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

2012-2013
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:

<table>
<thead>
<tr>
<th>Cal. Wk</th>
<th>Dates 2012/13 (week beginning)</th>
<th>Outline Structure of Academic Year 2012/13</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>27-Aug-12</td>
<td>Supplemental Examinations</td>
<td>Statutory Term (Michaelmas) begins</td>
</tr>
<tr>
<td>2</td>
<td>03-Sep-12</td>
<td></td>
<td>Michaelmas Lecture term begins</td>
</tr>
<tr>
<td>3</td>
<td>10-Sep-12</td>
<td>Orientation Week/Freshers' Week</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>17-Sep-12</td>
<td>Teaching Week 1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>24-Sep-12</td>
<td>Teaching Week 2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>01-Oct-12</td>
<td>Teaching Week 3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>08-Oct-12</td>
<td>Teaching Week 4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>15-Oct-12</td>
<td>Teaching Week 5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>22-Oct-12</td>
<td>Teaching Week 6 (Monday, Public Holiday)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>29-Oct-12</td>
<td>Teaching Week 7 - Study Week</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>05-Nov-12</td>
<td>Teaching Week 8</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12-Nov-12</td>
<td>Teaching Week 9</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>19-Nov-12</td>
<td>Teaching Week 10</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>26-Nov-12</td>
<td>Teaching Week 11</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>03-Dec-12</td>
<td>Teaching Week 12</td>
<td>6-Michaelmas term ends Friday 14 December 2012</td>
</tr>
<tr>
<td>16</td>
<td>10-Dec-12</td>
<td>Christmas Period (College closed)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>17-Dec-12</td>
<td>24 December 2012 to 1 January 2013, inclusive</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>24-Dec-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>31-Dec-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>07-Jan-13</td>
<td>Foundation Scholarship Examinations</td>
<td>Note: it may be necessary to hold some exams in the preceding week.</td>
</tr>
<tr>
<td>21</td>
<td>14-Jan-13</td>
<td>Teaching Week 1</td>
<td>Hiliary Term begins</td>
</tr>
<tr>
<td>22</td>
<td>21-Jan-13</td>
<td>Teaching Week 2</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>28-Jan-13</td>
<td>Teaching Week 3</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>04-Feb-13</td>
<td>Teaching Week 4</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>11-Feb-13</td>
<td>Teaching Week 5</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>18-Feb-13</td>
<td>Teaching Week 6</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>25-Feb-13</td>
<td>Teaching Week 7 - Study Week</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>04-Mar-13</td>
<td>Teaching Week 8</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>11-Mar-13</td>
<td>Teaching Week 9 (Sunday, St Patrick's Day)</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>18-Mar-13</td>
<td>Teaching Week 10 (Monday, Public Holiday)</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>25-Mar-13</td>
<td>Teaching Week 11 (Friday, Good Friday)</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>01-Apr-13</td>
<td>Teaching Week 12 (Monday, Easter Monday)</td>
<td>4-Hillary Term ends Friday 5 April 2013</td>
</tr>
<tr>
<td>33</td>
<td>08-Apr-13</td>
<td>Revision Trinity Week (Monday, Trinity Monday)</td>
<td>Trinity Term begins</td>
</tr>
<tr>
<td>34</td>
<td>15-Apr-13</td>
<td>Revision</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>22-Apr-13</td>
<td>Revision</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>29-Apr-13</td>
<td>Annual Examinations 1</td>
<td>Annual Examination period: 4 weeks at present followed by 4-5 weeks for marking, examiners’ meetings, publication of results, Courts of First Appeal and Academic Appeals.</td>
</tr>
<tr>
<td>37</td>
<td>06-May-13</td>
<td>Annual Examinations 2 (Monday, Public Holiday)</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>13-May-13</td>
<td>Annual Examinations 3</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>20-May-13</td>
<td>Annual Examinations 4</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>27-May-13</td>
<td>Marking/Courts of Examiners/Results</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>03-Jun-13</td>
<td>Marking/Courts of Examiners/Results (Monday, Public Holiday)</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>10-Jun-13</td>
<td>Marking/Courts of Examiners/Results</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>17-Jun-13</td>
<td>Marking/Courts of Examiners/Results/Courts of First Appeal</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>24-Jun-13</td>
<td>Courts of First Appeal/Academic Appeals</td>
<td>Statutory (Trinity) Term ends Friday 28 June 2013</td>
</tr>
<tr>
<td>45 to 52</td>
<td>01 Jul 2013 - 19 Aug 2013</td>
<td>Postgraduate dissertations/theses/Research 1-8</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Ends Friday 23 August 2013</td>
</tr>
</tbody>
</table>
Courses undertaken by the Junior Sophister students are:

<table>
<thead>
<tr>
<th>Faculty Courses</th>
<th>Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>3E1 Engineering Maths V</td>
<td>I</td>
</tr>
<tr>
<td>3E1a Engineering Analysis</td>
<td>I</td>
</tr>
<tr>
<td>3E4 Management for Engineers</td>
<td>II</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EEE Dept Courses</th>
<th>Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>3C1 Signals and Systems</td>
<td>I</td>
</tr>
<tr>
<td>3C2 Digital Circuits</td>
<td>I</td>
</tr>
<tr>
<td>3C3 Analogue Circuits</td>
<td>II</td>
</tr>
<tr>
<td>3C4 Applied Electromagnetism</td>
<td>II</td>
</tr>
<tr>
<td>3C5 Telecommunications</td>
<td>II</td>
</tr>
<tr>
<td>3C6(a&amp;b) Electronic Design Projects</td>
<td>I &amp; II</td>
</tr>
<tr>
<td>3C7 Digital Systems Design</td>
<td>II</td>
</tr>
<tr>
<td>3E3 Probabilistic Methods</td>
<td>I</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CS Dept Courses</th>
<th>Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D1 Microprocessor Systems I</td>
<td>I</td>
</tr>
<tr>
<td>3D2 Microprocessor Systems II</td>
<td>II</td>
</tr>
<tr>
<td>3D3 Computer Networks</td>
<td>II</td>
</tr>
<tr>
<td>3D4 Operating Systems &amp; Concurrent Systems</td>
<td>II</td>
</tr>
<tr>
<td>3D5(a&amp;b) Software Design and Implementation Project</td>
<td>I &amp; II</td>
</tr>
<tr>
<td>CS2022 Computer Architecture II</td>
<td>II</td>
</tr>
<tr>
<td>ST2004 Applied Probability</td>
<td>I</td>
</tr>
</tbody>
</table>
## C Stream

### Semester 1

<table>
<thead>
<tr>
<th>Module Title</th>
<th>ECTS</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maths V / Engineering Analysis</td>
<td>5</td>
<td>3E1a</td>
</tr>
<tr>
<td>Probabilistic Methods</td>
<td>5</td>
<td>3E3</td>
</tr>
<tr>
<td>Signals and Systems</td>
<td>5</td>
<td>3C1</td>
</tr>
<tr>
<td>Digital Circuits</td>
<td>5</td>
<td>3C2</td>
</tr>
<tr>
<td>Microprocessors I</td>
<td>5</td>
<td>3D1</td>
</tr>
<tr>
<td>Design Project I</td>
<td>5</td>
<td>3C6a</td>
</tr>
</tbody>
</table>

### Semester 2

<table>
<thead>
<tr>
<th>Module Title</th>
<th>ECTS</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mgt for Engineers</td>
<td>5</td>
<td>3E4</td>
</tr>
<tr>
<td>Digital Systems Design</td>
<td>5</td>
<td>3C7</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>5</td>
<td>3C5</td>
</tr>
<tr>
<td>Analogue Circuits</td>
<td>5</td>
<td>3C3</td>
</tr>
<tr>
<td>Applied Electromagnetism</td>
<td>5</td>
<td>3C4</td>
</tr>
<tr>
<td>Design Project II</td>
<td>5</td>
<td>3C6b</td>
</tr>
</tbody>
</table>

## D Stream

### Semester 1

<table>
<thead>
<tr>
<th>Module Title</th>
<th>ECTS</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maths V / Engineering Analysis</td>
<td>5</td>
<td>3E1a</td>
</tr>
<tr>
<td>Applied Probability</td>
<td>5</td>
<td>ST2004</td>
</tr>
<tr>
<td>Signals and Systems</td>
<td>5</td>
<td>3C1</td>
</tr>
<tr>
<td>Digital Circuits</td>
<td>5</td>
<td>3C2</td>
</tr>
<tr>
<td>Microprocessors I</td>
<td>5</td>
<td>3D1</td>
</tr>
<tr>
<td>Design Project I</td>
<td>5</td>
<td>3D5a</td>
</tr>
</tbody>
</table>

### Semester 2

<table>
<thead>
<tr>
<th>Module Title</th>
<th>ECTS</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mgt for Engineers</td>
<td>5</td>
<td>3E4</td>
</tr>
<tr>
<td>Computer Architecture II</td>
<td>5</td>
<td>CS2022</td>
</tr>
<tr>
<td>Computer Networks</td>
<td>5</td>
<td>3D3</td>
</tr>
<tr>
<td>Operating Systems &amp; Concurrent Systems</td>
<td>5</td>
<td>3D4</td>
</tr>
<tr>
<td>Microprocessors II</td>
<td>5</td>
<td>3D2</td>
</tr>
<tr>
<td>Design Project II</td>
<td>5</td>
<td>3D5b</td>
</tr>
</tbody>
</table>
CD Stream

**Semester 1**

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

**Semester 2**

<table>
<thead>
<tr>
<th>Module Title</th>
<th>ECTS</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mgt for Engineers</td>
<td>5</td>
<td>3E4</td>
</tr>
<tr>
<td>Digital Systems Design OR</td>
<td>5</td>
<td>3C7</td>
</tr>
<tr>
<td>Computer Architecture II</td>
<td></td>
<td>CS2022</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>5</td>
<td>3C5</td>
</tr>
<tr>
<td>Computer Networks</td>
<td>5</td>
<td>3D3</td>
</tr>
<tr>
<td>Microprocessors II</td>
<td>5</td>
<td>3D2</td>
</tr>
<tr>
<td>Design Project II</td>
<td>5</td>
<td>3C6b</td>
</tr>
</tbody>
</table>
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 microchip 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

To comply with Council approved Harmonisation Regulations, the BAI Examination Rules need to be re-drafted. When this task is completed in the near future, the Examination Regulations will be made available to students.
### Module Title: Engineering Mathematics V

<table>
<thead>
<tr>
<th>Code: MA3E1</th>
</tr>
</thead>
</table>

| Level: Junior Sophister |
| Credits: 5 |
| Prerequisites: None |

<table>
<thead>
<tr>
<th>Lecturer(s): Dr Brendan Browne</th>
</tr>
</thead>
</table>

| Terms: Semester 1 |
| Lectures/Week: 3 |
| Tutorials/Week: 1 |

| Duration (weeks): 12 |
| Total: 33 |
| Total: 11 |

### Aims/Objectives

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

### Syllabus

- **Review of Fourier Methods**
  - Definition of complex and real Fourier series;
  - Application of Fourier series to solve ordinary differential equations;
  - Even and odd half-range expansions;
  - Definition of Fourier transform;
  - Interpretation of Fourier modes as frequencies;
  - Convolution.

- **Partial Differential Equations**
  - Laplace’s equation;
  - The heat equation;
  - The wave equation;
  - D’Alembert’s solution;
  - Fundamental solutions;
  - Separation of variables;
  - Application of Fourier analysis to initial value problems.

- **Optimisation**
  - Linear programming.
  - Non-linear optimization. Lagrange multipliers.
  - Newton’s and Conjugate Gradient methods.

### Recommended Text(s)

- Advanced Engineering Mathematics, E. Kreyszig
**LEARNING OUTCOMES**

Upon completion of this module, students should be able to:

- calculate the coefficients of both the complex and the real Fourier series for a variety of 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 module is carried out by means of a written two-hour examination at the end of the academic year. The subject mark is based entirely on the result of this written examination.
**MODULE TITLE:** Engineering Analysis  
**CODE:** 3E1a

<table>
<thead>
<tr>
<th>LEVEL: Junior Sophister</th>
<th>CREDITS: 5</th>
<th>PREREQUISITES: 2E1 AND 2E2, OR PERMISSION OF THE LECTURER</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LECTURER: Assistant Professor Liam Dowling</th>
<th>TEACHING ASSISTANT:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TERMS: Semester 1</th>
<th>LECTURES/WEEK: 3</th>
<th>TUTORIALS/WEEK: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DURATION (WEEKS): 12</td>
<td>TOTAL: 33</td>
<td>TOTAL: 11</td>
</tr>
</tbody>
</table>

**AIMS/OBJECTIVES**

This module 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 module 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**

<table>
<thead>
<tr>
<th>LEARNING OUTCOMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>On completion of this module the student will be able to:</td>
</tr>
</tbody>
</table>

1. Analyse continuous-time signals using Fourier transforms and Fourier series.


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.

<table>
<thead>
<tr>
<th>TEACHING STRATEGIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is taught using a combination of lectures and problem solving tutorials.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ASSESSMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>The annual examination counts for 70% and each of the two in-class test counts for 15% of the overall subject mark.</td>
</tr>
</tbody>
</table>
AIMS/OBJECTIVES
This module provides a thorough grounding in probability for electrical, computer and bio-engineering students. In particular, it equips the student with methods for dealing with uncertainty in engineering practice, notably in the analysis of experiments, and the interpretation and processing of data. The keynote of the module is a philosophical one. The relationship between uncertainty and information (learning) is explored from the start. A full review of propositional logic is provided, so that the student can be confident in formulating propositions around an uncertain experiment, and in understanding the logical relationships between propositions. The probability calculus is developed as a consistent means of quantifying and manipulating belief in these uncertain propositions (i.e. the Bayesian perspective). In this way, the foundation of the module is a unified one, with uncertainty, logic, information, observation and imprecision (noise) all embraced within a Bayesian notion of probability.

The main aim is to confront engineering contexts that induce the canonical probability models, in both the discrete case (Bernoulli, geometric, binomial, multinomial, Poisson) and the continuous case (rectangular, exponential, m-Erlang, normal). Their mixtures and transformations are developed, as a response to practical modelling needs. There is a special emphasis on the concept of dependence (conditioning), and its relation to the key engineering notions of correlation and prediction. Sequential dependence is handled in the discrete case only, via a full introduction to Markov chains. A main learning outcome for the student is knowledge of which model applies in which context, and the assumptions justifying the deployment of each model.

This theory is not separated from the engineering practice it aims to serve. Rather, the module carries a number of extended case studies throughout, each progressively refined as our new probability tools become available to us. The main case studies are (i) the noisy digital communication system, using discrete models and, later, the additive Gaussian noise model; (ii) Poisson count bio-imaging contexts (FLIM, SPECT, FLIM); and (iii) reliability and traffic modelling in large device assemblies.

A feature of the module is that it develops statistics consistently using the same inductive inference principles. Typically a black-spot in the formation of the engineering student, statistical inference is re-cast as a problem of probability modelling. Only the simple nonparametric case is considered, using the elegant device of the empirical distribution, readily allowing students to derive sampling statistics for their data. Themes of importance to engineers within the data analysis context are emphasized: (a) quantization error, (b) probability estimation from survey data, and (iii) regression between variables.

The module primes the student for later modules on random processes and statistical
signal processing. In its own terms, it is an invitation to confront uncertainty as a fundamental phenomenon – and resource – in engineering systems, and to appreciate probability as a consistent framework for the design, analysis and optimization of such systems.

**SYLLABUS**

- **Review of Propositional Logic**  
  Uncertain experiments in electrical, computer and bio-engineering  
  Sample space, propositions and events  
  Propositional logic: equivalence, necessity, sufficiency, mutual exclusivity

- **The Foundation of Probability Modelling**  
  The axioms of probability and the probability triple  
  Conditional probability; independence  
  Key relationships: chain rule, Bayes’ rule, theorem of total probability

- **Sequential Experiments**  
  Independent sequential experiments: geometric, binomial and multinomial probability laws  
  Homogeneous Markov chains

- **Univariate Random variables**  
  Probability functions for random variables (cdf, pdf, pmf)  
  Key discrete probability models (Bernoulli → geometric → binomial → Poisson)  
  Key continuous probability models (rectangular, exponential, m-Erlang, normal)  
  Functions of random variables  
  Expectation

- **Multiple random variables**  
  Marginal and conditional distributions  
  Discrete-continuous case: finite mixture models  
  The bivariate normal distribution  
  Correlation and linear regression  
  Introduction to graphical models

- **Statistics and Data Analysis**  
  Random sampling: the empirical distribution and its moments (sampling statistics)  
  Probability estimation from survey data  
  Quantification of error; quantization noise
## RECOMMENDED TEXT(S)

The main recommended text for the module is:


Secondary recommended texts are as follows:


## LEARNING OUTCOMES

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

- Quantify beliefs in uncertain propositions related to key electrical, computer and bio-engineering contexts, such as noisy communication, bio-imaging, and large assemblies
- Distinguish between the vital 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 contexts
- Evaluate measures of location, spread and dependence for these distributions
- Convert random experimental data (samples and surveys) into quantified beliefs, summarize these data via sampling statistics, and assess dependence between data

## 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 sheets, with solutions provided on the webpage. Students are reminded that attendance at all timetabled activities is compulsory.

## ASSESSMENT MODE(S)

70% of the final mark is determined via the annual examination. The remaining 30% is reserved for continuous assessment, by means of about 5 assignments during the semester, and a one-hour end-of-semester quiz.
### Course Title: Management for Engineers

<table>
<thead>
<tr>
<th>Level: Junior Sophister</th>
<th>Credits: 5</th>
<th>Prerequisites: None</th>
</tr>
</thead>
</table>

**Lecturer(s):** Assistant Prof. Niamh Harty, Ms Joanna Gardiner, Assistant Prof. 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

### Learning Outcomes

On completion of this course the student should 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 end-of-year two-hour written 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.

**WEBSITE**

https://www.tcd.ie/Engineering/Courses/BAI/JS_Subjects/3E3/
Module Title: Signals and Systems
Code: 3C1

Level: Junior Sophister
Credits: 5
Prerequisites: None

Lecturer(s): Assistant Professor David Corrigan

Terms: Semester 1
Lectures/week: 3
Total: 33
Tutorials/week: 1
Total: 11

Aims/Objectives

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

The module contains elements of Signal Processing as a foundation for modules 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 and as such is ubiquitous in modern technology. The module 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 module 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\textsuperscript{nd} Order pole locations.
- Introduction to the root locus.

### Associated Laboratory/Project Programme

**Lab S1:** Analysis of the Transient and Steady State Responses of Dynamic Systems.
**Lab S2:** Control Systems and Signal Analysis in the Frequency Domain.
**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 module the student should 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 module 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.
<table>
<thead>
<tr>
<th><strong>MODULE TITLE:</strong> Digital Circuits</th>
<th><strong>CODE:</strong> EE3C2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LEVEL:</strong> (Junior Sophister)</td>
<td><strong>CREDITS:</strong> 5</td>
</tr>
<tr>
<td><strong>LECTURER(s):</strong> Associate Professor Martin J. Burke</td>
<td><strong>PREREQUISITES:</strong> None</td>
</tr>
<tr>
<td><strong>TERMS:</strong> Semester 1</td>
<td><strong>LECTURES/WEEK:</strong> 3</td>
</tr>
<tr>
<td><strong>DURATION (WEEKS):</strong> 12</td>
<td><strong>TUTORIALS/WEEK:</strong> 1</td>
</tr>
<tr>
<td><strong>TOTAL:</strong> 33</td>
<td><strong>TOTAL:</strong> 11</td>
</tr>
</tbody>
</table>

**AIMS/OBJECTIVES**
Digital Circuits is a one semester module 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 module 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 module, 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 modules, 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, architectural 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 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: the principles of operation of simple CMOS gates.
ASSOCIATED LABORATORY/PROJECT PROGRAMME

**Lab D1:** Bipolar Junction Transistor: the assessment of transistor switching characteristics.
**Lab D2:** TTL Logic Family Characteristics

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

RECOMMENDED TEXT(S)


LEARNING OUTCOMES

On completion of this module the student should be able to:

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

TEACHING STRATEGIES

The module 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 two-hour 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 supplemental examinations.
**Module Title:** Analogue Circuits  
**Code:** EE3C3

**Level:** Junior Sophister  
**Credits:** 5  
**Prerequisites:** None

**Lecturer(s):** Assistant Professor Edmund Lalor

**Terms:** Semester 2  
**Lectures/week:** 3  
**Tutorials/week:** 1  
**Duration (weeks):** 12  
**Total:** 33  
**Total:** 11

**Aims/Objectives**
Analogue Electronics is a one semester module 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 module aims to provide students with knowledge of the 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 module, students will develop the analytical and synthesis skills needed to design analogue circuits for electronic equipment intended for any modern application area.

**Syllabus**
- The MOSFET as Amplifier: Biasing, Small signal analysis, Amplifier Configurations, Frequency Response.
- The BJT as Amplifier: Biasing, Small signal analysis.
- Single Stage IC Amplifiers.
- Differential Amplifiers: MOS Differential Pair, BJT Differential Pair.
- Multistage Amplifiers: Two-stage CMOS Op Amp, the 741 Op Amp.
- Feedback and Stability in Analogue Circuits.
- Digital/Analog and Analog/Digital Converters.
- Oscillators and Other Waveform Shaping Circuits.
- Active Filters.

**Associated Laboratory/Project Programme**
Lab A1: Active Filters: PSpice simulation of an operational amplifier based analogue active filter
Lab A2: 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)**

**Learning Outcomes**
On completion of this module the student will be able to:

1. Explain the operation of the bipolar junction and MOS field effect transistors in terms of their equivalent circuits.
2. Analyze simple linear amplifiers to determine their performance criteria and limitations.
3. Explain Op-Amp ideal and practical characteristics
4. Explain the Principles of Oscillation.
5. Design amplifiers and oscillators based on the performance criteria.

**Teaching Strategies**
The module 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 two-hour 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 supplemental examinations.
Module Title: Applied Electromagnetism | Code: EE3C4

Level: Junior Sophister | Credits: 5 | Prerequisites: None

Lecturer(s): Professor William Coffey

Terms: Semester 2
Duration (weeks): 12
Lectures/week: 3
Total: 33
Tutorials/week: 1
Total: 11

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

Syllabus
Starting from Coulomb’s law of force between charges, this module 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.

<table>
<thead>
<tr>
<th><strong>Recommended Text(s)</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Learning Outcomes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>On completion of this module the student will be able to:</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th><strong>Teaching Strategies</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The module is taught using a combination of lectures, tutorials and two supporting laboratories.</td>
</tr>
</tbody>
</table>

**Module webpages:**

- [http://www.mee.tcd.ie/~wcoffey/teaching](http://www.mee.tcd.ie/~wcoffey/teaching)
- [http://www.tcd.ie/engineering/undergraduate](http://www.tcd.ie/engineering/undergraduate)
Assessment Mode(s)
The two-hour 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 supplemental examinations.
**Module Title:** Telecommunications

**Code:** EE3C5

**Level:** Junior Sophister

**Credits:** 5

**Prerequisites:** None

**Lecturer(s):**
Assistant Professor Nicola Marchetti

**Teaching support (labs):**
Mr. Shane Hunt

**Terms:** Semester 2

**Duration (weeks):** 12

**Lectures/week:** 3

**Total:** 33

**Tutorials/week:** 1

**Total:** 11

**Aims/Objectives**

The aim of the module is to introduce students to the key aspects of radio systems 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., fibre 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., Hamming Coding)
- Analog Modulation (e.g., 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 Analog Modulation
T2 Digital Modulation

**Recommended Text(s)**

2. T. Cover, Elements of Information Theory, Wiley, 2006
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 module 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 channel coding techniques to data and demonstrate the types of errors which can be corrected.
6. Describe and explain a number of analog 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 radios.

## Teaching Strategies

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

## Assessment Mode(s)

Continuous assessment will be adopted. The written examination will contribute 55%, two in-class quizzes will contribute 15% each, 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, including those of the two in-class quizzes, will be carried forward to the supplemental examinations.
**COURSE TITLE:** Electronic Design Project: Analogue

**CODE:** EE3C6a

**LEVEL:** Junior Sophister

**CREDITS:** 5

**PREREQUISITES:** None

**LECTURER(S):** Assistant Professor Edmund Lalor

**TERMS:** Semester 1

**DURATION (WEEKS):** 12

**LABORATORIES:** 10

**TOTAL:** 10

**TUTORIALS:** 0

**TOTAL:** 0

**AIMS/OBJECTIVES**
The main purpose of the analogue design project is 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 three persons, who must manage the project, divide up the workload between them and apply a ‘divide-and-conquer’ approach to tackling a design task in electronic engineering. Thus, the project involves constructing circuits from components in modular form and testing and documenting each section before proceeding further. This simplifies the task of verifying and modifying the design.

**SYLLABUS**
- Measurement of the input and output characteristics of an NPN junction transistor.
- Use of the transistor as a simple common emitter amplifier and also in a difference amplifier configuration.
- Use of the 741 Op-Amp and IC audio amplifiers to amplify the output from a personal stereo or MP3 player.
- Investigation of the ECG signal and how to amplify it for the purpose of observing a subject’s heartbeat.
- Design and construction of an instrumentation amplifier.
- Transmission of the audio signal using a simple AM radio kit.

**RECOMMENDED TEXT(S)**

**LEARNING OUTCOMES**
On successful completion of this project the student will be able to:
- Describe and plan a project involving analogue electronics.
- Construct a hardware solution for an analogue electronics problem.
- Sketch a schematic diagram of the circuit along with 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 op-amps which include both construction of circuits and MULTISIM simulations. Support is on hand from the demonstrator and technical officers 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 MULTISIM. Students are then required to research, and construct a far more complex amplifier capable of amplifying 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.
**MODULE TITLE:** Electronic Design Project: Digital

**CODE:** EE3C6b

**LEVEL:** Junior Sophister

**CREDITS:** 5

**PREREQUISITES:** None

**LECTURER(S):** Assistant Professor Nicola Marchetti

**TEACHING SUPPORT:**
- Mr. Robert Dempsey
- Mr. Conor Nolan

**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 MULTISIM, 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 TEXT(S)**

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

- Describe 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. The project is launched from introductory laboratory exercises with CMOS ICs. Support is on hand from the demonstrator and technical officers 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.
<table>
<thead>
<tr>
<th><strong>Module Title:</strong> Digital Systems Design</th>
<th><strong>Code:</strong> 3C7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level:</strong> Junior Sophister</td>
<td><strong>Credits:</strong> 5</td>
</tr>
<tr>
<td><strong>Prerequisites:</strong> None</td>
<td><strong>Prerequisites:</strong> None</td>
</tr>
<tr>
<td><strong>Lecturer(s):</strong> Assistant Professor</td>
<td><strong>Teaching Assistant:</strong> TBA</td>
</tr>
<tr>
<td>Naomi Harte</td>
<td></td>
</tr>
<tr>
<td><strong>Terms:</strong> Semester 1</td>
<td><strong>Lectures/week:</strong> 2</td>
</tr>
<tr>
<td>** Duration (weeks):** 12</td>
<td><strong>Practicals/week:</strong> 2</td>
</tr>
<tr>
<td><strong>Total:</strong> 22</td>
<td><strong>Total:</strong> 22</td>
</tr>
</tbody>
</table>

**Aims/Objectives**
This module introduces digital systems design with a focus on FPGA design. The module will build on the basics of digital 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**
The student 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 given during module.
### Learning Outcomes

On completion of this module the student should 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.
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.

### Teaching Strategies

This is a highly practical module. 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, 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 extra hours 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 module practical work will contribute 50% of the overall grade. The practical element of the module **CANNOT** be repeated. The practical mark will be carried forward for all 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.

**Late assignments policy:**

- Lose half of marks if up to 1 week late.
- No marks if over 1 week late.
**Aims/Objectives**

Microprocessor Systems 1 is a one-semester module 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.
**Recommended Text(s)**

Additional recommended texts:

**Website**

**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 module, 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 module 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.
<table>
<thead>
<tr>
<th><strong>Module Title:</strong> Microprocessor Systems 2</th>
<th><strong>Code:</strong> CS3D2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level:</strong> Junior Sophister</td>
<td><strong>Credits:</strong> 5</td>
</tr>
<tr>
<td><strong>Lecturer(s):</strong> Assistant Prof. Mike Brady</td>
<td><strong>Prerequisites:</strong> None</td>
</tr>
<tr>
<td><strong>Terms:</strong> Semester 2</td>
<td><strong>Lectures/week:</strong> 3</td>
</tr>
<tr>
<td><strong>Duration (weeks):</strong> 12</td>
<td><strong>Tutorials/week:</strong> 1</td>
</tr>
<tr>
<td></td>
<td><strong>Total:</strong> 33</td>
</tr>
<tr>
<td></td>
<td><strong>Total:</strong> 11</td>
</tr>
</tbody>
</table>

**Module Description/Aim**

Students bring the knowledge and expertise of programming, digital logic and some electronics to the development of small system integration projects combining interface design and breadboarding with program design and implementation. Each team is provided with a small ARM-based computer, a PC-based integrated development system, some electronic components and breadboarding facilities. Students have to design, build and test integrated hardware and software systems to provide a certain required functionality.

**Module Content**

- Microprocessor system overview
- Microprocessor architecture overview
- Memory hierarchy
- Interfacing with peripheral devices
- Device programming
- Interrupt and Exception Handling

**Recommended Texts**


**Learning Outcomes**

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

1. Select peripherals and software architectures most suited to implementing each project.
2. Visualise complex exception processing that implement system calls and interrupts in order to recognise potential problem sources and to develop a debugging strategy that tries to isolate the problem further until the cause is identified.
3. Work in small teams to design and implement the hardware and software aspects of their projects by dividing the tasks and responsibilities among the group members.
<table>
<thead>
<tr>
<th><strong>Teaching Strategies</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The teaching strategy is a mixture of lectures, problem-solving tutorials and two-hour practicals.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Assessment Mode(s)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment is by examination and continuous assessment. Continuous assessment may account for up to 20% of the full assessment with the remainder from the examination.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Website</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.scss.tcd.ie/CourseModules/CS3D2/">http://www.scss.tcd.ie/CourseModules/CS3D2/</a></td>
</tr>
</tbody>
</table>
### Module Title: Computer Networks  
**Code:** CS3D3

<table>
<thead>
<tr>
<th><strong>Level:</strong> Junior Sophister</th>
<th><strong>Credits:</strong> 5</th>
<th><strong>Prerequisites:</strong> None</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lecturer(s):</strong> Assistant Prof. Hitesh Tewari</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Terms:</strong> Semester 2</th>
<th><strong>Lectures/week:</strong> 3</th>
<th><strong>Tutorials/week:</strong> 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duration (weeks):</strong> 12</td>
<td><strong>Total:</strong> 33</td>
<td><strong>Total:</strong> 11</td>
</tr>
</tbody>
</table>

### Aims/Objectives
This module 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 module 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)
2. Computer Networking and the Internet (5th edition), Fred Halsall, Addison Wesley

### Learning Outcomes
Upon completion of this module, 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
5. Have a working knowledge of datagram and internet socket programming
**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 module notes with a few gaps that they fill in during the module 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 two-hour examination and by assessment of the practical sessions. Practical sessions (the module 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.
Module Title: Operating Systems and Concurrent Systems  
Code: CS3D4

Level: Junior Sophister  
Credits: 5  
Prerequisites: None

Lecturer(s): Assistant Prof. Mike Brady

Terms: Semester 2  
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
**Recommended Text(s)**

I have been using material and ideas from these books. You may find them useful too.

- Bovet and Cesati, "Understanding the Linux Kernel, Third Edition", O'Reilly Media, 2005

This is a very comprehensive reference work:

**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. develop models of concurrent programs using the Promela modelling language;
4. verify the correctness of concurrent programs of moderate complexity using SPIN;
5. describe algorithms, data structures and policies used in modern operating systems for thread scheduling, memory management, disk I/O and file management;
6. evaluate, compare and reason about the relative performance of algorithms used by operating systems (e.g. page replacement policies);
7. apply a knowledge of operating system behaviour when developing user-level programmes.

**Teaching Strategies**

The module 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)**

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

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

**Website**

[http://www.scss.tcd.ie/CourseModules/CS3D4/](http://www.scss.tcd.ie/CourseModules/CS3D4/)
**Module Title:** Software Design and Construction

**Code:** CS3DS5

<table>
<thead>
<tr>
<th>Level:</th>
<th>Junior Sophister</th>
<th>Credits:</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites:</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lecturer(s):</th>
<th>Mr. Glenn Strong</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Terms:</th>
<th>Semester 1 and 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (weeks):</td>
<td>24</td>
</tr>
<tr>
<td>Lectures/week:</td>
<td>1</td>
</tr>
<tr>
<td>Total:</td>
<td>22</td>
</tr>
<tr>
<td>Tutorials/week:</td>
<td>1</td>
</tr>
<tr>
<td>Total:</td>
<td>22</td>
</tr>
</tbody>
</table>

**Aims/Objectives**

Software Design and Implementation is a two-semester module; 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 module 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 module 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 module 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 module 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 time is forty-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 forty-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 module, 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 a more fully rounded project implementation.

Assessment Mode(s)

Assessment of this module 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 module lecturer. Progress towards these milestones are not formally included in the student's 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);

- 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.
### Aims/Objectives

The lectures and tutorials treat the detailed design and organisation of microprocessor.

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 module are to learn register-transfer specification and design and learn the fundamentals of an instruction processor.

### 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

### Recommended Text(s)

- Introductory VHDL: From Simulation to Synthesis
- Logic and Computer Design Fundamentals” 2nd Edition updated, Mano

### 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;
Assessment Mode(s)

Assessment is by examination (80%) and continuous assessment (20%).
Continuous assessment is composed of a number of marked laboratory exercises and two substantial assignments.
<table>
<thead>
<tr>
<th><strong>Module Title:</strong> Applying Probability: Introduction</th>
<th><strong>Code:</strong> ST2004</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level:</strong> Junior Sophister</td>
<td><strong>Credits:</strong> 5</td>
</tr>
<tr>
<td><strong>Lecturer(s):</strong> Prof. John Haslett</td>
<td></td>
</tr>
<tr>
<td><strong>Terms:</strong> Semester 1</td>
<td><strong>Lectures/week:</strong> TOTAL:</td>
</tr>
<tr>
<td><strong>Duration (weeks):</strong> 12</td>
<td><strong>Labs/week:</strong> TOTAL:</td>
</tr>
</tbody>
</table>

### Aims/Objectives

An introduction to probability, intended as preparatory for later course

### Syllabus

Specific topics addressed in this module include:

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

### Recommended Text(s)

**Main Text:**


**Other recommended reading**

- 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
## Learning Outcomes
When students have successfully completed this module they should:

- Understand the basic ideas of probability modelling; i.e., how to describe uncertainty by probability statements, and how to combine uncertainties to make statements about the uncertainty of systems.

## Assessment Mode(s)
Two group assignments and one two-hour examination.
Electronic Engineering Labs

Laboratory Programme Coordinator:

Dr. David Corrigan, Assistant Professor, 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: The deadline for handing up your report is 1 week after completion of the lab unless otherwise stated by the relevant lecturer. Reports are submitted by placing them in the marked box in the PC Lab on the first floor of the printing house. The box will be emptied once a week and you will receive an email acknowledgement of your submission.

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