TRINITY COLLEGE DUBLIN

School of Computer Science and Statistics

BA Moderatorship in Computer Science
Master in Computer Science

Course Handbook
2011/2012

Funded by the Irish Government under the National Development Plan 2007–2013
1 Table of contents

Contents

1 Table of contents 3

2 A Note on this Handbook 6

3 Introduction 7

4 General Information 7
  4.1 Trinity College Dublin 7
  4.2 The School of Computer Science and Statistics 7
  4.3 School Contact Details 8
  4.4 Academic and Administrative Staff 8
  4.5 Student Contact 9

5 The Degree Programme 9
  5.1 Structure of the Programme 10
    5.1.1 European Credit Transfer System 11
  5.2 Programme Aims 12

6 Year 1 – Junior Freshman Year 12
  6.1 CS1001 Mathematics I 13
  6.2 CS1002 Mathematics II 13
  6.3 CS1011 & CS1012 Introduction to Programming I & II 13
  6.4 CS1021 Introduction to Computing I 14
  6.5 CS1022 Introduction to Computing II 14
  6.6 CS1023 Digital Logic Design I 14
  6.7 CS1024 Digital Logic Design II 15
  6.8 CS1025 Electrotechnology 15
  6.9 CS1031 Telecommunications I 15
  6.10 CS1013 Programming Project I 15
  6.11 CS1081 Computers and Society 16

7 Year 2 – Senior Freshman Year 16
  7.1 MA2C01 Discrete Mathematics I 16
  7.2 MA2C02 Discrete Mathematics II 16
  7.3 CS2011 Programming Techniques 17
  7.4 CS2012 Programming Techniques II 17
  7.5 CS2013 Programming Project II 17
<table>
<thead>
<tr>
<th></th>
<th>Course Title</th>
<th>Year</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.6</td>
<td>CS2014 Systems Programming I</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>7.7</td>
<td>CS2015 Systems Programming II</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>7.8</td>
<td>CS2021 Microprocessor Systems</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>7.9</td>
<td>CS2022 Computer Architecture II</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>7.10</td>
<td>CS2031 Telecommunications II</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>7.11</td>
<td>CS2041 Information Management I</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td><strong>Year 3 – Junior Sophister Year</strong></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>8.1</td>
<td>CS3011 Symbolic Programming</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>8.2</td>
<td>CS3012 Software Engineering</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>8.3</td>
<td>CS3013 Software Engineering Group Project</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>8.4</td>
<td>CS3014 Concurrent Systems I</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>8.5</td>
<td>CS3015 Concurrent Systems II</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>8.6</td>
<td>CS3016 Introduction to Functional Programming</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>8.7</td>
<td>CS3017 Introduction to the Semantics of Formal Languages</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>8.8</td>
<td>CS3021 Computer Architecture III</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>8.9</td>
<td>CS3031 Advanced Telecommunications</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>8.10</td>
<td>CS3041 Information Management II</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>8.11</td>
<td>CS3051 Foundations of Visual Computing</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>8.12</td>
<td>CS3061 Artificial Intelligence I</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>8.13</td>
<td>CS3071 Compiler Design I</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>8.14</td>
<td>ST1002 Statistical Analysis</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td><strong>Year 4 – Senior Sophister Year</strong></td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>9.1</td>
<td>Year 4 – Moderatorship Only</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>9.2</td>
<td>Year 4 – MCS Programme</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>9.3</td>
<td>CS4051 Human Factors</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>9.4</td>
<td>CS4081 Technology Entrepreneurship</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>9.5</td>
<td>CS4098 Group Computer Science Project</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>9.6</td>
<td>CS4099 Final Year Project</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>9.7</td>
<td>CS7091 Industrial / Research Lab Internship</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>10</td>
<td><strong>Year 5 (MCS)</strong></td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>10.1</td>
<td>CS7039 Research Methods</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>10.2</td>
<td>CS7092 MCS Dissertation</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>11</td>
<td><strong>Senior Sophister (Year 4) and Year 5 Options</strong></td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>11.1</td>
<td>CS4001 Fuzzy Logic</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>11.2</td>
<td>CS4003 Formal Methods</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>11.3</td>
<td>CS4012 Topics in Functional Programming</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>11.4</td>
<td>CS4031 Mobile Communications</td>
<td></td>
<td>28</td>
</tr>
</tbody>
</table>
11.5 CS4032 Distributed Systems ................................................. 29
11.6 CS4052 Computer Graphics .................................................. 29
11.7 CS4053 Computer Vision ...................................................... 30
11.8 CS4061 Artificial Intelligence IIa ............................................ 30
11.9 CS4062 Artificial Intelligence IIb ............................................ 30
11.10 CS4071 Compiler Design II ................................................ 31
11.11 CS7003 Middleware for Distributed Systems ......................... 31
11.12 CS7004 Embedded Systems ............................................... 31
11.13 CS7008 Vision Systems ...................................................... 31
11.14 CS7009 Networked Applications ......................................... 32
11.15 CS7012 Management of Networks and Distributed Systems ........ 32
11.16 CS7030/CS7058 Numerical Methods and Advanced Mathematical Modelling I/II ......................................................... 32
11.17 CS7031 Graphics and Console Hardware ................................ 32
11.18 CS7033 Real-time Animation .............................................. 33
11.19 CS7048 Data Communications and Wireless Networking .......... 33
11.20 CS7052 Sustainable Computing ............................................ 33

12 Prizes .......................................................... 33
12.1 Prizes in the Computer Science Programme .......................... 34
   12.1.1 The Victor W. Graham Prize .............................................. 34
   12.1.2 The Ludgate Prize .......................................................... 34
   12.1.3 The William Nurock Prize ............................................... 34
12.2 Scholarship ............................................................... 34

13 Regulations .......................................................... 35
13.1 College Regulations ......................................................... 35
13.2 Attendance and Participation .............................................. 35
13.3 Non-Satisfactory Attendance or Performance ........................ 35
13.4 Retaining Coursework ....................................................... 36
13.5 Plagiarism ................................................................. 36
13.6 Identifying Plagiarism ....................................................... 37
13.7 Examinations ............................................................... 37
13.8 Progression ................................................................. 38
   13.8.1 Annual Examinations ...................................................... 38
   13.8.2 Supplemental Examinations ............................................. 39
13.9 Module Assessment ......................................................... 39
   13.9.1 Annual Examinations ...................................................... 39
   13.9.2 Supplemental Examinations ............................................. 40
13.10 Progression to Year 4 of the Masters Programme .................. 40
13.11 Progression to Year 5 of the Masters Programme .................. 40
A Note on this Handbook

The Computer Science degree programme leads to a BA Moderatorship in Computer Science degree after four years and a Master in Computer Science (MCS) degree after five. This handbook contains information and regulations for all Computer Science degree programme students in the 2011/2012 academic year. It provides a guide to what is expected of you on this programme, and the academic and personal support available to you. Please retain it for future reference.

Information provided in this handbook is accurate at time of preparation. Any necessary revisions will be notified by college email. Please note that, in the event of any conflict or inconsistency between the General Regulations published in the University Calendar and information contained in course handbooks, the provisions of the General Regulations will prevail. The University Calendar is available at http://www.tcd.ie/calendar/.
This handbook is available from the School of Computer Science and Statistics website. A hard copy of this document is available from the School Reception office on request.

3 Introduction

Welcome to the Computer Science degree programme offered by the School of Computer Science and Statistics.

We are confident that you will find this programme challenging and demanding but we also hope that you will find your studies at Trinity College Dublin both stimulating and rewarding. Our courses have been designed to offer students a dynamic, structured and coherent learning experience. Our programme has several features which we believe will contribute to your studies being an effective and enjoyable period of personal and academic developments.

If you are an incoming student, we invite you to read First Year in University, on page 48, which may help you understand what you need to do to have an enjoyable and productive time at college.

We wish you every success in the coming year.

Mike Brady, Jonathan Dukes, Jeremy Jones.

4 General Information

4.1 Trinity College Dublin

Trinity College Dublin (TCD)—the College of the Holy and Undivided Trinity of Queen Elizabeth near Dublin—was founded in 1592 by Queen Elizabeth I. Trinity is sometimes referred to as the University of Dublin or Dublin University. Today, Trinity has approximately 800 academics catering for more than 16,000 students (of which a third are postgraduates). TCD is recognized internationally as Ireland’s premier university and is ranked in the top 100 world universities and amongst the top 50 European universities.

4.2 The School of Computer Science and Statistics

The School of Computer Science and Statistics (SCSS) was formed in 2005 by the amalgamation of the Department of Computer Science and the Department of Statistics. The School has more than 60 academic staff and more than 200 full-time postgraduate students and support staff. It comprises five academic disciplines:
• **Computer Systems** undertakes systems research at the hardware/software interface and has a particular research focus on telecommunications and networked computer systems.

• **Information Systems** studies the impact of Information and Communication Technology on society and business with a particular focus on the impact of technology on learning.

• **Intelligent Systems** has a research focus is on computational issues related to perception, cognition, decision and interaction by and between systems and their human users.

• **Software Systems** has significant strength in programming language and software technologies and the formal foundations that underlie them.

• **Statistics** provides the School’s research strength in statistical learning techniques and in modelling uncertainty.

### 4.3 School Contact Details

The School Reception office is located beside Room G.8 in the O’Reilly Institute. Opening hours during lecture terms are 9:00 am to 11:00 am, 11:30 am to 1:00 pm and 2:00 pm to 5:00 pm.

Tel: 8961765  
Fax: 6772204  
Email: enquiries@scss.tcd.ie  
Web: http://www.scss.tcd.ie/

The postal address of the school is: **School of Computer Science and Statistics, O’Reilly Institute, Trinity College Dublin, Dublin 2.**

The noticeboard for this programme is located beside School Reception in the O’Reilly Institute.

### 4.4 Academic and Administrative Staff

<table>
<thead>
<tr>
<th>Name</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr Jeremy Jones</td>
<td>Head of School</td>
</tr>
<tr>
<td>Dr Mike Brady</td>
<td>Director of the BACS/MCS Programme</td>
</tr>
<tr>
<td>Dr Jonathan Dukes</td>
<td>Coordinator of the BACS/MCS Programme (Years 1, 2 &amp; 3)</td>
</tr>
<tr>
<td>Dr Andrew Butterfield</td>
<td>Director of Undergraduate Teaching and Learning</td>
</tr>
<tr>
<td>Prof. Simon Wilson</td>
<td>Director of Postgraduate Teaching and Learning</td>
</tr>
<tr>
<td>Ms. Lynn Daly</td>
<td>Executive Officer, Teaching Support Unit</td>
</tr>
</tbody>
</table>
The BACS/MCS director and coordinator are Mike Brady and Jonathan Dukes, who can be contacted by telephone at 8961786 and 8968421 respectively, or by email at brady@scss.tcd.ie or jdukes@scss.tcd.ie. Please note that in the first instance all enquiries regarding modules, assignments, feedback and supervision should be directed to the administrative staff in the Teaching Support Unit who will then, where appropriate, inform the director and coordinator.

4.5 Student Contact

The College and the School of Computer Science and Statistics will communicate with you by email. This will be done using your college email address (i.e. your tcd.ie account) which you will receive when you register. You are advised to check your email account regularly or to set up a forwarding facility on your TCD account. Instructions can be found at: http://isservices.tcd.ie.

5 The Degree Programme

The Computer Science degree programme began in 1979 as a four-year course leading to an honors degree in Computer Science. Honors\(^1\) degrees are traditionally called Moderatorships in Trinity, thus the formal title of the degree is *B.A. (Mod.) in Computer Science*.

Starting in 2011, the programme offers, in addition, the degree of *Master in Computer Science* over a five-year course.

---

\(^1\)This is the correct spelling of the word when applied to degrees awarded in TCD.
5.1 Structure of the Programme

The diagram shows the five-year composition of the programme. Students typically enter Year 1 from secondary school via the Central Applications Office (CAO) system. Suitably qualified students may also join the programme in Year 3 or Year 4.

Students normally exit the programme at the end of Year 4, with a B.A. (Mod.) in Computer Science degree, or at the end of Year 5, with a Master in Computer Science degree in addition to the B.A. (Mod.) in Computer Science degree. Provision is made for students to exit the programme with an ordinary B.A. degree at the end of Year 3.

The terms Junior Freshman, Senior Freshman, Junior Sophister and Senior Sophister are used to refer to a first-year, second-year, third-year and fourth-year student respectively; thus, for example, Junior Freshman year, (or JF year), refers to first year.

In the Freshman years instruction is given in the theoretical underpinnings of computer science along with courses in hardware and software. In the Sophister years students may select a number of options in addition to core courses. Students participating in the MCS course are required to engage in a one semester internship in industry or in a university research laboratory in their senior sophister year. In the fifth year students undertake a significant project with a substantial element of
independent research leading to a dissertation. There are laboratory classes in each year of the programme.

The teaching year is divided into two twelve-week semesters. The first semester is the Michaelmas Term, the second is the Hilary Term\(^2\). The seventh week of each semester is a reading week, during which no lectures are held.

Subjects are taught in modules. Modules are generally taught for one semester, and consist of lectures, tutorials, seminars, and laboratory sessions. All students on the programme take the same modules in Year 1, Year 2 and the first semester of Year 3. From the second half of Year 3 onwards, students take some compulsory modules and a selection of elective modules. Each module is assigned an European Credit Transfer System (ECTS) rating. Modules in the first three years are each assigned five ETCS points. Modules and other parts of the programme in Year 4 and Year 5 may have a rating of more than five ECTS points each.

### 5.1.1 European Credit Transfer System

The European Credit Transfer System (ECTS) is an academic credit transfer and accumulation system representing the student workload required to achieve the specified objectives of a study programme.

The ECTS weighting for a module is a measure of the student input or workload required for that module, based on factors such as the number of contact hours, the number and length of written or verbally presented assessment exercises, class preparation and private study time, laboratory classes, examinations, clinical attendance, professional training placements, and so on as appropriate. There is no intrinsic relationship between the credit volume of a module and its level of difficulty.

In College, one ECTS unit is defined as 20–25 hours of student input so a five-credit module will be designed to require 100–125 hours of student input including class contact time and independent or group work. Each year of the programme is composed of modules worth a total of 60 credits. Where there is the option to choose from a range of modules, it is the responsibility of the student to ensure that they successfully complete modules worth 60 credits.

ECTS credits are awarded to a student only upon successful completion of the course year. Progression from one year to the next is determined by the course regulations. Students who fail a year of their course will not obtain credit for that year even if they have passed certain component modules. Exceptions to this rule are one-year and part-year visiting students, who are awarded credit for individual modules successfully completed.

\(^2\)The names come from traditional Christian feast days: Michaelmas Day is September 29 and St. Hilary’s Day is January 13.
5.2 Programme Aims

At the end of the programme the students should be able to:

- Develop and apply computer systems from a broad base of knowledge in mathematics, computer science, computer technology and human factors.

- Identify and formulate advanced technical challenges and demonstrate judgement to design appropriate computer science solutions.

- Design systems, components or processes to meet specified functional objectives and to measure and analyse performance against these objectives.

- Understand and express the role of computer science in the community including the need for high standards of ethical behaviour and professional responsibility.

- Work effectively, independently and within multidisciplinary teams, and act as a mentor in team settings and engage in lifelong learning.

- Communicate effectively both professionally with other computing professionals and with the wider community.

- Participate in contemporary research activity as appropriate and demonstrate the knowledge and skills needed to undertake independent research.

6 Year 1 – Junior Freshman Year

First year is called *Junior Freshman* (JF) year in Trinity. In this section is a list of Junior Freshman subject modules and a brief description of each. Full details are linked to the junior freshman web page at http://www.scsc.tcd.ie/undergraduate/ba/current/junior-freshman.php.

<table>
<thead>
<tr>
<th>Michaelmas Term</th>
<th>Hilary Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS1001 Mathematics I</td>
<td>CS1002 Mathematics II</td>
</tr>
<tr>
<td>CS1011 Introduction to Programming I</td>
<td>CS1012 Introduction to Programming II</td>
</tr>
<tr>
<td>CS1021 Introduction to Computing I</td>
<td>CS1022 Introduction to Computing II</td>
</tr>
<tr>
<td>CS1023 Digital Logic Design I</td>
<td>CS1024 Digital Logic Design II</td>
</tr>
<tr>
<td>CS1025 Electrotechnology</td>
<td>CS1031 Telecommunications I</td>
</tr>
<tr>
<td>CS1013 Programming Project I</td>
<td>CS1081 Computers and Society</td>
</tr>
</tbody>
</table>
6.1 CS1001 Mathematics I

Mathematics is of interest to computer scientists due to the fact that it is both practical and theoretical in nature. Not only does it have a myriad of applications (e.g. in wireless communications and computer graphics), it is also of intrinsic interest to theoretical computer scientists. The mathematical techniques learned as part of this module have wider applications in areas as diverse as Business (e.g. for modelling volatility and risk), Economics and Engineering (e.g. for structural monitoring).

This module aims to reflect these properties by providing students with an introduction to the mathematics, both continuous and discrete, which lies at the foundation of many real-world applications in Computer Science, Engineering and the Social Sciences.

This module aims to develop the students’ skills and abilities in the mathematical methods necessary for solving practical problems. One of the key objectives for this module is to introduce students to the learning styles needed for university level mathematics. Students will be encouraged to develop the independent, reflective learning skills needed for success at University level.

6.2 CS1002 Mathematics II

This module aims to introduce students to discrete mathematics and mathematical logic with applications to computer science. In particular, it will introduce basic set operations, discrete maths functions in Number Theory that are used in computer science and also program verification as an application of mathematical logic. The course is influenced by the approaches of Backhouse, Dijkstra and Gries.

6.3 CS1011 & CS1012 Introduction to Programming I & II

These modules provide an introductory course in computer programming. A practical approach is taken to teaching the fundamental concepts of computer programming, with a strong emphasis on tutorial and laboratory work and provide an important vehicle for developing students’ analytical and problem-solving skills.

The aim is to give students an understanding of how computers may be employed to solve real-world problems. Specifically, students are introduced to the object-oriented approach to program design and are taught how to write programs in an object-oriented language (in this case, the language is Java).

Students also have the opportunity to reinforce their problem solving and programming skills by developing solutions to programming problems and implementing those solutions as object-based programs.
6.4 CS1021 Introduction to Computing I

This module provides students with an introduction to the basic structure and behaviour of microprocessor systems. By designing, 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 introduces students to certain concepts, which are fundamental to the study of Computer Science, including binary number systems and the representation of basic information such as values and text.

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.

6.5 CS1022 Introduction to Computing II

This module continues directly from CS1021 (which is a prerequisite) and examines the structure and behaviour of microprocessor systems in greater depth. In particular, this module will introduce students to the implementation of subroutines and simple data structures, handling exceptions and interrupts and simple I/O.

Students will have the opportunity to further reinforce their problem solving, programming and written communication skills by developing solutions to programming problems of increasing complexity through the use of decomposition, implementing those solutions, first in the form of high-level programming constructs and then as one or more assembly language subroutines, which must be documented and tested.

6.6 CS1023 Digital Logic Design I

The lectures do not assume any prior knowledge of the subject, and build gradually in difficulty towards the end of the course. Starting with the theoretical foundations of logic, the students learn about combinatorial logic and how it can be used to construct logic functions that are useful in computing systems. They learn that feedback around combinatorial logic introduces asynchronous sequential behaviour that is the basis for latches and gated latches. The focus is on laying the groundwork for the hardware courses in the second year. Care is taken that the students realize
the subject applies to both computer software and hardware. Laboratory experiments reinforce the concepts as well as adding variety and introducing practical elements.

6.7 CS1024 Digital Logic Design II

This module assumes prior knowledge of the subject material of CS1023 Digital Logic Design I as a pre-requisite. The lectures build gradually in difficulty towards the end of the course. Starting with edge-triggered flip-flops, the students learn about synchronous sequential logic and finally algorithmic state machines. The focus is on laying the groundwork for the hardware courses in the second year. Care is taken that the students realize the subject applies to both computer software and hardware. Laboratory experiments reinforce the concepts as well as adding variety and introducing practical elements.

6.8 CS1025 Electrotechnology

The Electrotechnology module examines the structure of matter from an electrical perspective dealing with conductors, dielectrics and semiconductors. Concepts of electrical charge, electric and magnetic fields are also examined. With this material concepts of voltage, current, impedance in electrical circuits are explored. Circuit elements such as the resistor, capacitor, inductor and semiconductor diode are examined as well as ideal and practical AC and DC supplies. Methods of analyzing electrical circuits consisting of the above components are examined.

6.9 CS1031 Telecommunications I

The Telecommunications module begins with an overview of networks, their topologies and how they are categorized. The relevant standards organizations and the OSI and TCP/IP reference models are discussed. Concepts of frequency domain, bandwidth and channel characteristics are introduced along with Fourier analysis. Transmission media, guided and wireless are also examined. Finally various encoding, modulation and multiplexing schemes are examined in detail.

6.10 CS1013 Programming Project I

This module concentrates on development of practical programming ability through example-based lecturing coupled with intensive laboratory sessions. The emphasis throughout is on producing working programs, starting with interactive graphical applications and moving on to construction of a larger group project involving a data visualisation task.
6.11 CS1081 Computers and Society

This module will allow students to develop an awareness of, and techniques for studying, the way in which information technology and society interact and influence each other, as well as developing crucial communication and team-work skills.

When students have successfully completed this module they should have a deeper awareness of the ways in which information technology and society interact and influence each other. They should also have improved and developed their communication and team-work skills.

7 Year 2 – Senior Freshman Year

Second year is called Senior Freshman (SF) year in Trinity. In this section is a list of senior freshman subject modules and a brief description of each. Full details are linked to the senior freshman web page at http://www.scss.tcd.ie/undergraduate/ba/current/senior-freshman.php.

<table>
<thead>
<tr>
<th>Michaelmas Term</th>
<th>Hilary Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS2011 Programming Techniques I</td>
<td>CS2021 Microprocessor Systems</td>
</tr>
<tr>
<td>CS2012 Programming Techniques II</td>
<td>CS2022 Computer Architecture</td>
</tr>
<tr>
<td>CS2014 Systems Programming I</td>
<td>CS2015 Systems Programming II</td>
</tr>
<tr>
<td>CS2031 Telecommunications II</td>
<td>CS2041 Information Management I</td>
</tr>
<tr>
<td>MA2C01 Discrete Mathematics I</td>
<td>MA2C02 Discrete Mathematics II</td>
</tr>
<tr>
<td></td>
<td>CS2081 Broad Curriculum—Michaelmas Term or Hilary Term*</td>
</tr>
<tr>
<td>CS2081 Broad Curriculum—Michaelmas Term</td>
<td>CS2013 Programming Project II</td>
</tr>
</tbody>
</table>

*See http://www.tcd.ie/Broad_Curriculum/administration/timetable.php

All modules have an ECTS weighting of 5 points.

7.1 MA2C01 Discrete Mathematics I

This module provides students with an introduction to Discrete Mathematics.

Students are exposed to diverse course material presented in the formal style and language that is commonplace in contemporary mathematics, with the aim that they should develop the skills required to engage effectively with such material.

7.2 MA2C02 Discrete Mathematics II

This module provides students with an introduction to topics in Discrete Mathematics, and to more advanced calculus topics relevant to electronics, acoustics and image processing.
Students are exposed to diverse course material presented in the formal style and language that is commonplace in contemporary mathematics, with the aim that they should develop the skills required to engage effectively with such material.

### 7.3 CS2011 Programming Techniques

The overall aim of this module is for students to learn advanced object-oriented programming abstractions and techniques for building important software programs.

This module follows on directly from CS1011 and CS1012 (Introduction to Programming I and II), and gives the students a solid grounding in programming using object orientation. It reinforces the concepts studied in first year programming and extends them to cover more advanced topics, such as inheritance, access modifiers, polymorphism, generics and design pattern.

This is a practical module and hence has many practical assignments as students are expected to be able to make use of the concepts taught.

### 7.4 CS2012 Programming Techniques II

This module is concerned with the construction of reliable, efficient and readable Java programs based on the approach of Design by Contract (i.e. using assertions) as well as deterring students from defensive programming.

The programming development environment will be based on BlueJ.

The aims will be achieved through developing appropriate abstractions and techniques for problem solving. This will be facilitated by re-using components from the Java class libraries.

### 7.5 CS2013 Programming Project II

In this module, students are introduced to the discipline of software engineering and must work in groups to complete a complex software project. Groups will be managed in conjunction with more senior students taking the CS3013 module. For the duration of this module, students are divided into groups, each of which is closely supervised by the module lecturer and a separate project customer. The module provides students with their first formal experience of group work. They are required to follow a rigorous process consisting initially of requirements gathering, analysis and system design. Thereafter they are required to implement a complex software product using industry standard software engineering tools and methodologies.

The principal aim of this module is to provide students with experience of working together in groups to complete a complex software project. Upon completion of the module students will have gained experience of analysing, specifying, designing and implementing a complete software system. They will also have been exposed to the
challenges posed by working in teams and the need to communicate effectively both within their respective groups and to their project supervisor.

7.6 CS2014 Systems Programming I

Students taking this module have already successfully completed courses in object-oriented Java programming and ARM assembly language programming. This module starts with a new perspective on program construction, with structured programming in C. This part of the module deals with the features of C, sound design principles for structured programming, and the necessary self-discipline required to program in a low-level language like C. Students also learn the basics of how programs are commonly implemented, with special emphasis on the layout of program data in memory. In parallel with learning C programming, the students learn about the UNIX operating system, including program development tools, interaction between C programs and the operating system.

7.7 CS2015 Systems Programming II

This module is a continuation of CS2014 and continues the study of Unix, looking at such aspects as shell script programming, and regular expressions. UNIX programming continues, but this time using the C++ programming language. We also cover more difficult topics in C++ programming such as memory management, templates and the Standard Template Library (STL). The course involves extensive lab work, which includes work on modifying existing large programs written by others, and the students re-implementing parts of the STL.

7.8 CS2021 Microprocessor Systems

This course builds on learning outcomes from many other modules: 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.

7.9 CS2022 Computer Architecture II

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.

- A processor unit (ALU + shifter + fast registers) design and simulation,
- 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.

7.10 CS2031 Telecommunications II

Telecommunications II follows on from CS1031 taken by junior freshman students. It requires students to have a basic understanding of physical media that are used for communication and modulation of signals on these media. The module is structured following the Open Systems Interconnect (OSI) model and consists of two parts.

The first part of the module focuses on the concepts and mechanisms that are employed in the 2nd layer of the OSI model, the data link layer. This layer is concerned with the delivery of data between two immediately connected devices i.e. devices that share a common physical medium. The layer coordinates the access to the physical medium and attempts to detect and correct errors introduced by the transfer of signals over the physical medium. The concepts that are employed in this layer are discussed and the students exercises that demonstrate the application of these concepts.

The second part of the module focuses on the 3rd and 4th layer of the OSI model. The 3rd layer, the network layer, focuses on the connection of local area networks (LANs). This layer employs concepts that hide the communication through a LANs and provide an abstraction that allows the communication across various interconnected LANs. This abstraction forms the foundation for todays internet and represents essential knowledge for todays computer science graduates. The 4th layer, the transport layer, provides services such as reliable transport to applications. The understanding of the mechanisms employed in this layer is essential to the understanding of the implementations and performance of current network technology.

7.11 CS2041 Information Management I

Information Management I focuses on the methods and techniques for efficient management (storage, manipulation and retrieval) of data and information in a
computer and on the world wide web. It provides a foundation for later modules in database management and advanced information retrieval on the web.

The first part of the course focuses on fundamental issues related to information management on a computer, how data is organised on storage devices, transferred between storage and computer, structured within files, and effectively searched through indexing. Concepts from the first part of the course are exercised through the examination of XML as an example file format that is increasingly used to both store and transfer information on the web. Accompanying technologies for structuring, manipulating and querying XML will be studied and practiced.

Whereas the first part of the course examines the fundamental concepts for working with structured information and provides a basis for subsequent modules on database technology, the second part of the course focuses on information on the web, ranging from traditional information retrieval techniques through to emerging semantic techniques.

8 Year 3 – Junior Sophister Year

Third year is called Junior Sophister (JS) year in Trinity. In this section is a list of junior sophister subject modules and a brief description of each. In the Hilary semester, students take two of three optional modules. Full details are linked to the junior sophister web page at http://www.scss.tcd.ie/undergraduate/ba/current/junior-sophister.php.

<table>
<thead>
<tr>
<th>Michaelmas Term</th>
<th>Hilary Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS3011 Symbolic Programming</td>
<td>CS3061 Artificial Intelligence*</td>
</tr>
<tr>
<td>CS3012 Software Engineering</td>
<td>CS3013 Software Engineering Group Project</td>
</tr>
<tr>
<td>CS3014 Concurrent Systems I</td>
<td>CS3015 Concurrent Systems II</td>
</tr>
<tr>
<td>CS3016 Introduction to Functional Pro-</td>
<td>CS3017 Introduction to the Semantics of Formal Lang-</td>
</tr>
<tr>
<td>gramming</td>
<td>uages*</td>
</tr>
<tr>
<td>CS3041 Information Management II</td>
<td>CS3031 Advanced Telecommunications*</td>
</tr>
<tr>
<td>CS3071 Compiler Design I</td>
<td>CS3021 Computer Architecture III</td>
</tr>
<tr>
<td>ST1002 Statistical Analysis</td>
<td>CS3051 Foundations of Visual Computing</td>
</tr>
</tbody>
</table>

*denotes an elective module; two elective modules must be taken.

All modules have an ECTS weighting of 5 points.

8.1 CS3011 Symbolic Programming

This module introduces Prolog, a symbolic programming language. On completion, students will be able use standard Prolog techniques such as recursion, Definite Clause Grammars, cuts and negation.
8.2 CS3012 Software Engineering

This module provides students with a solid grounding in various aspects related to building large software systems.

The overall aim of this module is for students to learn the fundamental skills for building large, important software systems. This entails (i) recognising the general software lifecycle and its stages from domain analysis to maintenance, (ii) analysing software in the problem domain, (iii) identifying the fundamental approaches to managing software projects and teams, (iv) distinguishing the roles of stakeholders in a software project in general and in software teams in particular, (v) recognising architectures for building large-scale distributed software systems.

The module covers various aspects related to building software systems ranging from the use of software lifecycle models, to project and portability management, to large-scale software architectures. Specifically, software lifecycle models, including variations of the waterfall and spiral models as well as extreme programming, are introduced along with concepts that are relevant to the specific model stages. These concepts include UML-based O-O, and domain analysis, requirements and specification analysis, portability management, testing and debugging, and version control. Moreover, strategies for managing large software projects as well as project teams are presented and discussed.

8.3 CS3013 Software Engineering Group Project

This module follows on directly from CS3012 (Software Engineering) and focuses on the practical application of the various concepts and tools related to building software systems studied in third year software engineering. This practical application takes the form of a large “hands-on” group project that covers numerous aspects of building object-oriented software systems including problem analysis, usage of development environments, project management, team management, design, implementation, testing and documentation. Students will take a leadership role within these groups which are combined with students taking module CS2013.

8.4 CS3014 Concurrent Systems I

The goal of this module is to provide students with a deep understanding of parallel and multi-core architectures and to provide students with necessary architecture background for careers in professional software development and/or further research on these emerging platforms. The module uses recent multi-core processors for course work, which have been donated by Intel.
8.5 CS3015 Concurrent Systems II

The first part of this module 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 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.

8.6 CS3016 Introduction to Functional Programming

Functional programming languages present a powerful, abstract, and important direction in programming languages. The high level of abstraction and the expressive syntax makes program decomposition and composition unusually easy, while the close connections to the underlying semantics make formal reasoning tractable. Systems such as Google’s “Map/Reduce” framework demonstrate the influence of this approach, and the importance to a computer scientist of understanding it.

On this module students will learn to apply the techniques of functional programming in a practical context. The focus is on software design and programming in the functional style, and students will “learn by doing”, through regular weekly programming assignments and case studies.

The module draws on the programming and mathematics background the students have acquired in the first two years of the degree and extends it by teaching new approaches to program design and implementation.

8.7 CS3017 Introduction to the Semantics of Formal Languages

This module introduces students to formal methods for specifying the semantics of programming languages and formal techniques for verifying the behaviour of programs.
8.8 CS3021 Computer Architecture III

This module focuses on the architecture of modern high performance microprocessor systems. Topics covered are the architecture of RISC CPUs, instruction level pipelining, virtual memory, caches, multiprocessors, multiprocessor cache coherency and multiprocessor spin locks implementations.

The aim of the module is to explain (i) how high performance is obtained and (ii) the close relationship between the hardware and software inherent in the design of a modern microprocessor.

8.9 CS3031 Advanced Telecommunications

This module builds upon the junior and senior freshman Telecommunications modules and introduces them to advanced topics in the area of data communications and telecoms. In particular the area of high speed local area networks operating at speeds of 100 megabits and higher are examined. A more detail study is made of the TCP/IP protocols and strategies to introduce Quality of Service (QoS) and Mobility are examined. Topics in the area of telecommunications protocols such as ISDN and ATM are studied. The students are also introduced to the areas of Network Security.

8.10 CS3041 Information Management II

This module is focused on modelling of information and database system technology. More specifically it focuses on state-of-the-art database technology, from both the user and systems perspectives.

From a system engineering perspective, the module examines the concepts and algorithms for: Transaction processing, Concurrency control, Metadata Representation, Semantic Representation and Active Databases, Recovery, Database Security Policies, Integration of databases on the web, Semantic Web, Object Oriented, Object Relational Databases, Web based Databases.

From an information designers perspective, the module examines the theoretical model underpinning relational databases, functional dependency theory and normalisation (for information modelling), functional dependency modelling, Object Relational Modelling, implementation of databases and database applications using SQL PL (relational database query language) as well as PHP/SQL.

Thus the module is intended to enable the students to design information models and implement these models in state of the art databases (relational and native web databases), as well as be able to analyses and evaluate approaches to information organisation, storage, transaction support and management.
8.11 CS3051 Foundations of Visual Computing

This module introduces students to the mathematical concepts and methods appropriate to the synthesis, analysis, visualisation, and processing of imagery in computer programming.

8.12 CS3061 Artificial Intelligence I

This elective module introduces classic topics in AI, including search, constraint satisfaction, knowledge representation, abduction and action.

8.13 CS3071 Compiler Design I

The aim of the course is to teach the principles of compiler design, including finite-state automata and push-down machines, lexical analysers, top-down parsers, l-attributed translation grammars, and recursive-descent parsers. Students will learn how to use Coco/R to construct high-level language analysers and parsers.

8.14 ST1002 Statistical Analysis

The aim of the course is to introduce the students to basic statistical concepts. There will be considerable emphasis on the use of a statistical package to analyse data.

9 Year 4 – Senior Sophister Year

Fourth year is called Senior Sophister (SS) year in Trinity. In this section is a list of senior sophister subject modules and a brief description of each. Full details are linked to the senior sophister web page at http://www.scss.tcd.ie/undergraduate/ba/current/senior-sophister.php.

9.1 Year 4 – Moderatorship Only

Senior sophisters who have elected to follow the BA (Mod) programme with the intention of graduating with a BA (Mod) in Computer Science after successfully completing four years of study must take the following modules:

<table>
<thead>
<tr>
<th>Michaelmas Term</th>
<th>Hilary Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS4051 Human Factors (5 credits)</td>
<td>CS4098 Group Computer Science Project (10 credits)</td>
</tr>
<tr>
<td>CS4081 Technology Entrepreneurship (5 credits)</td>
<td>CS4099 Final Year Project (20 credits)</td>
</tr>
</tbody>
</table>
In addition, students must select four options from column A of the Options Table in section 11.

### 9.2 Year 4 – MCS Programme

Senior sophisters who have elected to follow the MCS programme with the intention of graduating with a Master in Computer Science after successfully completing five years of study must take the following modules:

<table>
<thead>
<tr>
<th>Michaelmas Term</th>
<th>Hilary Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS4051 Human Factors (5 credits)</td>
<td>CS7091 Industrial / Research Lab Internship (30 credits)</td>
</tr>
<tr>
<td>CS4081 Technology Entrepreneurship (5 credits)</td>
<td></td>
</tr>
</tbody>
</table>

In addition, student must select four options from columns A and B of the Options Table in section 11, subject to a maximum of 20 ECTS credits from Column A over Years 4 and 5.

### 9.3 CS4051 Human Factors

This module aims to provide students with an understanding of the main issues underlying the usability of systems, and the main techniques and processes for interface design and evaluation. They will also gain a basic understanding of the theories which account for human performance.

### 9.4 CS4081 Technology Entrepreneurship

This module introduces the fundamentals of technology entrepreneurship. It will cover the process technology entrepreneurs use to start companies. This involves taking a technology idea and finding a high-potential commercial opportunity, gathering resources such as talent and capital, figuring out how to sell and market the idea and managing rapid growth.

### 9.5 CS4098 Group Computer Science Project

The Group Computer Science Project is undertaken in Hilary Term by students who have elected to follow the BA (Mod) programme. The goal of this project is to promote teamwork and also to allow students to use their individual skills and experiences within the context of developing defined projects from specification to delivery. A key aim is to develop students abilities in framing design problems and working iteratively to achieve a working solution.
9.6 CS4099 Final Year Project

The Final Year Project is undertaken in Hilary Term by students who have elected to follow the BA (Mod) programme. The aim of the project is to integrate the theoretical and practical knowledge of the student across all of the years of their study and provide a practical demonstration of their capability in executing a challenging and large-scale project.

9.7 CS7091 Industrial / Research Lab Internship

The Internship is undertaken by students who have elected to follow the MCS programme. The aim is to enable students to further develop an understanding of how design aspects and theoretical aspects of computer science are applied to practical problems in a real world context.

10 Year 5 (MCS)

In this section is a list of Year 5 subject modules and a brief description of each. Students must take the following modules:

<table>
<thead>
<tr>
<th>Michaelmas Term</th>
<th>Hilary Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS7039 Research Methods (5 credits)</td>
<td>CS7092 MCS Dissertation (30 credits)</td>
</tr>
</tbody>
</table>

In addition, student must select five options from columns A and B of the Options Table in section 11, subject to a maximum of 20 ECTS credits from Column A over Years 4 and 5.

10.1 CS7039 Research Methods

To develop an awareness of research methodologies in general and those applicable to Masters and PhD research in CS & Statistics in particular. To develop written and oral communication skills.

10.2 CS7092 MCS Dissertation

Students will select and carry out an in-depth research project which is expected to yield publishable results. Students must select the project, carry out required investigations and submit their dissertation within the academic year.
11 Senior Sophister (Year 4) and Year 5 Options

In this section is a list of optional modules that may be taken by students in Years 4 and 5, as set out in sections 9 and 10. A brief description of each module is also provided.

<table>
<thead>
<tr>
<th>Column A — BA (Mod.) and MCS</th>
<th>Column B — MCS Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS4001 Fuzzy Logic</td>
<td>CS7003 Middleware for Distributed Systems</td>
</tr>
<tr>
<td>CS4003 Formal Methods</td>
<td>CS7004 Embedded Systems</td>
</tr>
<tr>
<td>CS4012 Topics in Functional Programming</td>
<td>CS7008 Vision Systems</td>
</tr>
<tr>
<td>CS4031 Mobile Communications</td>
<td>CS7009 Networked Applications</td>
</tr>
<tr>
<td>CS4032 Distributed Systems</td>
<td>CS7012 Management of Networks and Distributed Systems</td>
</tr>
<tr>
<td>CS4052 Computer Graphics</td>
<td>CS7030 Numerical Methods and Advanced Mathematical Modelling I</td>
</tr>
<tr>
<td>CS4053 Computer Vision</td>
<td>CS7058 Numerical Methods and Advanced Mathematical Modelling II</td>
</tr>
<tr>
<td>CS4061 Artificial Intelligence IIA</td>
<td>CS7031 Graphics and Console Hardware</td>
</tr>
<tr>
<td>CS4062 Artificial Intelligence IIB</td>
<td>CS7033 Real-time Animation</td>
</tr>
<tr>
<td>CS4071 Compiler Design II</td>
<td>CS7048 Data Communications and Wireless Networking</td>
</tr>
<tr>
<td>CS4052 Computer Graphics</td>
<td>CS7052 Sustainable Computing</td>
</tr>
</tbody>
</table>

Notes:

- It may not be possible to offer all the options each year and some modules have pre-requisites.
- Additional modules may be added to this list.
- All modules have an ECTS weighting of 5 points.
- Students who have elected to follow the BA (Mod) programme must choose four options from Column A.
- Students who have elected to follow the MCS programme must choose four options in Year 4 and four options in Year 5, subject to a total of no more than 20 credits from Column A in Years 4 and 5.
11.1 CS4001 Fuzzy Logic

This module will introduce you to the exciting new field of fuzzy systems. Fuzzy systems are in almost daily use: your washing machine has fuzzy controls and many of the refrigerators do too. In both cases, the machines can sense the amount of stuff (clothes or food) and adjust their operations accordingly.

Fuzzy logic has been developed by computer scientists and control engineers over the last 30 odd years. Fuzzy logic is now being used in a range of critical systems ranging from image processing to the processing of financial time series.

11.2 CS4003 Formal Methods

This is a once-off course that follows on material presented in CS3001, which introduce the Unifying Theories of Programming framework (UTP) and showed how it could be used to reason about sequential imperative programs. This follow-on course explores using the UTP to reason about programs that interact with their environment, and to explore how to model concurrent and parallel execution. It also looks in more detail at the unification aspects of UTP. In addition it looks at other formal theories outside of UTP, and their associated tool support.

11.3 CS4012 Topics in Functional Programming

Functional programming languages present a powerful, abstract, and important direction in programming languages. The high level of abstraction and the expressive syntax makes program decomposition and composition unusually easy, while the close connections to the underlying semantics make formal reasoning tractable. Systems such as Google’s “Map/Reduce” framework demonstrate the influence of this approach, and the importance to a computer scientist of understanding it.

On this course students will learn to apply the techniques of functional programming in a practical context. The focus is on software design and programming in the functional style, and students will learn by doing, through regular weekly programming assignments and case studies.

The course draws on the programming and mathematics background the students have acquired in the first two years of the degree and extends it by teaching new approaches to program design and implementation.

11.4 CS4031 Mobile Communications

Effective wireless communication is the key enabling technology for realising the emerging ubiquitous computing vision. Mobile Communications is a final year option module which imbues the next generation of graduates with a cognisance and
awareness of both the capabilities and limitations of modern mobile devices. In this module students being by learning about the underlying principles of wireless transmission and how these underpin the design of wireless communication networks. This material forms a platform for the subsequent analysis, assessment and implementation of a wide variety of modern wireless communication systems.

Critical appraisal of recent publications in IEEE and ACM journals is used to enhance each students ability to communicate effectively through the written medium.

This module aims to:

1. provide students with a sound technical basis in current and emerging mobile communications technologies.

2. require students to derive and implement solutions to problems in the mobile communications domain.

3. encourage students to develop and refine their technical writing and critical appraisal skills in a supportive environment.

4. equip students with the capabilities to realize innovative solution platforms from minimal problem domain specifications.

11.5 CS4032 Distributed Systems

Building distributed applications is a difficult task due to the concurrency, communication latency, and possibility of partial failure that is inherent in distributed systems. As in other areas of computer science, the trend in providing support for building distributed applications has been towards presenting the application developer with ever higher levels of abstraction and, in the particular case of distributed programming, of location transparency. This course takes a critical look at some of the paradigms and architectural issues involved in distributed programming and their likely evolution.

Students will be given opportunities to develop their problem solving, programming and written communication skills by designing solutions to programming problems, implementing those solutions as fully networked distributed systems.

11.6 CS4052 Computer Graphics

The objective of this module is to equip the students with the fundamental understanding of the major elements of Computer Graphics and explore related areas including geometric modelling, rendering and animation. The main focus of the module is on the mathematics and algorithms used in the synthesis of computer
graphics imagery and animation, and their practical application. Students are introduced to the standard architectures of modern graphical applications including details on the underlying hardware and low-level software components common to all such systems. The module is intended to enable students to bridge the gap between these low-level fundamental, components common to all computer applications, and the high-level abstract output in most interactive graphical applications.

Students are also introduced to OpenGL, a modern high-level graphics API which is widely used for 3D Design and Visualisation, along with the industry standard modelling software, 3D Studio Max, and this software is used throughout the course to demonstrate concepts and to allow the students to develop their own 3D models, scenes and applications.

11.7 CS4053 Computer Vision

The aim of this module is to give students a firm understanding of the theory underlying the processing and interpretation of visual information and the ability to apply that understanding to ubiquitous computing and entertainment related problems. It provides them with an opportunity to apply their problem-solving skills to an area which, while it is firmly part of computer science/engineering, draws strongly from other disciplines (physics, optics, psychology). The course is based around problems so that the technology is always presented in context and during some tutorials students work in groups to design solutions to real world problems using the techniques that they have been taught. In addition, the course has a significant practical component so that students can appreciate how difficult it can be to apply the technology.

11.8 CS4061 Artificial Intelligence IIa

This is an in-depth initiation into some topics in AI, including the use of simple description logics and the application of finite-state methods to natural language processing. The syllabus includes topics such as knowledge representation, description logics, finite-state methods and reasoning about change.

11.9 CS4062 Artificial Intelligence IIb

This module provides students with an introduction to machine learning, with special attention to its applications to NLP. Some of the topics covered include text categorisation, dimensionality reduction, probabilistic classification (the 3 flavours of Naive Bayes classifiers), symbolic methods, instance-based methods, other supervised methods, unsupervised learning.
11.10 CS4071 Compiler Design II
To teach the principles of compiler optimization.

11.11 CS7003 Middleware for Distributed Systems
To expose students to the complexities involved in designing and building distributed applications and to develop students analytical skills. To gain in-depth understanding of the principle paradigms used in the area. To gain an appreciation of the open research issues in the area. The course covers the underlying theory of distributed computing. A significant feature of the course is the use of an interactive teaching style in which students are encouraged to discover the fundamental principles of distributed computing by considering challenge problems collectively or in small groups before being presented with the relevant course material.

11.12 CS7004 Embedded Systems
The module will give students the opportunity to gain the knowledge and skills necessary to develop embedded systems. Students taking the module will study a real embedded hardware platform in depth (based on the widely used ARM7TDMI microcontroller) and will use this platform in conjunction with industry-standard software tools to develop embedded systems of varying complexity. Topics covered will be in the broad areas of computer architecture, systems software and I/O. Throughout the module, students will be given opportunities to consider issues of particular relevance in embedded systems design (e.g. development cost, power, performance and reliability).

11.13 CS7008 Vision Systems
The aim of this module is to give students a firm understanding of the theory underlying the processing and interpretation of visual information and the ability to apply that understanding to ubiquitous computing and entertainment related problems. It provides them with an opportunity to apply their problem-solving skills to an area which, while it is firmly part of computer science/engineering, draws strongly from other disciplines (physics, optics, psychology). The course is based around problems so that the technology is always presented in context and during some tutorials students work in groups to design solutions to real world problems using the techniques that they have been taught. In addition, the course has a significant practical component so that students can appreciate how difficult it can be to apply the technology.
11.14 CS7009 Networked Applications

This module aims to provide an understanding of the world-wide web as an application platform that is becoming increasingly important economically and socially. It covers the fundamental content, social and metadata structures that make up the web and how they can be represented, analysed and manipulated. It addresses the practical tools and techniques of web application programming, including client and server side programming languages, XML and semantic web information representation and analysis of application usage. It will encourage critical analysis of the impact of web applications on business and social concerns.

11.15 CS7012 Management of Networks and Distributed Systems

The aim of this module is to identify the issues and design approaches involved in managing networks & Services. To be capable of designing management solutions for various management application areas and organisations. Recognise and analyse the current management standards and technology trends in management of networks and distributed systems.

11.16 CS7030/CS7058 Numerical Methods and Advanced Mathematical Modelling I/II

The aim of this module is to encourage and foster the development of independent critical thinking. In particular students should be able to: (a) model problems using mathematics and statistics, (b) formulate and propose solutions, (c) infer from observation and interpret results.

The students should be able to use, critique, and edit (where relevant) Internet resources such as Wikipedia, Wolfram Research, etc.

11.17 CS7031 Graphics and Console Hardware

This module will give students a thorough overview of modern graphics hardware and multi-core systems. Each of the current generation of consoles will be analysed and compared in detail. The course will cover general purpose computer architecture e.g. memory hierarchies, SIMD & VLIW architectures, Vector units, multi-core, hyperthreading architectures and I/O busses. Students will become familiar with GPU pipeline architectures e.g. geometry, rasterisation, texture, fragment pixel and vertex shaders and newer Physics Processing Unit (PPU) and multi-GPU technology. Students will become familiar with the challenges of developing for these architectures through optimising compilers, compiler intrinsics and graphics card drivers.
11.18 **CS7033 Real-time Animation**

The aim of this course is to provide students with a deep understanding of the theory and techniques behind real time animation. We will explore computer animation and advanced issues such as behavioural animation and motion capture and also look at specific fundamental concepts such as interpolation.

11.19 **CS7048 Data Communications and Wireless Networking**

Students will acquire and demonstrate competence and capability in the areas of: Teamwork, Time management, Research Methods, Project Planning, Literature Review, Project Specification, Project and Experiment Design, Project Execution, Project Outcomes. Clear, concise, appropriate and articulate dissemination of project outputs in the form of an IEEE/ACM calibre paper. Students will be assessed and graded on all these elements.

11.20 **CS7052 Sustainable Computing**

This course introduces the foundations of sustainability and gives an appreciation for how energy is currently used in ICT and the problems created by the continuous growth of the ICT industry. The course will then delve into some detail on where power is consumed in current networks and how new techniques and trends will affect this. It will examine initiatives that have been taken to date, the impact that they have had and the prospects for future initiatives that will shape the industry.

12 **Prizes**

Gold medals are awarded by the Board to candidates of the first class who have shown exceptional merit at the annual degree examination in honor or professional courses (see [http://www.tcd.ie/vpcao/administration/examinations/criteria-for-gold-medal.php](http://www.tcd.ie/vpcao/administration/examinations/criteria-for-gold-medal.php)).

Various studentships, scholarships, exhibitions, and other prizes are awarded to students on the results of honor and other examinations, provided that sufficient merit is shown. Monetary awards are sent direct to prize-winners unless otherwise stated under the regulations for the particular prize. For details please refer to the University Calendar.

At the annual examinations, a book prize (under review) is awarded to each candidate obtaining an overall first class honors grade in an honor or professional course. These prizes are not awarded in the senior sophister year. These prizes, which
are issued in the form of vouchers, can be exchanged by the student at designated booksellers. Book prizes are issued by the Examinations Office and are posted to recipient students at their home address.

12.1 Prizes in the Computer Science Programme

The following prizes are listed in the University Calendar for the Computer Science programme.

12.1.1 The Victor W. Graham Prize

This prize, founded in 1986 from funds subscribed by friends and pupils to mark Mr V. W. Graham's retirement, is awarded to the junior freshman in the moderatorship in computer science course who obtains the highest mark in the summer examination in pure mathematics. Value, €750.

12.1.2 The Ludgate Prize

This prize was instituted in 1991 in memory of Percy E. Ludgate, an Irish designer of an analytical engine. It is awarded to the student who submits the best project in the senior sophister year of the moderatorship in computer science. Value, €127.

12.1.3 The William Nurock Prize

This prize was founded in 1938 by a bequest from William Nurock. The conditions for the award of the prize were changed in 1984. It is now awarded annually to the best student in the final year examinations of the moderatorship in computer science, providing that such student also attains gold medal standard. Value, €1,000.

12.2 Scholarship

Foundation scholarship—("Schol")—is a College institution with a long history and high prestige. The examination for the award of scholarship is set and assessed so as to select students of outstanding ability. The objective of the foundation scholarship examination is to identify students who, at a level of evaluation appropriate to the senior freshman year, can consistently demonstrate exceptional knowledge and understanding of their subjects.

The examination requires candidates to demonstrate skill in synthesising and integrating knowledge across the full range of the set examination materials; to demonstrate rigorous and informed critical thought; and, in appropriate disciplines, to demonstrate a highly-developed ability to solve problems and apply knowledge.
Attempting the scholarship examination is highly recommended. For more information, please visit the University Calendar entry at http://www.tcd.ie/calendar/assets/pdf/foundation-scholarships.pdf.

13 Regulations

13.1 College Regulations

College regulations are set out in the University Calendar, which may be consulted in any College Library, the Enquiries Office, any academic or administrative office or online at http://www.tcd.ie/calendar/. The two most relevant extracts of the Calendar, entitled General Regulations and Information and Faculty of Engineering, Mathematics and Science, are handed out at registration at the beginning of the year. You are expected to be aware of the various regulations. Ignorance of the regulations is not a valid reason for failure to comply.

13.2 Attendance and Participation

Students are required to attend all lectures, laboratory and tutorial sessions associated with their course programme of study and to participate fully in the academic work of their class.

Students must notify the lecturer concerned or their tutor as early as possible if they are unable to attend lectures, laboratories or tutorials or to submit coursework for any reason. Students who are absent for medical reasons should notify their tutor and will usually be required to provide a medical certificate.

13.3 Non-Satisfactory Attendance or Performance

At the end of each teaching term, students whose attendance or performance in coursework has not been satisfactory may be reported to the Senior Lecturer's Office as non-satisfactory for that term (see University Calendar, General Regulations and Information, section II, §§23-24). Normally, where students are non-satisfactory in a course for two terms in the year they may be refused permission to take their annual examinations and may be required to repeat the year.

Unless otherwise specified for an individual module, a student's attendance and participation will be deemed to be non-satisfactory if they fail to attend more than one third of the scheduled lectures, laboratories or tutorials or make a serious attempt to complete more than one third of the coursework for any individual module.
13.4 Retaining Coursework

It is the responsibility of each student to retain a copy of any coursework that they submit.

13.5 Plagiarism

Students should be aware of the University’s policy regarding plagiarism. Plagiarism is interpreted by the University as the act of presenting the work of others as one’s own work, without acknowledgement. Plagiarism is considered as academically fraudulent, and an offence against University discipline. The University considers plagiarism to be a major offence, and subject to the disciplinary procedures of the University. Plagiarism can arise from deliberate actions and also through careless thinking and/or methodology. The offence lies not in the attitude or intention of the perpetrator, but in the action and in its consequences. Plagiarism can arise from actions such as:

(a) copying another student’s work;
(b) enlisting another person or persons to complete an assignment on the student’s behalf;
(c) quoting directly, without acknowledgement, from books, articles or other sources, either in printed, recorded or electronic format;
(d) paraphrasing, without acknowledgement, the writings of other authors.

Examples (c) and (d) in particular can arise through careless thinking and/or methodology where students:

(i) fail to distinguish between their own ideas and those of others;
(ii) fail to take proper notes during preliminary research and therefore lose track of the sources from which the notes were drawn;
(iii) fail to distinguish between information which needs no acknowledgement because it is firmly in the public domain, and information which might be widely known, but which nevertheless requires some sort of acknowledgement;
(iv) come across a distinctive methodology or idea and fail to record its source.

All the above serve only as examples and are not exhaustive. Students should submit work done in co-operation with other students only when it is done with the full knowledge and permission of the lecturer concerned. Without this, work submitted
which is the product of collusion with other students may be considered to be plagiarism.

It is clearly understood that all members of the academic community use and build on the work of others. It is commonly accepted also, however, that we build on the work of others in an open and explicit manner, and with due acknowledgement. Many cases of plagiarism that arise could be avoided by following some simple guidelines:

(i) Any material used in a piece of work, of any form, that is not the original thought of the author should be fully referenced in the work and attributed to its source. The material should either be quoted directly or paraphrased. Either way, an explicit citation of the work referred to should be provided, in the text, in a footnote, or both. Not to do so is to commit plagiarism.

(ii) When taking notes from any source it is very important to record the precise words or ideas that are being used and their precise sources.

(iii) While the Internet often offers a wider range of possibilities for researching particular themes, it also requires particular attention to be paid to the distinction between one’s own work and the work of others.

Particular care should be taken to keep track of the source of the electronic information obtained from the Internet or other electronic sources and ensure that it is explicitly and correctly acknowledged. It is the responsibility of the author of any work to ensure that he/she does not commit plagiarism. Students should ensure the integrity of their work by seeking advice from their lecturers, tutor or supervisor on avoiding plagiarism.

If plagiarism as referred to above is suspected, procedures defined in the University Calendar, General Regulations and Information, Part 2, §§76–84 will be followed.

13.6 Identifying Plagiarism

The School reserves the right to use plagiarism detection services, such as “Turnitin”, to identify potential cases of plagiarism.

13.7 Examinations

Students are examined in the work of each year. Supplemental examinations will be held in Michaelmas term each year, except for the fifth year. Permission to take supplemental examinations will not normally be granted to students whom the court of examiners considers not to have made a serious attempt at the annual examinations unless an adequate explanation is furnished. Students must submit satisfactory course work in each year. Students who fail to do so, or whose
attendance is unsatisfactory, may be refused permission to take all or part of the
annual examinations for the year.

Students who have not passed in its entirety any examination within eighteen
months from the date on which they first became eligible for it, will be reported to
the University Council as unsatisfactory with a recommendation for their exclusion
from the course.

In years one to four, successful candidates at the annual examinations will be
awarded one of the following grades: first class honors, second class honors (with two
divisions, first and second) or third class honors.

The BA (Mod) degree result will be awarded based on a combined mark from the
annual sitting of the junior sophister examinations (which count for 20% of the
moderatorship result) and senior sophister examinations (which count for 80% of the
moderatorship result). Successful candidates at the year five examinations will be
awarded a classified BA (Mod.) and an MCS or an MCS with Distinction.

13.8 Progression

Students are examined in the work of each year. To progress to the next year of the
programme, students must be successful at the annual examinations. 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 in the
examination results. The method of assessment of modules varies between annual
and supplemental examinations—see section 13.9 below.

13.8.1 Annual Examinations

In order to be successful in the annual examinations, students must pass all modules,
as defined in section 13.9.1 below, or satisfy all the following requirements:

1. pass modules with a combined ETCS rating of at least 50 points,

2. achieve a mark of 30% or more in each failed module,

3. achieve an overall mark of 45% or more in the annual examinations

A student’s overall mark will be calculated as the average of each module’s mark
weighted by its ECTS rating. Students who pass the annual examinations will be
awarded an overall grade according to the scale below.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>70%–100%</td>
</tr>
<tr>
<td>II.1</td>
<td>60%–69%</td>
</tr>
<tr>
<td>II.2</td>
<td>50%–59%</td>
</tr>
<tr>
<td>III</td>
<td>40%–49%</td>
</tr>
</tbody>
</table>
13.8.2 Supplemental Examinations

In order to be successful in the supplemental examinations, students must pass, as defined in section 13.9.2 below, all the modules they are required to sit, or satisfy the following requirements:

1. pass modules with a combined ETCS rating of at least 50 points at either the annual or supplemental sittings,

2. achieve a mark of 35% or more in each failed module at the supplemental sitting,

3. achieve an overall mark of 45% or more.

A student’s overall mark will be calculated as the average of each module’s mark weighted by its ECTS rating. Where a module has been examined more than once, the mark achieved in the most recent examination will be used.

13.9 Module Assessment

The form of assessment may vary between modules and may include coursework, written examination or a combination of both. Unless otherwise specified, the regulations in this section will apply to the assessment of modules.

13.9.1 Annual Examinations

Students must satisfy the following criteria to pass a module at the annual examination:

Modules assessed by written examination only  To pass the module students must achieve a mark of 40% or more in the examination.

Modules assessed by coursework only  To pass the module students must achieve a mark of 40% or more in the coursework.

Modules assessed by a combination of coursework and written examination  To pass the module, students in the freshman years and in junior sophister year must achieve a mark of 40% or more in the written examination and separately a mark of 40% or more in the coursework component. In senior sophister year, students must achieve a combined mark of 40% or more in the examination and the coursework to pass the module.

Additional criteria set out in module descriptors  To pass the module, students must also satisfy any additional criteria (e.g. attendance at group meetings, submission of coursework) set out in the module descriptor.
13.9.2 Supplemental Examinations

Modules that were assessed by written examination only, by coursework only or a combination of both written examination and coursework in the annual examination will usually be assessed in the supplemental examination by written examination only. Modules that were assessed by coursework only in the annual examinations may again be assessed by coursework only in the supplemental examinations.

Where a module was wholly or partly assessed by coursework in the annual examinations, the mark awarded for that coursework will not be included in any mark awarded for the module in the supplemental examinations.

Students must satisfy the following criteria to pass a module at a supplemental examination:

Modules assessed by supplemental written examination only To pass the module candidates must achieve a mark of 40% or more in the supplemental examination.

Modules assessed by supplemental coursework only To pass the module candidates must achieve a mark of 40% or more in the supplemental coursework.

Where a junior sophister sits supplemental examinations, the contribution of each module’s result towards the calculation of the moderatorship degree result will be the mark achieved for the module at the supplemental examination, limited to a maximum of 40%.

13.10 Progression to Year 4 of the Masters Programme

In order to progress to year 4 of the Masters programme, students must achieve an overall mark of 50% or better in their junior sophister examinations. Students may take supplemental examinations in those modules in which they obtained less than 50% in the annual examinations, or may repeat their junior sophister year in its entirety to achieve the required standard.

13.11 Progression to Year 5 of the Masters Programme

In order to progress to year 5 of the Masters programme, students must be on year 4 of the Masters programme and must achieve an overall mark of 60% or better in their year 4 examinations. Students may take supplemental examinations in those modules in which they obtained less than 60% in the annual examinations, or may repeat year 4 in its entirety to achieve the required standard. Supplemental examinations in year 4 may be taken solely in order to attain the progression requirement for entry into the fifth year.
13.12 **Ordinary BA Degree (exit only)**

Students who have passed their junior sophister examinations may have an ordinary BA degree conferred if they do not choose, or are not allowed, to proceed to the senior sophister year of the programme or if they fail to complete satisfactorily the senior sophister year of the course. Except by permission of the University Council, on the recommendation of the Executive Committee of the School of Computer Science and Statistics, an ordinary BA degree may be conferred only on candidates who have spent at least two years in the University.

13.13 **Moderatorship Degree**

The BA (Moderatorship) degree result is awarded based on a combined mark from the annual sitting of the junior sophister examinations (which count for 20% of the moderatorship result) and senior sophister examinations (which count for 80% of the moderatorship result).

An honors degree is awarded to candidates who achieve a weighted average mark of 40% or more and, if one or more senior sophister modules has been failed, those modules account for not more than 10 ECTS credits. Where students are awarded an honors degree, the class of degree awarded is based on the weighted average mark achieved as follows: First Class Honors: 70%–100%, Second Class Honors, First Division: 60%–69%, Second Class Honours, Second Division: 50%–59%, Third Class Honors: 40%–49%. If the weighted average mark is below 40% or if modules accounting for more than 10 ECTS credits are failed, then students may be awarded a pass degree if they are clear of the junior sophister year.

Students who have been successful in their senior sophister examinations may have the BA (Mod) degree conferred if they do not choose, or are not allowed, to proceed to the fifth year of the programme. Students exiting the programme after the senior sophister year obtain a moderatorship based on their results at their first attempt at the senior sophister examinations.

13.14 **Master in Computer Science Degree Result**

Successful candidates at the Year 5 examinations will be awarded a classified BA (Moderatorship) based on their results in Years 3 and 4, as set out above, and a Master in Computer Science or a Master in Computer Science with Distinction.
## Academic Year Structure

<table>
<thead>
<tr>
<th>Cal. Wk</th>
<th>Dates 2011/12 (week beginning)</th>
<th>Outline Structure of Academic Year 2011/12</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29-Aug-11</td>
<td>Supplemental Examinations</td>
<td>Statutory Term (Michaelmas) begins</td>
</tr>
<tr>
<td>2</td>
<td>05-Sep-11</td>
<td>PG Registration</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>19-Sep-11</td>
<td>UG New Entrant Registration/Freshers' Week</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>26-Sep-11</td>
<td>Teaching Week 1 Registration continuing students</td>
<td>Michaelmas Lecture term begins</td>
</tr>
<tr>
<td>6</td>
<td>03-Oct-11</td>
<td>Teaching Week 2 Registration continuing students</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10-Oct-11</td>
<td>Teaching Week 3 Registration continuing students</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>17-Oct-11</td>
<td>Teaching Week 4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>24-Oct-11</td>
<td>Teaching Week 5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>31-Oct-11</td>
<td>Teaching Week 6 (Monday, Public Holiday)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>07-Nov-11</td>
<td>Teaching Week 7 - Study Week</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>14-Nov-11</td>
<td>Teaching Week 8</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>21-Nov-11</td>
<td>Teaching Week 9</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>28-Nov-11</td>
<td>Teaching Week 10</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>05-Dec-11</td>
<td>Teaching Week 11</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>12-Dec-11</td>
<td>Teaching Week 12</td>
<td>C-Michaelmas term ends Friday 16 December 2011</td>
</tr>
<tr>
<td>17</td>
<td>19-Dec-11</td>
<td>Christmas Period (College closed from 23 December 2011 to 3 January 2012)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>26-Dec-11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>02-Jan-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>09-Jan-12</td>
<td>Foundation Scholarship Examinations</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>16-Jan-12</td>
<td>Teaching Week 1</td>
<td>Hilary Term begins</td>
</tr>
<tr>
<td>22</td>
<td>23-Jan-12</td>
<td>Teaching Week 2</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>30-Jan-12</td>
<td>Teaching Week 3</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>06-Feb-12</td>
<td>Teaching Week 4</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>13-Feb-12</td>
<td>Teaching Week 5</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>20-Feb-12</td>
<td>Teaching Week 6</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>27-Feb-12</td>
<td>Teaching Week 7 - Study Week</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>05-Mar-12</td>
<td>Teaching Week 8</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>12-Mar-12</td>
<td>Teaching Week 9 (Saturday, St Patrick’s Day)</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>19-Mar-12</td>
<td>Teaching Week 10 (Monday, Public Holiday)</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>26-Mar-12</td>
<td>Teaching Week 11</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>02-Apr-12</td>
<td>Teaching Week 12 (Friday, Good Friday)</td>
<td>C-Hilary Term ends Friday 6 April 2012, Good Friday</td>
</tr>
<tr>
<td>33</td>
<td>09-Apr-12</td>
<td>Revision (Monday, Easter Monday)</td>
<td>C-Revision begins, Easter Monday 9 April 2012</td>
</tr>
<tr>
<td>34</td>
<td>16-Apr-12</td>
<td>Revision Trinity Week (Monday, Trinity Monday)</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>23-Apr-12</td>
<td>Revision</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>30-Apr-12</td>
<td>Annual Examinations 1</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>07-May-12</td>
<td>Annual Examinations 2 (Monday, Public Holiday)</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>14-May-12</td>
<td>Annual Examinations 3</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>21-May-12</td>
<td>Annual Examinations 4</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>28-May-12</td>
<td>Marking/Courts of Examiners/Results</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>04-Jun-12</td>
<td>Marking/Courts of Examiners/Results (Monday, Public Holiday)</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>11-Jun-12</td>
<td>Marking/Courts of Examiners/Results</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>18-Jun-12</td>
<td>Marking/Courts of Examiners/Results/Courts of First Appeal</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>25-Jun-12</td>
<td>Courts of First Appeal/Academic Appeals</td>
<td>C-Statutory (Trinity) Term ends Friday 29 June 2012</td>
</tr>
<tr>
<td>45 to 52</td>
<td>02-Jul-12 to 20-Aug-12</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. C-Ends Friday 24 August 2012</td>
</tr>
</tbody>
</table>

4 July 2011
<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.00 – 13.00</td>
<td>MT: CS1011: Lect M17 H: CS1012: Lect LB04</td>
<td>HT: CS1021: Lab (B) LG36 H: CS1022: Lab (B) LG36</td>
<td>MT: CS1001: Lect M20 H: CS1002: Lect LB08</td>
<td>MT: CS1011: Lab (B) ICT Lab1</td>
<td></td>
</tr>
</tbody>
</table>

**Locations:**
- LB01: Lloyd Institute, Lecture Theatre 01, 1444
- ICTLab1: ICT Lab 1, Upper Floor
- AM: Arts Building, Lecture Theatre 008
- M17/M20: Museum Building Rooms 17/20
- GU: O’Reilly Institute
- MacNeill Theatre: Hamilton Building
- DO: Drawing Office, Museum Building

**Time Dates:**
- MT: Wk 1 – Wk 4
- HT: 17.00 – 18.00
- MT: Wk 5 – Wk 10
- HT: 16.00 – 17.00
- MT: Wk 11 – Wk 12
- HT: 15.00 – 16.00

**NOTE:** FOR TUTORIAL GROUPINGS PLEASE CONSULT THE BA COMPUTER SCIENCE NOTICE BOARD IN THE O’REILLY INSTITUTE.
### Senior Freshman Timetable

**School of Computer Science and Statistics**
**BA Computer Science: Senior Freshman Timetable 2011-12**

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>09.00 – 10.00</td>
<td>HT: CS2022: Lect M20</td>
<td>MT: CS2031: Lab /ICT Lab2</td>
<td>HT: CS2041: Lect L01 / ICT Lab 1</td>
<td>HT: CS2015: Lab LG12</td>
<td></td>
</tr>
<tr>
<td>10.00 – 11.00</td>
<td>MT: CS2031: Lect L01</td>
<td>MT: CS2031: Lect L01 / ICT Lab 1</td>
<td>MT: CS2014: Lab LG12</td>
<td>MT: CS2015: Lab LG12</td>
<td></td>
</tr>
</tbody>
</table>

**Code: Module ECTS: Lecturer**
- CS2011 Programmin Techniques I 5 ECTS: Dr A O’Connor
- CS2012 Programming Techniques II 5 ECTS: Dr D. Gogg
- CS2013 Programming/Project II 5 ECTS: Dr G. Strong / Dr T. Seagoe
- CS2014 System Programming II 5 ECTS: Dr D. Gogg
- CS2015 Senior Programming II 5 ECTS: Dr D. Gogg
- CS2021 Microprocessor Systems 5 ECTS: Dr M. O’Brien
- CS2022 Computer Architecture 5 ECTS: Dr M. O’Brien
- CS2013 Telecommunications II 5 ECTS: Dr S. W. Warah
- CS2014 Information Management I 5 ECTS: Dr D. H. Sullivan
- CS2015: Discrete Mathematics I 5 ECTS: Mr D. M. Williamson
- CS2021: Discrete Mathematics II 5 ECTS: Mr D. M. Williamson
- CS2023: Discrete Mathematics III 5 ECTS: Dr D. M. Williamson

**Locations:**
- LB01/02: Lloyd Institute: First Floor, Room 107-1.20
- M21: M210, Science Building, Room M21
- LG12/24: O’Reilly Institute: Seminar Theatre
- ICT Lab 1/2: ICT Hut, Lab
- JOLY: Hamilton Building
- Salmon: Hamilton Building
- Synge: Hamilton Building
- Maxwell 5: Hamilton Building

**Term Dates:**
- MT 2011/12 - 6/12/12 (Reading: Wk 11 Nov)
- MT 2011/12 - 6/12/12 (Reading: Wk 11 Nov)

**NOTE:** Please consult noticeboards in O’Reilly Institute for tutorial groupings.

**Broad Curriculum:** Please consult the Broad Curriculum Timetable.
<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
</table>

See [http://www.scss.tcd.ie/undergraduate/timetables/](http://www.scss.tcd.ie/undergraduate/timetables/).
<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>09.00 – 10.00</td>
<td>MT: CS4001: Lect LB01</td>
<td>MT: CS4001: Lect LB01</td>
<td>MT: CS4051: Lect Synge</td>
<td>MT: CS4051: Lect LB12</td>
<td></td>
</tr>
<tr>
<td>13.00 – 14.00</td>
<td>MT: CS4053: Lect LB04/ICTLab2</td>
<td>MT: CS4053: Lect LB01/ICTLab2</td>
<td>MT: CS4032: Lect LB04/ICTLab2</td>
<td>MT: CS4032: Lect LB04</td>
<td></td>
</tr>
</tbody>
</table>

Code: Module ECTS: Lecture:
- CS4001: Fuzzy Logic: 5 ECTS: Prof. K Ahmad
- CS4003: Formal Methods: 5 ECTS: Dr Andrew Buttimer
- CS4012: Topics in Functional Programming: 5 ECTS: Mr G Strong & Dr A Butterfield
- CS4031: Mobile Communications: 5 ECTS: Mr M Huggard & Dr M Ruffini
- CS4032: Distributed Systems: 5 ECTS: Dr P. Brosnan & Dr G O'Sullivan
- CS4043: Human Factors: 5 ECTS: Dr G O'Sullivan
- CS4052: Computer Graphics: 5 ECTS: Dr C. Sheilavan
- CS4053: Computer Vision: 5 ECTS: Dr G Lacey & Dr K. Dawson-Howe
- CS4081: Entrepreneurship: 5 ECTS: Prof. O. O'Mahony
- CS4061: Artificial Intelligence II: 5 ECTS: Dr T Fernando

Locations:
- LB01/02: Lloyd Institute, Lower Basement, Lecture Theatre
- LB20/21: Lloyd Institute, First Floor, Room 2.20/21
- SALMON: Hamilton Lecture Theatre 01
- M7/01: Museum Building, Room M7/01
- ICT Lab 1: ICT Huts
- JUC: Hamilton Building Lecture Theatre 4
- MACNEILL: Hamilton Building Lecture Theatre 3
- SYNGE: Hamilton Building Lecture Theatre 2
- 8 Warden Square: Room 1.5

Term Dates:
- MT: 27/09-10/12/10 (Reading Wk 8-12 Nov)
- HTE: 17/10-11/12/10 (Reading Wk 28/4 March)
Campus Map
First Year in University

Everybody says college is different from school. Of course, in lots of obvious ways it is different, and no doubt you’ll enjoy finding out just what those differences are. In not-so-obvious ways though, college is very different from school, and in this section we concentrate on how the academic side of university life is different and what you need to do about it.

1. You are not at school. We want you to do more than simply reproduce what you are told in a lecture. You need to get a good command of the material. In computing-related disciplines, the best way to do this—and the best way to know that you have really learned something—is to apply your new knowledge to solving new problems; not just the examples done in class, but to similar problems you’ll find in textbooks or elsewhere (later on, as a professional computer scientist, you will have to apply your knowledge to problems you have never seen before—now is the time to start).

2. Expect the material to be covered much faster than at school. Lecture time is at a premium, so it must be used efficiently. You cannot be taught everything in lectures and tutorials. It is your responsibility to learn the material. Most of this learning will take place outside the classroom, and you must be willing to put in the study time necessary to ensure that this learning takes place. If you do fall behind in a course—that is, if you can’t continue to understand the lectures as they are given—then you really need to make the effort to catch up right away. Don’t be tempted to think that you can somehow catch up at the end of the year—it’s almost impossible.

3. A lecturer’s job is primarily to provide you with a framework, with some of the particulars, to guide you in doing your learning of the concepts and methods that comprise the material of the course. It is not to ‘programme’ you with isolated facts and problem types or to monitor your progress. Your job is to fill out that framework with a thorough understanding of the material.

4. You are expected to read the textbook for comprehension. It gives the detailed account of the material of the course. It also contains many examples of problems worked out, and these should be used to supplement those you see in the lecture. The textbook is not a novel; you cannot simply skim through it from start to finish. Reading the textbook must often be slow-going and careful; frequently you’ll need to use pencil and paper to work through the material, but you can work at your own pace.

5. As for when to read the textbook, it’s a good idea to read the appropriate section ahead of the lecture. This way, although you may not understand it
fully, you’ll be prepared for the lecture, and you’ll have a good idea what areas
to ask questions about. If you haven’t looked at the book beforehand, pick up
what you can from the lecture (absorb the general idea and/or take thorough
notes) and count on sorting it out later while studying the book and
transcribing your notes.

6. Laboratories and tutorials are far more important than the marks you might get
for them, because they give you a chance to develop your understanding of the
subject. They are also a good ‘reality check’ for you to see just how much you
really do understand. Use them wisely.

7. In examinations, the examiners set out to probe your mastery of the material in
the course. Primarily, they’ll be looking for your command of the material, as
noted above. You’ll probably have to solve problems you’ve never seen before.
(To be sure, you’ll have encountered similar problems, but they won’t be the
same.) Hence, preparing for examinations simply by remembering lots of
answers without understanding them simply won’t work; examinations test your
understanding of the material as well.

This section is adapted from Teaching at the University Level by Steven Zucker in
Notices of the AMS August 1996.

21 Student Supports

21.1 Programming Support Centre

The Programming Support Centre is available to all Computer Science students free
of charge. The centre operates as a drop-in service where you can get help with any
problems you might have with programming in your courses. For further information,
please visit http://www.scss.tcd.ie/misc/psc/.

21.2 Academic Concerns: Sources of Assistance

- Other students in the class.
- The course lecturer.
- Engineering class representatives.
- Your tutor (or any other tutor if you cannot find yours), or the Senior Tutor.
- The Course Coordinator, Dr Jonathan Dukes.
• The Course Director, Dr Mike Brady.

• The Students’ Union Education Officer, email education@tcdsu.org, web http://www.tcdsu.org.

• Peer Mentors. Junior Freshmen are introduced to their Peer Mentors during Freshers’ Week. The Student to Student Service runs also provides peer mentoring for the other years. For information about all Student to Student services, please email student2student@tcd.ie or phone 8962438.

21.3 Personal Concerns: Sources of Assistance

• Your tutor (or any other tutor if you cannot find yours), or the Senior Tutor, phone 8962251.

• The Student Counselling Service, 3rd Floor, 7–9 South Leinster Street, College. Opening hours: 9:15 am to 5:10 pm Monday to Friday during lecture term. Phone: 8961407. Email: student-counselling@tcd.ie Web: http://www.tcd.ie/Student_Counselling.

• Niteline (Thursday to Tuesday during term time only, 9 pm–2.30 am) Phone: 1800 793 793. Web: http://www.niteline.ie/.

• The College Health Service, House 47, College. Medical Director: Dr David McGrath. Phone: 8961591 or 8961556.

• The Welfare Officer, Students’ Union, House 6, College. Email: welfare@tcdsu.org;

• The Chaplains, House 27, College. Paddy Gleeson and Peter Sexton SJ (Catholic) 8961260 Darren McCallig (Church of Ireland) 8961402 Julian Hamilton (Methodist and Presbyterian) 8961901

• Any student, member of staff or other person with whom you feel able to discuss your concerns;

• Disability Services Coordinator, Mr Declan Treanor, Room 3055, Arts Building, phone: 8963475, email: dtreanor@tcd.ie

NOTE: IF YOU HAVE A CONCERN OF ANY SORT, PLEASE TALK TO SOMEONE STRAIGHT AWAY
21.4 Tutors

A tutor is a member of the academic staff who is appointed to look after the general welfare and development of the students in his or her care. Whilst your tutor may be one of your lecturers, the role of tutor is quite separate from the teaching role.

Tutors are a first point of contact and a source of support, both on arrival in college and at any time during your time in college. They provide confidential help and advice on personal as well as academic issues or on anything that has an impact on your life. They will also, if necessary, support and defend your point of view in your relations with the college.

Here is a table of tutors of Computer Science students. It may be incomplete, and if your tutor’s name is not listed, please let one of us know.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Office Location</th>
<th>Email</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>0A7</td>
<td>Dr Arthur Hughes</td>
<td>Room G.40, O’Reilly Institute</td>
<td><a href="mailto:arthur.hughes@tcd.ie">arthur.hughes@tcd.ie</a></td>
<td>8962459</td>
</tr>
<tr>
<td>0AC</td>
<td>Mr Dermot Geraghty</td>
<td>Floor 4, Parsons Building</td>
<td><a href="mailto:tgerghty@tcd.ie">tgerghty@tcd.ie</a></td>
<td>8961042</td>
</tr>
<tr>
<td>0CN</td>
<td>Dr Aonghus McNabola</td>
<td>3rd Floor, Simon Perry Building</td>
<td><a href="mailto:amcnabol@tcd.ie">amcnabol@tcd.ie</a></td>
<td>8963837</td>
</tr>
<tr>
<td>0CP</td>
<td>Dr Simon McGinnes</td>
<td>Room 1.31, Lloyd Building</td>
<td><a href="mailto:simon.mcginnes@scss.tcd.ie">simon.mcginnes@scss.tcd.ie</a></td>
<td>8962092</td>
</tr>
<tr>
<td>0DX</td>
<td>Dr Bidisha Ghosh</td>
<td>Room 2.1, 2nd Floor, Simon Perry Building</td>
<td><a href="mailto:bghosh@tcd.ie">bghosh@tcd.ie</a></td>
<td>8963646</td>
</tr>
<tr>
<td>0F7</td>
<td>Dr Hugh Gibbons</td>
<td>Room LG20, O’Reilly Institute</td>
<td><a href="mailto:hugh.gibbons@cs.tcd.ie">hugh.gibbons@cs.tcd.ie</a></td>
<td>8961781</td>
</tr>
<tr>
<td>0M8</td>
<td>Dr John Graham</td>
<td>Room 1.3, Museum Building</td>
<td><a href="mailto:jrgraham@tcd.ie">jrgraham@tcd.ie</a></td>
<td>8961165</td>
</tr>
<tr>
<td>0R3</td>
<td>Dr Dermot O’Dwyer</td>
<td>Museum Building</td>
<td><a href="mailto:dwodwyer@tcd.ie">dwodwyer@tcd.ie</a></td>
<td>8962532</td>
</tr>
<tr>
<td>0S7</td>
<td>Ms. Mary Sharp</td>
<td>Room G.34, O’Reilly Institute</td>
<td><a href="mailto:mary.sharp@cs.tcd.ie">mary.sharp@cs.tcd.ie</a></td>
<td>8962732</td>
</tr>
<tr>
<td>0T3</td>
<td>Dr Lucy Hederman</td>
<td>Room G.13, O’Reilly Institute</td>
<td><a href="mailto:lucy.hederman@scss.tcd.ie">lucy.hederman@scss.tcd.ie</a></td>
<td>8962245</td>
</tr>
<tr>
<td>0W2</td>
<td>Dr Cathal Walsh</td>
<td>Room 100, Lloyd Institute</td>
<td><a href="mailto:cathal.walsh@tcd.ie">cathal.walsh@tcd.ie</a></td>
<td>8961731</td>
</tr>
<tr>
<td>0W5</td>
<td>Dr Alan O’Connor</td>
<td>2nd Floor, Civil Engineering Labs</td>
<td><a href="mailto:alan.oconnor@tcd.ie">alan.oconnor@tcd.ie</a></td>
<td>8961822</td>
</tr>
</tbody>
</table>

22 Health and Safety

The College Emergency Number, for safety or personal security concerns, is extension 1999 on the internal College telephone system, +353-1-8961999 from a mobile phone or an external landline.

The Faculty of Engineering, Mathematics and Science (FEMS)—of which our school is part—publishes a Health & Safety Guidance Manual. Copies of this manual are available on request at School Reception. It can be viewed online at http://ems.tcd.ie/assets/docs/10-11%20Health%20and%20Safety%20Booklet%20Fems.pdf.

2011-12-07