routing strategies for an IP based networking infrastructure;

TEACHING STRATEGIES

The lectures are designed to provide students with a better knowledge of some of the important networking protocols - students attend formal lectures during which they are given handouts of the course notes with a few gaps that they fill in during the course of the lecture. The emphasis during the lecture is on allowing the student to focus on the lecture and pose questions on various issues that may arise. They are given recent journal and conference papers that provide them with an overview of recent and emerging developments in data communications and networking.

ASSESSMENT MODE(S)

Assessment is by means of a formal written end-of-year examination and by assessment of the practical sessions. Practical sessions (the course project work and tutorials) carry a mark of up to 20% of the end-of-year mark and the examination makes up the remaining 80%.

Note: Laboratories not completed during the teaching semesters cannot be repeated for supplemental examinations and existing marks will be carried forward to the supplemental examinations.

3D4 – Concurrent Systems and Operating Systems

COURSE TITLE: Concurrent Systems II		**CODE: 3D4**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Dr Mike Brady	**TEACHING ASSISTANT:**	
TERMS: Semester 1 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 2** **TOTAL: 22**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

The first part of this module, lasting six weeks, introduces students to concurrency and concurrent programming. The aim is to provide students with the ability to develop concurrent software systems using standard techniques and constructs.

To achieve this aim, students must have a thorough understanding of common problems that arise in concurrent systems and how those problems can be avoided. This module will teach the use of tools and techniques for modelling and verifying the correctness of concurrent systems, applying this through practical laboratory exercises in which small concurrent software systems are developed.

The second part of the module, lasting five weeks, addresses various aspects of the design of modern operating systems. The main aim is to explore how programmers can apply a knowledge of operating system features to the design of efficient applications. This is achieved by examining common algorithms and policies used by modern operating systems, as well as the facilities provided to application programmers. This knowledge is then applied in laboratory exercises.

SYLLABUS

Concurrent Systems

- introduction to concurrency;
- simple multi-threaded programs in Java and C;
- modelling concurrent systems;
- interference;
- mutual exclusion;
- critical sections;
- verification of concurrent programs;
- semaphores;
- monitors.

Operating Systems

- operating system architectures;
- memory management;
- processor scheduling;
- disk I/O, file systems.

SUGGESTED READING

Suggested Reading:

- *Principles of Concurrent and Distributed Programming (2nd edition)*, M Ben-Ari, Addison-Wesley, 2006
- *Operating System Concepts (7th edition)*, A Silberschatz, PB Galvin and G Gagne, Wiley, 2005

LEARNING OUTCOMES

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

1. recognise standard concurrent programming problems;
2. solve concurrent programming problems using standard techniques;
3. design and implement concurrent programs using different programming languages and APIs (e.g. Java, C++, Win32);
4. develop models of concurrent programs using the Promela modelling language;
5. verify the correctness of concurrent programs of moderate complexity using SPIN;
6. describe algorithms, data structures and policies used in modern operating systems for thread scheduling, memory management, disk I/O and file management;
7. evaluate, compare and reason about the relative performance of algorithms used by operating systems (e.g. page replacement policies);
8. apply a knowledge of operating system behaviour when developing user-level programmes.

TEACHING STRATEGIES

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

ASSESSMENT

Assessment is continuous assessment (up to 20%) with the remainder by examination.

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

3D5 – Software Design and Implementation

COURSE TITLE: Software Design and Construction		**CODE: 3D5**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Mr Glenn Strong	**TEACHING ASSISTANT:**	
TERMS: Semester 1 & 2 **DURATION (WEEKS): 24**	**LECTURES/WEEK: 1** **TOTAL: 22**	**TUTORIALS/WEEK: 1** **TOTAL: 22**

Software Design and Implementation is a two-semester course; the first semester is taken by Electronic/Computer and Computer Engineering students, and the second semester by Computer Engineering students only (Electronic/Computer Engineering students take 3C6 in the second term). The course covers software engineering practice through one or two semester-long programming projects where students gain practical experience with good software engineering techniques through the design, implementation and testing of a software application.

The course is intended to give students direct experience with programming to teach application of techniques for program design, software implementation, testing and documentation. Concepts such as requirements capture, the software engineering life cycle, object-oriented software design, the design of the Java programming language, GUI implementation in Java, project management, software testing, selection of appropriate algorithms and data structures for problem solving, and basic concurrency and networking are introduced.

The project presented in the second semester (to Computer Engineering students only), while being a separate piece of work, builds on the expertise gained in the first project and deepens the students' understanding of the software engineering issues presented in the course while introducing new challenges in the technology and method.

SYLLABUS

Specific topics addressed in this module include:

review of object-oriented design techniques;

- expressing program design with UML class diagrams;
- some basic principles of project management;
- principles of programming in the Java language;
- high level program design in the Java programming language (classes, interfaces, inheritance);
- file and console IO in Java;
- robust software implementation, exceptions;
- principles of user-interface implementation in Java;
- event driven programming, Java listeners;
- basic threading and concurrency in Java;
- algorithms and data-structures as required by the project;
- software documentation;
- design and application of software testing;
- discussion of professional and ethical issues relating to project development.

RECOMMENDED TEXT(S)

- Any Java programming textbook, such as by Deitel & Deitel, Smith, etc
- Any software project management textbook, such as Stiller & LeBlanc, etc

LEARNING OUTCOMES

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

1. analyse, design and implement software of reasonable (e.g. up to 4000 lines) complexity in the Java programming language. The student will be able to employ suitable advanced programming techniques in this implementation;
2. examine a problem specification and write an object-oriented programme design for the problem;
3. plan the implementation of the program and manage their time to ensure each phase of the implementation is given sufficient attention;
4. write documentation for a software project using a standard technique (e.g. javadoc);
5. employ standard testing techniques (e.g. unit, integration, system tests; black and white box testing) to ensure the quality of their software.

TEACHING STRATEGIES

This course runs in two parts: Part A, taken by Electronic/Computer and Computer Engineering students runs for all eleven weeks of the first semester (Michaelmas term) and comprises of one tutorial plus one three-hour practical laboratory session per week. The total contact times is fourty-four hours total.
The second part is taken by D-stream students only and runs for the Hilary term on the same schedule as part 1, for a total of fourty-four contact hours.

The teaching strategy is primarily a hands-on practical one, with the students expected to engage in considerable informal discussion and interaction, particularly during the design and early implementation phase of each project. Tutorials are run as informal lectures with question-and-answer and discussion sessions. In the early part of the course, laboratory sessions are run as practical sessions during which students design, write and debug small programs in the Java programming language. During the rest of the term, laboratory sessions are used to demonstrate progress towards specific milestones on the project implementation, and as discussion and help sessions. Students are encouraged to interact with each other in small groups during the laboratory sessions to discuss issues in the design and implementation of the project, and to discover practical solutions to programming problems. Students are encouraged to do additional research and reading into the project backgrounds and details in order to enrich their understanding of the project and provide amore fully rounded project implementation.

ASSESSMENT MODE(S)

Assessment of this course is entirely by coursework. Each project contributes 50% towards the end-of year mark (Electronic/Computer Engineering students have the first project mark combined with the 3C6B mark).
During the term, students are expected to demonstrate suitable progress towards milestones in the project. These demonstrations occur in the third, sixth and eighth week laboratory sessions. These laboratory sessions are run by postgraduate student demonstrators who are experts in the subject, and by the course lecturer. Progress towards these milestones are not formally included in the students final marks, however an inability to adequately demonstrate progress is taken as a warning sign that a student may be experiencing difficulty.

In the examination of the submitted project, marks are primarily awarded for correctness and completeness of implementation with respect to the problem specification, with smaller components for quality of technical documentation and employment of suitable high-level design.

A more detailed breakdown of the marking scheme for one project (some projects contain specific technical challenges which may warrant a slightly different scheme) is as follows:

- 50% of the project marks are allocated for the correctness of the project implementation. The two most important components of the correctness are (i) basic correctness with respect to the project specification, for which 30% of the overall marks can be awarded. This covers the basic ability of the programme to perform as described in the project specification and (where appropriate) to correctly process provided sample data. The second most important component for which 20% of the overall marks may be awarded consists of correctness of the programme with respect to good programming practice (selection of suitable algorithms and data structures, resource management), and for the program taking appropriate behaviour when provided with invalid user input if appropriate to the project;
- 30% of the project marks are allocated for the project design, covering a variety of functional and

non-functional components. This is broken down into equal components for demonstrating suitable high-level application of object-oriented programming concepts (selection of suitable classes, designing of appropriate interfaces between those classes, projected ability to re-use the software components where applicable), and for demonstrating suitable design in the construction of each class (selection of suitable methods and design patterns for each class, suitable parameterisation of the methods, appropriate visibility for class members);

1. 20% of the project marks are allocated for documenting the project. This reflects the marks available for the documentation which accompanies the final project submission itself. This documentation covers standard programmer documentation intended for use by maintenance programmers enhancing or bug fixing the application. This documentation generally comes in two forms, both a standard set of class/method documents implemented in the JavaDoc standard (although students may submit documentation in another form if it is suitable), and inline programming comments documenting specific implementation decisions. The documentation is evaluated from both a completeness perspective (ensuring that the entire submission is suitably documented) and correctness (ensuring that the documentation is correct with respect to the behaviour of the programme). Consideration is also given here to the student's choice of identifiers and any other elements in the program which contribute to an ability to comprehend the system.

CS2002 – Computer Architecture II

Module Code	CS2022
Module Title	Computer Architecture II
Pre-requisites	None
ECTS	5
Chief Examiner	Dr Michael Manzke
Teaching Staff	Dr Michael Manzke
Delivery	Computer Architecture II runs over the eleven weeks of Michaelmas Term. In each week, there are two one-hour lectures, a one-hour tutorial and a one-hour laboratory session. Attendance at all lectures, tutorials and laboratory sessions is compulsory.
Aims	The lectures and tutorials treat the detailed design and organisation of an instruction processor. Course Work: Two projects using VHDL and ModelSim to simulate and test their design. 1. A processor unit (ALU + shifter + fast registers) design and simulation, 2. An instruction processor design and simulation. Contents: Digital Logic, Register transfer definition, micro-operations, bus transfers, ALU design, shifter design, hardwired control design, microprogrammed processor control, design of an instruction processor. The aims of the course are to learn register-transfer specification and design and learn the fundamentals of an instruction processor.
Learning Outcomes	Students will be able to • design substantial logic circuits using register transfer descriptions; • test and verify their design using an industry standard hardware description language (VHDL); • understand the organisation and execution behaviour of general-purpose processor systems;
Syllabus	Specific topics addressed in this module include: • Digital Logic • Register transfer language • ALU and shifter design • Multiplexer and tristate busses. • Datapath design • Instruction fetch-decode-execute cycle
Assessment	Assessment is by examination (80%) and continuous assessment (20%). Continuous assessment is composed of a number of marked laboratory exercises and two substantial assignments.
Bibliography	Recommended text: • Introductory VHDL: From Simulation to Synthesis • Logic and Computer Design Fundamentals" 2nd Edition updated, Mano Additional recommended texts: • TBD
Website	http://www.scss.tcd.ie/Michael.Manzke/2ba4.html#Semester_2

ST 2004 – Applied Probability

Academic Year	2009-2010
Module Code	ST2004
Module Title	***Applied Probability: Introduction***
Pre-requisites	
ECTS	5
Chief Examiner	Prof John Haslett
Teaching Staff	Prof John Haslett
Delivery	Classroom teaching; lectures/tutorials/computer labs
Aims	An introduction to probability, intended as preparatory for later courses
Learning Outcomes	When students have successfully completed this module they should: • Understand the basic ideas of probability modelling; ie how to describe uncertainty by probability statements, and how to combine uncertainties to make statements about the uncertainty of systems

Syllabus	Specific topics addressed in this module include:
	• Basic concepts in probability, including
	  o the basic rules
	  o probability mass functions, density functions and cumulative distribution functions for discrete and continuous univariate random variables
	  o bivariate probability distributions for discrete random variables
	  o expected values, variances and standard deviations, covariances
	• The use of Monte Carlo methods in modelling systems of random variables
	• Standard models including:
	  o for discrete random variables, the Binary (Bernoulli) , Discrete Uniform, Binomial, Poisson, and
	  o for continuous random variables, the Uniform, Normal and Exponential
	• The implications for approximations of the Central Limit Theorem

Assessment	Two group assignments and one 2-hour exam
Bibliography	Main Text: Dekker, Kraaikamp, Lopuhaa, Meester: A Modern Introduction to Probability and Statistics, Springer,2005 Other recommended reading Hand, D.J. Statistics: A Very Short Introduction, Oxford, 2008 Applebaum, D: Probability and Information, 2nd ed Cambridge 2008 Stirzaker, D: Probability and Random Variables: a beginners guide, 2005 Williams, D: Weighing the Odds: a Course in Probabilty and Statistics; Chap 1-4; Cambridge, 2004 Mitzenmacher, M and Upfal, E, Probability and Computing, Cambridge, 2005 Tijms, H: Understanding Probability, 2nd ed, Cambridge, 2007 Swift, L: Mathematics and Statistics for Business, Management and Finance, MacMillan 1997
Website	

3E1 – Mathematics V

COURSE TITLE: Engineering Mathematics V		**CODE: 3E1**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Dr. B. Browne	**TEACHING ASSISTANT:**	
TERMS: Semester 1 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 3** **TOTAL: 33**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

Engineering Mathematics V is a one-semester course available to all JS Engineering streams and continues and extends the material from the previous mathematics courses in the first and second years - 1E1, 1E2, 2E1 and 2E2. The emphasis is primarily on the development of analytical techniques.

SYLLABUS

Optimization.

(i) Linear Programming: Simplex Algorithm ,Duality Theory.

(ii)Nonlinear Optimization: Newton's method.

Applications to Engineering and Business problems.

Solution of Partial Differential Equations

(i) Characteristics and Classification of Partial Differential Equations.

(ii) Method of Separation of Variables.

(iii) D'Alembert' solution of the Wave equation.

(iv) Orthogonal Functions and General Expansion Problems.

Applications to Electrical, Civil and Mechanical Engineering Problems.

All lectures and tutorials will be available on the Intranet.

RECOMMENDED TEXT(S)

Advanced Engineering Mathematics, E. Kreyszig, J. Wiley & Sons.

LEARNING OUTCOMES

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

- calculate the coefficients of both the complex and the real Fourier series for a variety functions, and to use them to solve some ordinary differential equations.
- calculate Fourier transforms, discrete or continuous, for a variety of simple functions - students will then be able to use these to compute convolutions in simple cases;
- solve the Laplace, heat and wave equations for a variety of boundary conditions in domains of simple

geometry and with simple boundary conditions; the techniques available will include, separation of variables, Laplace and Fourier Transform methods.

- solve linear and non-linear optimization problems.
- apply above methods to solve problems in different areas of engineering.

TEACHING STRATEGIES

The teaching strategy is a mixture of lectures and problem-solving tutorials. Whilst the format of lectures is conventional and the atmosphere is informal, some interaction and discussion is common and students are encouraged to ask questions. In the tutorials, all students work on problems which practice and apply the methods introduced in the lectures. Discussion of problems in small groups is encouraged and facilitated.

ASSESSMENT MODE(S)

Assessment for this course is carried out by means of a written examination at the end of the academic year. The subject mark is based entirely on the result of this written examination.

3E1a – Engineering Analysis

COURSE TITLE: **Engineering Analysis**		CODE: **3E1a**
LEVEL: **Junior Sophister**	CREDITS: **5**	PREREQUISITES: **2E1 AND 2E2, OR PERMISSION OF THE LECTURER**
LECTURER: **Liam Dowling**	TEACHING ASSISTANT:	
TERMS: **Semester 1** DURATION (WEEKS): **12**	LECTURES/WEEK: **3** TOTAL: **33**	TUTORIALS/WEEK: **1** Total: 11

AIMS/OBJECTIVES

This course is developed to strengthen the student's skills in applied engineering analysis and is organised into three main subsections: signal and system analysis; partial differential equations; and optimization. The first section deals with transform analysis applied to engineering signals and systems. The second part of the course deals with methods for solving partial differential equations. The final section focuses on linear and nonlinear optimization for engineering design.

SYLLABUS

Signal and System Analysis:

Properties and applications of Fourier and Laplace transforms.

Linear Time-Invariant Systems: Impulse response and the convolution integral; properties of LTI systems; transfer function and frequency response of an LTI system.

Sampling Theorem: Representation of a continuous-time signal by its samples; undersampling and aliasing; the sampling theorem; reconstruction of a bandlimited signal from its samples.

Partial Differential Equations

Solution by separating variables: the Wave Equation; the Heat Equation; and Laplace's Equation.

Optimization

Linear Programming: The Simplex algorithm

Unconstrained Optimization: The gradient method; the golden section method

RECOMMENDED TEXT

Kreyszig, E., *Advanced Engineering Mathematics.* 9th ed. New York: Wiley, 2006

LEARNING OUTCOMES

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

1. Analyse continuous-time signals using Fourier transforms and Fourier series.
2. Analyse linear time-invariant systems using Fourier and Laplace transform methods.
3. Solve the Wave equation, Heat equation, and Laplace's equation for various initial and boundary conditions.
4. Solve linear programming problems using the Simplex algorithm.
5. Use gradient methods to optimize a function.

TEACHING STRATEGIES

The course is taught using a combination of lectures and problem solving tutorials.
ASSESSMENT The annual examination counts for 70% and each of the two in-class test counts for 15% of the overall subject mark.

3E4 – Management for Engineers

COURSE TITLE: Management for Engineers		**CODE: 3E4**
LEVEL: Junior Sophister	**CREDITS: 5**	**PREREQUISITES: None**
LECTURER(S): Dr. Niamh Harty, Ms **Joanna Gardiner, Dr Brian Caulfield**	**TEACHING ASSISTANT:**	
TERMS: Semester 2 **DURATION (WEEKS): 12**	**LECTURES/WEEK: 2** **TOTAL: 22**	**TUTORIALS/WEEK: 1** **TOTAL: 11**

AIMS/OBJECTIVES

Management for Engineers introduces engineering students to Entrepreneurship and Communication. The aims of the course are:

- To foster a sense of entrepreneurship among the JS Engineering students, by requiring the students to come up with a business idea and during the semester produce a business plan.
- To enable students to communicate well in engineering contexts, both when *talking* about projects, plans and problems, and when *writing* about these.

SYLLABUS

The course covers the following topics:

Entrepreneurship:

- Coming Up with a Business Idea
- Marketing
- Feasibility
- Market Research
- Legal Issues and Ethics
- Finance and Accounting
- Business Plan
- Ethics
- Growth of the Business

Communication:

- Intersubjectivity
- Emails
- Reports
- Presentations
- Intercultural communication
- Media Interviews

RECOMMENDED TEXT(S)

TBA

LEARNING OUTCOMES

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

- Prepare a business plan, including details of marketing, market research, finance, legal issues and growth.
- Give a presentation
- Summarise a technical article

ASSESSMENT MODE(S)

There will be three assignments on entrepreneurship, and two assignments on Communication, plus a final examination.

Entrepreneurship counts for 50% of overall mark in 3E4. Marks for Entrepreneurship will be divided 60% for continuous assessment, and 40% for questions on the Final examination.

Communication counts for 50% of overall mark in 3E4. Marks for Communication will be divided 40% for continuous assessment, and 60% for questions on the Final examination.