Dublin Institute of Technology

BSc (Honours) Computer Science
BSc (Ordinary) Computer Science
Higher Certificate Computer Science

DT228

Programme Document (Part B)

This document was prepared by the
Programme Committee
on behalf of the School of Computing

March 2011

Table of Contents

Table of Tables

1 Introduction

1.1 Overview

This is the Programme Document (Part B) for the programme in Computer Science offered by the School of Computing at the Dublin Institute of Technology. This programme leads to the awards of:

- BSc (Honours) Computer Science
- BSc (Ordinary) Computer Science
- Higher Certificate Computer Science

The documentation is prepared in accordance with the Dublin Institute of Technology Handbook for Academic Quality Enhancement (Revised June 2010) to satisfy the quality assurance requirement for a Dublin Institute of Technology academic award.

1.2 School of Computing Programmes

The School of Computing is one of six schools in the College of Sciences and Health.

The School offers a range of programmes at undergraduate and postgraduate levels, as well as a variety of short course and minor award programmes. The programmes leading to major awards in the School of Computing are:

Undergraduate Full Time

1. BSc (Honours) Computer Science (exit option with BSc (Ordinary), Higher Certificate)
2. BSc (Honours) Computing (exit option with BSc (Ordinary), Higher Certificate)

Undergraduate Part Time

1. BSc (Honours) Information Systems and Information Technology (exit option with BSc (Ordinary), Higher Certificate)
2. Higher Certificate Computing (Recognition of Prior Learning Option)

Undergraduate International (Franchises)

1. BSc (Honours) Computer Science (exit option with BSc (Ordinary), Higher Certificate)

   Offered by partners in Harbin Institute of Technology, Harbin, China

2. BSc (Honours) Computer Science (exit option with BSc (Ordinary), Higher Certificate)

   Offered by partners in Institute of Finance Management, Dar es Salaam, Tanzania

Postgraduate Full Time

1. MSc Computing (Information Technology)
2. MSc Computing (Knowledge Management)
3. MSc Digital Games

Postgraduate Part Time

1. MSc Computing (Information Technology)
2. MSc Computing (Knowledge Management)
3. MSc Computing (Assistive Technology)
4. MSc Computing (Data Analytics)
5. Postgraduate Certificate in Software Development

1.3 School of Computing Staff

The School of Computing has 34 full time academic staff, one full time school administrator and two full time technical staff. All student registration, examination administration and accounting functions are managed at the level of the campus, covering five of the six constituent schools of the College of Sciences and Health located in the Kevin St campus of the Institute, as well as a small selection of other schools from other faculties located in the Kevin St. Other administrative functions are operated at the level of the Institute, including the admissions function. Additionally, the College provides a Head of Learning Development to support the development and implementation of relevant activity in that area, and the implementation of quality assurance, a Head of Research and a Promotion and Recruitment Coordinator to support the marketing of the various programmes in the College of Sciences and Health.

The School has strongly developed its research profile in the past decade. Most staff members participate in research to some degree, with some staff members strongly involved in focussed research areas. This has been driven by two factors: recruitment of staff who are active in research; and facilitation of staff members to pursue research towards the achievement of doctoral qualifications.

There has been a very strong influence from staff research on the taught programmes in the school, in addition to the recruitment of postgraduate research students. Within the various programmes, staff research interests have led to the development of focussed elective

modules, or focussed streams of modules, in advanced areas of computing, such as:

- Universal Design and Assistive ICT
- Artificial Intelligence
- Computer Games Development
- Business Systems Intelligence
- Security and Cryptography
- Mobile Robotics

The school has a strong track record in teaching and learning research. The school has been extremely successful over the past years in gaining funding under the Institute's Teaching and Learning Projects scheme and has successfully won a Teaching and Learning Fellowship as part of this scheme. All teaching and learning projects directly impact the student learning experience and there is an immediate transfer into learning and teaching practice in the school.

1.4 Engagement with Industry

The School has an Industry Liaison committee comprised of two members of staff, who are supported through assignment of a reduced teaching load. The committee is responsible for:

1. Placement of students for work experience.
2. Seeking and maintaining relationships with industrial partners.
3. Providing relevant information on school activity to industrial partners.
4. Identifying and organising research collaboration between the school and industry.
5. Promoting school programmes, in particular part-time programmes, for staff in industry.
6. Organising talks by industry for students on the school's programmes.
7. Liaising with the Institute's career advice service in providing support to students for preparation of CVs, as well as interview preparation and other relevant activity.
8. Recruiting industry practitioners for roles within the school, including final year project supervision, external examiner roles, membership of programme validation and review panels, guest lectureships and any other relevant casual roles that arise from time to time.

9. Organising conferences that involve academic and industrial partnerships.
10. Seeking support from industry for the provision of equipment.

The school currently has strong relationships with the following organisations:

1. Escher Group
2. LanComms
3. Horizon Open Systems
4. Sun Microsystems
5. Hibernia Group
6. HEANet
7. Musgrave
8. CISCO
9. Microsoft
10. SAS Institute

The school has a high proportion of staff with relevant industry experience and excellent contacts in industry. This serves as an input to the committee for the establishment of initial contact, as well as guiding and informing programme development.

The work placement option on this programme has been highly successful for the past year and will continue to be into the future, due to the close and strong links established with industry.

1.5 Engagement with Community

The recently published National Strategy for Higher Education in Ireland identifies the importance of the third and often ignored aspect of higher education's mission – that of engagement. Specifically, it states:

> "The relationship between the institution and the community is particularly important in the context of the promotion and achievement of greater equality in higher education."

In reference to the methods through which this can be achieved, it notes the value of service-learning, which, it states:

> "has the advantage of … providing students with the opportunity to engage in civic endeavours. Service learning is a teaching and learning strategy that integrates meaningful community service with instruction and reflection, to enrich the learning experience,

teach civic responsibility and strengthen communities."

Service-learning has been a part of this programme for the past decade, though typically implemented in an ad-hoc fashion. The inclusion of a specific module on this aspect of, and method of, learning in this version of the programme document represents a valuable and highly relevant response to the increased recognition of the value of engagement with the community for learners.

This programme is actively promoted in the schools and areas surrounding the College campus in Dublin, and more broadly. School visits, workshops and academies help annually which promote the programme and the School in the local community.

1.6 Professional Accreditation

This programme (along with four other programmes in the School of Computing) was successfully accredited by the BCS, Chartered institute for IT, in March 2010. In their final report the review panel made reference to the *strong industrial ethos* of the school.

2 Awards and Continuing Professional Development

The programme is a four year honours degree programme in Computer Science, leading to the following award, after successful completion of all four years:

BSc (Honours) Computer Science

This is a qualification at level 8 on the National Framework of Qualifications.

Students on the programme have the option of exiting (subject to having taken specific modules) the programme after successful completion of years 1, 2 and 3 of the programme, with the following award:

BSc (Ordinary) Computer Science

This is a qualification at level 7 on the National Framework of Qualifications.

Students on the programme have the option of exiting the programme after successful completion of years 1 and 2 of the programme, with the following award:

Higher Certificate Computer Science

This is a qualification at level 6 on the National Framework of Qualifications.

All modules on this programme are available as Continuing Professional Development short courses at their respective levels on the National

Framework of Qualifications.

3 Aims, Objectives and Learning Outcomes

3.1 Programme Aims

The main aim of the programme is to produce graduates for the software industry in Ireland. Graduates with the honours degree will have the necessary skills to integrate quickly into software companies and begin to work effectively within a variety of different teams in such organisations. Such graduates will also be positioned to proceed on to postgraduate study, and will have specialised skills in a specific area of computer science, based on their selections of module groups at the final stage. Graduates with the ordinary degree will have a sufficient understanding of advanced concepts and skills in computer science and software engineering to enter into a variety of positions in the software industry and other related industries. Graduates with the higher certificate will have a very strong foundation across the breadth of computer science, software engineering and information technology to apply for entry level positions in the ICT industry, and to advance onto study at higher levels.

Although primarily oriented as a programme to produce software engineers, software developers and system analysts this programme covers the breadth of the computer science discipline, as is appropriate for an undergraduate qualification, and also allows students achieve specialisation at the final stage as a bridge to postgraduate study. Necessarily, the specialisation does not preclude the student from advancing to further study in another area of computer science, and a breadth is maintained at the final stage by allowing the student select options from a range of diverse areas across the discipline.

Students on this programme will be immersed in the study of computer science both through in-depth treatment of the mathematical and algorithmic basis of the discipline, and a practical exposure to the methods of computer science and software engineering. There is an emphasis on practical skills, as evidenced by the level of time dedicated to laboratory and project work, and the weighting across the various years applied to coursework and non-examination assessment (growing from 45% in first year to 59% in final year).

Graduates of the programme are equipped to work in organisations due to an understanding of organisational structures and the role of computing in organisations, covered throughout the curriculum, but focussed mainly in third year of the programme, where students have the opportunity to apply to participate in work placement, and also study businesses, business plans, innovation and entrepreneurship in a dedicated core module.

3.2 Programme Objectives

The objectives of the programme are multiple, and include the following:

- to facilitate the student in making the transition from broadly-based second level studies to self-directed and self-motivated computing based curricular studies required for both industry and research
- to guide the student's progress through a four year programme of core computer science and software engineering allowing them to develop an understanding of the major areas of computing and its applications
- to guide the student's progress through a comprehensive programme of practical work so as to provide confidence and skill in the many aspects of computer science including but not restricted to programming, systems analysis and design, testing, database technologies, systems support and administration and network programming
- to give each student the opportunity of work experience in a relevant industrial setting, allowing personal and professional development and contextualisation of their field of study
- to have each student carry out a computing based project in the final year so as to provide them with experience in the definition of problems, the formulation, implementation and evaluation of solutions and in the effective communication of the problem and outcome
- to provide the student with presentation and reporting skills and experience as well as personal and career development
- to facilitate students to prepare for postgraduate study and research

3.3 Programme Learning Outcomes

This section outlines the programme learning outcomes. Learning outcomes are categorised using the National Qualifications Authority of Ireland (NQAI) range of standards of knowledge, skill and competence.

The learning outcomes for the three separate exit qualifications are provided in the sub-sections below.

3.3.1 Learning Outcomes for Higher Certificate Computer Science (Level 6)

3.3.1.1 Knowledge

On completion of Stages 1 and 2, the student will be able to demonstrate knowledge and understanding of the fundamental concepts of computer science and software engineering, including

1. Computer architecture and operating systems
2. Computer programming fundamentals, including object oriented programming
3. Algorithm selection, design and implementation
4. Software design, modelling and the software engineering processes
5. Data persistence, database design and implementation
6. Networking and internetworking protocols
7. Mathematics for computing
8. Professional and legal issues relevant to computing
9. Website design and implementation incorporating principles of interface design
10. The organisational context for computing systems

3.3.1.2 Know-how and skills

On successful completion of Stages 1 and 2, the student will be able to:

1. Explain the core concepts of computer science
2. Design, write, document and test procedural programs, with persistent data
3. Demonstrate the wide applicability of discrete mathematics to computing
4. Use both structured and object-oriented programming concepts and implement these concepts in elementary programs
5. Design and implement a robust data model in a relational database and manipulate it using sql
6. Develop complex web based applications, incorporating client, server and database technologies
7. Design complex and appropriate interfaces for users of computer systems
8. Interact effectively with a variety of operating systems, employing automated and non-automated methods

9. Make appropriate selections when faced with certain problems, including:
    - The type of data structures to use in a computer program
    - The type of algorithm to apply to a specific problem
    - The type of data storage to use for a specific application
10. Describe the main organisational issues for computer professionals, including:
    - The role of computing systems in organisations
    - The social, legal and professional issues attached to computing
    - The importance of understanding the role of the user in designing effective software systems

3.3.1.3 Competence

On successful completion of Stages 1 and 2, the student will be able to:

1. Learn to evaluate their own learning and identify learning needs within a structured learning environment
2. Exercise substantial personal autonomy
3. Think logically, and express themselves clearly

3.3.2 Learning Outcomes for BSc (Ordinary) Computer Science (Level 7)

3.3.2.1 Knowledge:

On completion of Stage 3 (exit path), in addition to the learning outcomes of Stages 1 and 2, the student will be able to demonstrate knowledge and understanding of a deep nature in core areas and some advanced areas and organisational issues including:

1. Advanced programming including event driven and advanced object-oriented development
2. Software design, modelling, implementation and testing
3. Database design and implementation
4. Network protocol design and implementation
5. Knowledge engineering and applications of artificial intelligence algorithms
6. Management structures and team based behaviour in organisations
7. Personal initiative, responsibility and autonomy

3.3.2.2 Know-how and Skills

On completion of Stage 3 (exit path), in addition to the learning outcomes of Stages 1 and 2, the student will be able to:

1. Identify business opportunities, develop a business plan and effectively present the plan to others
2. Select and adopt a software design methodology, justifying the selection and implementing it appropriately
3. Design, implement and test complete software systems, incorporating a variety of layers
4. Work effectively with clients and users, and design software to meet user needs and requirements
5. Design and implement a moderately complex database, using advanced features, and design and develop layered procedural transactions to interact with it
6. Build graphic user interfaces using event driven programming in an object-oriented environment

3.3.2.3 Competence

On completion of Stage 3 (exit path), in addition to the learning outcomes of Stages 1 and 2, the student will be able to:

1. Employ personal creativity, insight and innovation to address a range of problems, from technical to interpersonal
2. Take initiative to identify and address learning needs
3. Work individually or in a team on a project or in a professional manner to deliver complete software solutions

3.3.3 BSc (Honours) Computer Science (Level 8)

3.3.3.1 Knowledge

On completion of Stage 4, in addition to the learning outcomes of Stages 1, 2, and 3, the student will also be able to demonstrate:

1. Knowledge and understanding of a advanced nature in more than one specialised area of computer science and software engineering, such as
    - Enterprise computing incorporating design and integration of complex software systems
    - Concurrent and distributed system design and implementation
    - Advanced database design, implementation and

administration

- System security, cryptography and computer forensics
- Artificial intelligence and business system intelligence
- Advanced modelling of software systems, including mobile or embedded systems, or platforms for computer game implementation

2. A basic understanding of how to research and present research findings

3.3.3.2 Know-how and Skills

On completion of Stage 4, in addition to the learning outcomes of Stages 1, 2 and 3 the student will be able to:

1. Employ advanced skills from specialist areas to achieve goals, including the design and implementation of complex software systems, the documentation of the design process, and the testing and deployment of the resulting system
2. Apply advanced algorithms to structured and unstructured problems
3. Clearly illustrate to an audience, either orally or in written form, advanced and complex concepts in a specialist area of computing
4. Identify an appropriate method to adopt for a complex problem in software design, implementation, integration or operation, and justify the selection of that method
5. Justify the choice of a specific problem as an appropriate area for research, and place it in the context of research within the field of computing
6. Relate the study of computing to the organisational context, such that research in computing can be seen to impact on organisational use of computing

3.3.3.3 Competence

On completion of Stage 4, in addition to the learning outcomes of Stages 1, 2 and 3, the student will be able to:

1. Use advanced skills to conduct research
2. Act effectively under guidance in a peer relationship with project supervisors and others in positions of authority
3. Work successfully as part of a team, such as software development team working on medium to complex software

programs/projects or a system support team

4. Work independently to achieve goals, taking ownership of and responsibility for personal achievement
5. Learn how to learn and what to learn to direct personal career choices
6. Demonstrate a deep knowledge of the role of computing, computer science, software engineering and information technology, as well as the practitioners involved in those fields, in organisations and in broader society

4 Admission

4.1 Entry Requirements

There are four methods of entry to the programme:

4.1.1 Regular application via the CAO

Minimum requirements for the programme are the Irish Leaving Certificate with a grade D3 or higher in six subjects at ordinary level including

(i) a grade C3 or higher in ordinary level Mathematics or a grade D3 or higher in higher level Mathematics,

(ii) English or Irish at either level,

(iii) a grade C3 or higher in at least 2 subjects at higher level.

4.1.2 Application with an equivalent qualification

Places may be offered to national and international applicants with a qualification considered equivalent to the minimum requirements, as determined by the Institute.

4.1.3 Mature student and non-standard applications

Places may be offered to mature students and to non-standard applicants (those who have previously completed second level education but are too young to be considered as mature students) who meet certain criteria with respect to suitability, analytical skills and professional experience. Students may be requested to attend for interview.

4.1.4 International Applications

Places may be offered to international applicants in line with the School, College and Institute policy on International applications.

4.2 Entry to later stages of the Programme

Entry to subsequent stages of the programme is normally by passing the previous stage's examinations and assessments. On occasion, it is also possible for students to transfer into the programme following the Institute's Advanced Entry procedures and where the Programme Committee is satisfied as to the equivalency of their qualifications.

4.3 Exemptions

Exemptions can be offered to students based on prior learning in line with the School, College and Institute policies on the recognition of prior accredited and experiential learning.

5 Duration and General Structure

The programme is a four stage (notwithstanding the options for students to exit at an earlier stage), modular programme which can be completed in four years of full-time study. Each stage is split into two semesters. Stages 1, 2 and semester 1 of stage 3 are core and common to all students.

At semester 2 of stage 3 students have the option to take one of the following five paths:

1. Work Placement: Students undertake a placement in industry for six to seven months.
2. Service-Learning and Civic Engagement: Students undertake a project with a community partner for six months, resulting in a substantial deliverable.
3. International Experience: Students participate in an exchange programme with an international partner, according to the Erasmus programme or similar.
4. Specialist Modules and Team Project: Students undertake a team project and four modules in specialist subject areas
5. Specialist Modules and Individual Project: Students undertake an individual project and four modules in specialist subject areas

Students intending to progress to stage 4 of the programme must undertake one of options 1, 2, 3 and 4 above.

Only those students who undertake option 5 above can exit with BSc (Ordinary) Computer Science.

At stage 4 students select eight modules, four per semester, in addition to a Final Year Project. The modules that students select are guided by the module groups determined by the Programme Committee. For example, module groups for each of the following

specialist areas are determined by the Programme Committee:

1. Enterprise Systems Development
2. Applied Intelligence
3. Digital Games Development and Simulation
4. Security and Forensics

Students must select at least one module group (comprised of four modules) which represent a coherent set of modules focussed on a specialised area of Computer Science. The students' remaining modules will be selected from the available options and from other module groups. Modules and module groups will be offered subject to demand and availability of resources.

6 Curriculum and Assessment

6.1 Overview

The General Assessment Regulations of the Dublin Institute of Technology govern all examinations and assessment procedures on the programme except where otherwise specified below.

Assessment on this programme is by written examination and where appropriate practical examination and/or continuous, practical and/or project assessment (referred to as non-exam assessment in the tables below).

Each of the four stages of the programme is presented individually in the subsections 6.2 to 6.10.

For each stage, a table is provided which details:

- The weekly categorised contact hours for that module
- The total contact and non-contact hours expected for a student undertaking that module
- The weighting for each of the forms of assessment for that module

Notes are also provided on the selection of modules, and the delivery order of certain modules.

Section 6.11 onwards provide details on specific assessment regulations which apply to the programme.

6.2 Stage 1

6.2.1 Modules, Contact Hours and Assessment Weighting

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Labora-tory	Tutorial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU1001	Algorithm Design and Problem Solving	5	2	1	1	52	48	100	60%	2 hours	40%
CMPU1005	Communications for Computer Technologists	5	3			39	61	100		N/A	100%
CMPU1006	Computer Architecture and Technology	5	2	1	1	52	48	100	60%	2 hours	40%
CMPU1017	IT Fundamentals	5	3			39	61	100	70%	2 hours	30%
CMPU1018	Mathematics 1	5	2		1	39	61	100	70%	2 hours	30%
CMPU1019	Microprocessor Systems	5	2	2		52	48	100	50%	2 hours	50%
CMPU1022	Operating Systems 1	5	2	1		39	61	100	60%	2 hours	40%
CMPU1028	Programming with Persistent Data	5	1	1	1	39	61	100	70%	2 hours	30%
CMPU1024	Program Design	5	2	1	1	52	48	100	60%	2 hours	40%
CMPU1025	Programming	10	2	2	1	130	70	200	50%	3 hours	50%
CMPU1031	Web Development 1	5	2	2		52	48	100	50%	2 hours	50%
Totals and Averages		**60 (Tot**	**12.5 (Av/W**	**6 (Av/W)**	**3 (Av/W)**	**585 (Tot)**	**615 (Tot)**	**1200 (Tot)**	**55% (Avg)**		**45% (Avg)**

	)	)[1]									

Table 1: Stage 1 Modules, Contact Hours and Assessment Weighting

The pass mark is 40%. General Assessment Regulations apply.

1 **Av/W is the average number of hours per week, averaged across the whole year.**

6.2.2 Module Selection and Delivery

Students are required to complete all modules at this stage. All modules are core for all students.

All modules are delivered in one semester, with the exception of the Programming module, which runs in semesters 1 and 2.

The following three modules can only be taken when the student has completed at least half of the Programming module, or equivalent:

- Microprocessor Systems 1
- Programming with Persistent Data
- Web Development 1

As such, these modules must be offered in semester 2, with Programming running for the entirety of semesters 1 and 2.

Since Program Design is a prerequisite module for Algorithm Design and Problem Solving, Program Design must run in semester 1 and Algorithm Design and Problem Solving must run in semester 2.

6.3 Stage 2

6.3.1 Modules, Contact Hours and Assessment Weighting

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Labora-tory	Tutorial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU2001	Algorithms and Data Structures	5	2	1	1	52	48	100	70%	2 hours	30%
CMPU2005	Data Communications	5	3		1	52	48	100	70%	2 hours	30%
CMPU2007	Databases 1	5	2	2		52	48	100	50%	2 hours	50%
CMPU2008	Human Computer Interaction	5	2	2		52	48	100	60%	2 hours	40%
CMPU2011	Legal and Professional Issues	5	2		1	39	61	100	50%	2 hours	50%
CMPU2012	Mathematics 2	5	2		1	39	61	100	70%	2 hours	30%
CMPU2016	Object Oriented Programming	10	2	2	1	130	70	200	40%	3 hours	60%
CMPU2017	Operating Systems 2	5	2	1		39	61	100	60%	2 hours	40%
CMPU2019	Software Engineering 1	5	1	1	1	39	61	100	70%	2 hours	30%
CMPU2020	Software Engineering 2	5	1	1	1	39	61	100	70%	2 hours	30%
CMPU2022	Web Development 2	5	2	2		52	48	100	50%	2 hours	50%
Totals and Averages		**60 (Tot)**	**11.5 (Av/W)[1]**	**7 (Av/W)**	**4 (Av/W)**	**585 (Tot)**	**615 (Tot)**	**1200 (Tot)**	**58% (Avg)**		**42% (Avg)**

Table 2: Stage 2 Modules, Contact Hours and Assessment Weighting

The pass mark is 40%. General Assessment Regulations apply.

1 **Av/W is the average number of hours per week, averaged across the whole year.**

6.3.2 Module Selection and Delivery

Students are required to complete all modules at this stage. All modules are core for all students.

All modules are delivered in one semester, with the exception of the Object-Oriented Programming module, which runs in semesters 1 and 2.

Since Software Engineering 1 is a prerequisite module for Software Engineering 2, Software Engineering 1 must run in semester 1 and Software Engineering 2 must run in semester 2.

6.4 Stage 3 – Semester 1

6.4.1 Modules, Contact Hours and Assessment Weighting

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Labora-tory	Tutorial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU3005	Business and Enterprise	5	2			26	74	100	50%	2 hours	50%
CMPU3006	Client-Server Programming	5	2	2		52	48	100	60%	2 hours	40%
CMPU3010	Databases 2	5	1	2	1	52	48	100	60%	2 hours	40%
CMPU3017	Graphical User Interface Programming	5	1	2	1	52	48	100	50%	2 hours	50%
CMPU3024	Knowledge Engineering	5	2	2		52	48	100	60%	2 hours	40%
CMPU3038	Software Engineering 3	5	2	1		39	61	100	70%	2 hours	30%
Totals and Averages		**60 (Tot)**	**10 (Total)**	**9 (Total)**	**2 (Total)**	**273 (Tot)**	**327 (Tot)**	**600 (Tot)**	**58% (Avg)**		**42% (Avg)**

Table 3: Stage 3, Semester 1 Modules, Contact Hours and Assessment Weighting

The pass mark is 40%. General Assessment Regulations apply.

6.4.2 Module Selection and Delivery

Students are required to complete all modules in this semester at this stage. All modules are core for all students.

6.5 Stage 3 – Work Placement and Service-Learning and Civic Engagement

6.5.1 Modules, Contact Hours and Assessment Weighting (Semester 1)

Students who apply for and are accepted onto either *Work Placement* or *Service-Learning and Civic Engagement* will be assessed in semester 2 based entirely on non-examination. Their hours will be subject to the nature of their placement, and will cover a longer period of time than the academic semester, and are thus omitted from the table below.

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Labora-tory	Tutorial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU3035	Work Placement *or* Service-Learning and Civic Engagement	30								N/A	100%
Totals and Averages		**30 (Tot)**									**100% (Avg)**

Table 4: Stage 3, Semester 2 Modules, Contact Hours and Assessment Weighting (*Work Placement* and *Service-Learning and Civic Engagement* Path)

6.5.2 Modules, Contact Hours and Assessment Weighting (Overall)

The overall contact hours and assessment weightings for students to take this path at stage 3 are given in the table below:

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Labora-tory	Tutorial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
Semester 1		30	10	9	2	273	327	600	58%		42%
Semester 2		30									100%
Totals and Averages		**60 (Tot)**	**5 (Av/W)[1]**	**4.5 (Av/W)**	**1 (Av/W)**	**273 (Tot)**	**327 (Tot)**	**600 (Tot)**	**29% (Avg)**		**71% (Avg)**

Table 5: Stage 3, Overall Contact Hours and Assessment Weighting (*Work Placement* and *Service-Learning and Civic Engagement* Path)

1 **Av/W is the average number of hours per week, averaged across the whole year.**

6.5.3 Module Selection and Delivery

Both the Work Placement and the Service-Learning and Civic Engagement modules run from the start of semester 2 for a period of six to seven months, including the summer holiday period.

All students who are successful in all the stage 2 exams are eligible to apply for the Work Placement module or the Service-Learning and Civic Engagement module.

Advanced entry students may be considered for the Work Placement module or the Service-Learning and Civic Engagement module subject to approval by the Programme Committee. The Programme Committee may set a threshold on the numbers accepted for either module. Students may, as one of the criteria, be selected based on previous performance.

The School of Computing has an Industry Liaison Committee, a member of which is the Work Placement and Internship Coordinator whose responsibility is to place students in companies to allow them to complete a six to seven month work placement. While significant effort will be made to place students, any students who are not placed before the start of the second semester and are not participating in the Service-Learning and Civic Engagement module must take one of the internal paths for semester 2 of stage 3.

A student who is not successful in the Work Placement module cannot repeat the Work Placement, or apply for Service-Learning and Civic Engagement in their repeating year. They are required to re-enrol and complete one of the internal paths for semester 2 of stage 3.

A student who is not successful in the Service-Learning and Civic Engagement module cannot repeat the Service-Learning and Civic Engagement module, or apply for work placement in their repeating year. They are required to re-enrol and complete one of the internal paths for semester 2 of stage 3.

Procedures, assessment criteria, and guidelines for the structure and content of all Work Placement documentation, reports and presentations are included in the School of Computing Industrial Placement Handbook. This document is available from the Work Placement Co-ordinator, and is also made available to all students on the Web at http://www.comp.dit.ie/industrialplacement.

The Institute will endeavour to provide the student with an alternative placement in the event of a student being unable to continue with the work placement through no fault of their own. In the event of the Institute being unable to provide an alternative placement, the student must continue with the other modules of Semester 2, Stage 3. Depending on the time at which a student position is terminated

alternative modules within stage 3 may not be available to students or considered to be a suitable option by the Programme Committee. The School of Computing will review such cases and may provide an alternative option to the student including but not limited to an unpaid placement within the Institute.

6.6 Stage 3 – Specialist Modules

6.6.1 Modules, Contact Hours and Assessment Weighting

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Labor a-tory	Tut-orial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU3004	Applied Intelligence	5	2	2		52	48	100		N/A	100%
CMPU3007	Cloud Computing	5	2	2		52	48	100	70%	2 hours	30%
CMPU3008	Computational Mathematics	5	2	1		39	61	100	70%	2 hours	30%
CMPU3015	Games Logic and Design	5	2	2		52	48	100	60%	2 hours	40%
CMPU3025	Mobile Robotics	5	2	2		52	48	100	50%	2 hours	50%
CMPU3026	Mobile Software Development	5	1	2	1	52	48	100	50%	2 hours	50%
CMPU3033	Quantitative Methods and Tools for Data Analysis	5	1	1	1	39	61	100	60%	2 hours	40%
CMPU3049	System Administration	5	2	2		52	48	100	60%	2 hours	40%
CMPU3042	System Security	5	2	2		52	48	100	50%	2 hours	50%
CMPU3046	Universal Design and Assistive ICT	5	2	1		39	61	100	70%	2 hours	30%
Totals and Averages		**5 (Av/M)[1]**	**1.8 (Av/M)**	**1.7 (Av/M)**	**0.2 (Av/M)**	**48 (Av/M)**	**52 (Av/M)**	**100 (Av/M)**	**54% (Avg)**		**46% (Avg)**

Table 6: Stage 3, Semester 2 Specialist Modules, Contact Hours and Assessment Weighting

1 **Av/M is the average per module, giving the credits and hours that will be used as the average case in sections below**

The pass mark is 40%. General Assessment Regulations apply.

6.6.2 Module Selection and Delivery

Not all specialist modules will be offered in every academic year. Modules will be selected to run based on resource availability and demand.

Those modules which are offered will run in semester 2.

6.7 Stage 3 – Completing Modules Internally to Progress to Stage 4

6.7.1 Modules, Contact Hours and Assessment Weighting (Semester 2)

Students progressing to stage 4 who do not participate in the Work Placement or Service Learning and Civic Engagement modules, must complete the Team Project and four specialist modules in semester 2, as depicted in the table below. Note that the credits, hours and exam weightings for the specialist modules are taken from the average case as calculated in section 6.6 above.

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Laboratory	Tut-orial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU3045	Team Project	10	1	2		39	161	200		N/A	100%
	Specialist Module 1	5	1.8	1.7	0.2	48	52	100	54%	2 hours	46%
	Specialist Module 2	5	1.8	1.7	0.2	48	52	100	54%	2 hours	46%
	Specialist Module 3	5	1.8	1.7	0.2	48	52	100	54%	2 hours	46%
	Specialist Module 4	5	1.8	1.7	0.2	48	52	100	54%	2 hours	46%
Totals and Averages		**30 (Tot)**	**8.2 (Total)**	**8.8 (Total)**	**0.8 (Total)**	**231 (Tot)**	**369 (Tot)**	**600 (Tot)**	**36% (Avg)**		**64% (Avg)**

Table 7: Stage 3, Semester 2 Modules, Contact Hours and Assessment Weighting (*Internal Path for Progression*)

The pass mark is 40%. General Assessment Regulations apply.

6.7.2 Module Selection and Delivery

Students select four specialist modules from those available, and complete this with the Team Project.

6.7.3 Modules, Contact Hours and Assessment Weighting (Overall)

The overall contact hours and assessment weightings for the entirety of stage 3 for students to take this path at stage 3 are given in the table below:

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Labora-tory	Tutorial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
Semester 1		30	10	9	2	273	327	600	58%		42%
Semester 2		30	8.2	8.8	0.8	231	369	600	35%		65%
Totals and Averages		**60 (Tot)**	**9.1 (Av/W)[1]**	**8.9 (Av/W)**	**1.4 (Av/W)**	**504 (Tot)**	**696 (Tot)**	**1200 (Tot)**	**47% (Avg)**		**53% (Avg)**

Table 8: Stage 3, Overall Contact Hours and Assessment Weighting (*Internal Path for Progression*)

The pass mark is 40%. General Assessment Regulations apply.

1 **Av/W is the average number of hours per week, averaged across the whole year.**

6.7.4 Module Selection and Delivery

Students complete all modules in semester 1, and select four specialist modules from those available to run in semester 2, which are completed with the Team Project in semester 2.

6.8 Stage 3 – Completing Modules Internally to Exit

6.8.1 Modules, Contact Hours and Assessment Weighting (Semester 2)

Students exiting at stage 3 with BSc (Ordinary), must complete the Individual Project and four specialist modules in semester 2, as depicted in the table below. Note that the credits, hours and exam weightings for the specialist modules are taken from the average case as calculated in section 6.6 above.

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Laboratory	Tut-orial / Super-vision	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU3018	Individual Project	10			1[1]	13	187	200		N/A	100%
	Specialist Module 1	5	1.8	1.7	0.2	48	52	100	54%	2 hours	46%
	Specialist Module 2	5	1.8	1.7	0.2	48	52	100	54%	2 hours	46%
	Specialist Module 3	5	1.8	1.7	0.2	48	52	100	54%	2 hours	46%
	Specialist Module 4	5	1.8	1.7	0.2	48	52	100	54%	2 hours	46%
Totals and Averages		**30 (Tot)**	**7.2 (Total)**	**6.8 (Total)**	**1.8 (Total)**	**205 (Tot)**	**395 (Tot)**	**600 (Tot)**	**36% (Avg)**		**64% (Avg)**

Table 9: Stage 3, Semester 2 Modules, Contact Hours and Assessment Weighting (*Exit Path*)

The pass mark is 40%. General Assessment Regulations apply.

1 1 hours supervision is the average number of hours of individual supervision over one semester, which equates to a total of 13 hours supervision in total. If the project takes place over two semesters, or some other length of time, this supervision allowance should be appropriately distributed over that period of time.

6.8.2 Modules, Contact Hours and Assessment Weighting (Overall)

The overall contact hours and assessment weightings for the entirety of stage 3 for students to take this path at stage 3 are given in the table below:

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Labora-tory	Tutorial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
Semester 1		30	10	9	2	273	327	600	60%		40%
Semester 2		30	7.2	6.8	1.8	205	395	600	35%		65%
Totals and Averages		**60 (Tot)**	**8.6 (Av/W)[1]**	**7.9 (Av/W)**	**1.9 (Av/W)**	**478 (Tot)**	**722 (Tot)**	**1200 (Tot)**	**47% (Avg)**		**53% (Avg)**

Table 10: Stage 3, Overall Contact Hours and Assessment Weighting (*Exit Path*)

The pass mark is 40%. General Assessment Regulations apply.

1 **Av/W is the average number of hours per week, averaged across the whole year.**

6.8.3 Module Selection and Delivery

Students complete all modules in semester 1, and select four specialist modules from those available to run in semester 2, which are completed with the Individual Project in semester 2.

Students completing the Individual Project receive 13 hours of individual supervision in the academic year. Under normal circumstances this project will be undertaken over two semesters for all full-time students.

In exceptional circumstances and at the discretion of the Head of School, students may apply to complete it in a single semester. Consideration will be given to their capacity to succeed in their project, based on their past performance and their workload.

6.9 Stage 4 – Option Modules

6.9.1 Modules, Contact Hours and Assessment Weighting (Semester 2)

Students at stage 4 take the Final Year Project module and eight other modules, each of which are considered option modules, although some of these are grouped to form the module groups detailed in section 6.10 below.

The full set of option modules is given in the table below:

Module Code	Module Title	ECTS	Weekly Contact Hours			Total Hours			Assessment		
			Lecture	Laboratory	Tut-orial	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU4003	Advanced Databases	5	2	2		52	48	100	60%	2 hours	40%
CMPU4009	App Development and Commercialisation	5	2	2		52	48	100	50%	2 hours	50%
CMPU4007	Advanced Security 1	5	2	1		39	61	100	60%	2 hours	40%
CMPU4008	Advanced Security 2	5	2	1		39	61	100	50%	2 hours	50%
CMPU4010	Artificial Intelligence 1	5	2	2		52	48	100	70%	2 hours	30%
CMPU4011	Artificial Intelligence 2	5	2	2		52	48	100	70%	2 hours	30%
CMPU4012	Bioinformatics	5	2	1		39	61	100	70%	2 hours	30%
CMPU4013	Business Systems Intelligence	5	2	1		39	61	100	70%	2 hours	30%
CMPU4016	Compilers and Language Design	5	2	2		52	48	100	70%	2 hours	30%
CMPU401	Computer Graphics	5	2	2		52	48	100	70%	2 hours	30%

7											
CMPU4019	Designing and Building Semantic Web Applications	5	2	1		39	61	100	50%	2 hours	50%
CMPU4020	Digital Audio	5	2	2		52	48	100	70%	2 hours	30%
CMPU4021	Distributed Systems	5	2	2		52	48	100	70%	2 hours	30%
CMPU4023	Enterprise Application Dev.	5	2	2		52	48	100	70%	2 hours	30%
CMPU4025	Enterprise Systems and Architecture	5	2	2		52	48	100	60%	2 hours	40%
CMPU4028	Forensics	5	2	2		52	48	100	50%	2 hours	50%
Module Code	**Module Title**	**ECTS**	**Weekly Contact Hours**			**Total Hours**			**Assessment**		
			Lecture	**Laboratory**	**Tut-orial**	**Contact**	**Self-Study**	**Total**	**Exam Weight**	**Exam Duration**	**Non-Exam Weight**
CMPU4030	Games Engines 1	5	2	2		52	48	100	50%	2 hours	50%
CMPU4031	Games Engines 2	5	2	2		52	48	100	50%	2 hours	50%
CMPU4032	Geographic Information Systems	5	2	1		39	61	100	50%	2 hours	50%
CMPU4034	Image Processing	5	2	2		52	48	100	70%	2 hours	30%
CMPU4041	Language Technology	5	2	2		52	48	100	50%	2 hours	50%
CMPU4042	Music Technology	5	2	2		52	48	100	70%	2 hours	30%
CMPU4043	Rich Web Application Technology	5	2	2		52	48	100	70%	2 hours	30%
CMPU4045	Software Quality Management	5	2	1		39	61	100	80%	2 hours	20%
CMPU4046	Spatial Databases	5	2	1		39	61	100	50%	2 hours	50%

CMPU4047	Spatial Statistics and Spatial Knowledge Discovery	5	2	1		39	61	100	50%	2 hours	50%
CMPU4048	Strategic Management	5	3			39	61	100	80%	2 hours	20%
CMPU4051	Systems Software	5	2	2		52	48	100	70%	2 hours	30%
CMPU4052	Systems Software for Embedded Mobile Devices	5	2	2		52	48	100	50%	2 hours	50%
CMPU4054	Universal Design and Web Accessibility	5	2	1		39	61	100	50%	2 hours	50%
CMPU4055	Web Mapping	5	2	1		39	61	100	30%	2 hours	70%
Totals and Averages		**5 (Av/M)[1]**	**2 (Av/M)**	**1.6 (Av/M)**		**47 (Av/M)**	**53 (Av/M)**	**100 (Av/M)**	**61% (Avg)**		**39% (Avg)**

Table 11: Stage 4 Option Modules, Contact Hours and Assessment Weighting

1 **Av/M is the average per module, giving the credits and hours that will be used as the average case in sections below**

The pass mark is 40%. General Assessment Regulations apply.

6.9.2 Module Selection and Delivery

Not all specialist modules will be offered in every academic year. Modules will be selected to run based on resource availability and demand.

Due to timetabling constraints, some combinations of options may not be possible.

6.10 Stage 4 – Full Requirements

6.10.1 Modules, Contact Hours and Assessment Weighting (Semester 2)

Students exiting at stage 4 with BSc (Honours), must complete the Final Year Project and eight option modules, as depicted in the table below. Note that the credits, hours and exam weightings for the specialist modules are taken from the average case as calculated in section 6.9 above.

Module Code	Module Title	ECTS	Weekly Contact Hours				Total Hours			Assessment		
			Lecture	Laboratory	Tutorial	Supervision	Contact	Self-Study	Total	Exam Weight	Exam Duration	Non-Exam Weight
CMPU4027	Final Year Project	20			1	0.5[1]	39	361	400		N/A	100%
	Option 1	5	2	1.6			47	53	100	61%	2 hours	39%
	Option 2	5	2	1.6			47	53	100	61%	2 hours	39%
	Option 3	5	2	1.6			47	53	100	61%	2 hours	39%
	Option 4	5	2	1.6			47	53	100	61%	2 hours	39%
	Option 5	5	2	1.6			47	53	100	61%	2 hours	39%
	Option 6	5	2	1.6			47	53	100	61%	2 hours	39%
	Option 7	5	2	1.6			47	53	100	61%	2 hours	39%
	Option 8	5	2	1.6			47	53	100	61%	2 hours	39%
Totals and Averages		**30 (Tot)**	**8 (Total)**	**6.4 (Total)**	**1 (Total)**	**0.5 (Total)**	**389 (Tot)**	**811 (Tot)**	**1200 (Tot)**	**41% (Avg)**		**59% (Avg)**

Table 12: Stage 4, Overall Contact Hours and Assessment Weighting

The pass mark is 40%. General Assessment Regulations apply.

1 0.5 hours supervision is the average number of hours of individual supervision over two semesters, which equates to a total of 13 hours supervision. If the project takes place over some other length of time, this supervision allowance should be appropriately distributed over that period of time.

6.10.2 Module Selection and Delivery

Students must select one core module group from the set of module groups listed below

Each module group represents a coherent specialisation for the student's final year study.

Not all module groups will be offered in every academic year. Modules will be selected to run based on resource availability and demand.

The Programme Committee may set a threshold on the numbers accepted for a module group. Students may, as one of the criteria, be selected for a module group based on previous performance.

Module Group 1: Enterprise Systems Development

- Semester 1
  - Distributed Systems
  - Advanced Databases
- Semester 2
  - Enterprise Application Development
  - Enterprise Systems and Architecture

Module Group 2: Applied Intelligence

- Semester 1
  - Artificial Intelligence 1
  - Advanced Databases
- Semester 2
  - Artificial Intelligence 2
  - Business Systems Intelligence

Module Group 3: Digital Games Development and Simulation

- Semester 1
  - Artificial Intelligence 1
  - Games Engines 1
- Semester 2
  - Artificial Intelligence 2
  - Games Engines 2

Module Group 4: Security and Forensics

- Semester 1
  - Advanced Security 1
  - Systems Software
- Semester 2
  - Advanced Security 2
  - Forensics

In addition to the module group the student selects, the student will select four other modules. Due to timetabling constraints, some combinations of options may not be possible.

All students are required to carry out a Final Year Project and to submit a project report. This project report requires the students to use the skills they have developed over the programme to research, design and implement a substantial software system. The pass mark for the project is 40%. The project cannot be compensated against another subject. A student who is not successful in the project must repeat the project the following year and is normally permitted only one further attempt at the project unless decided by the Programme Committee.

Procedures and assessment criteria governing the Project are available from the Project Co-ordinator. These guidelines, procedures and assessment criteria are also available on the web for easy access.

All modules run for one semester with the exception of the Final Year Project.

6.11 Exiting with an Ordinary Degree or a Higher Certificate

It is expected that all students will wish to progress from stage to stage. In order for a candidate to be granted a Higher Certificate or an Ordinary Degree, the candidate must apply in writing to the Examinations office. They may not apply for advanced entry into the following Stage until one academic year has elapsed since graduation.

Students who apply to exit with an Ordinary Degree must take a Stage 3 individual project.

6.12 Reassessment of a Module

If a student has not passed a module, they can take the next available assessable component (e.g. supplemental examination) even if this is a component they have already passed, if they would otherwise be denied an opportunity to attempt to progress within the same period of time (they may otherwise need to wait one further academic year to retake the non-examination component).

6.13 Progression

In order to progress from one stage to the next stage the student should pass, be exempt or compensate for all modules in the stage.

6.14 Compensation

Compensation is normally permitted between modules in a stage, in accordance with the General Assessment Regulations except for the following:

1. Compensation cannot be applied to the Individual Project, Team Project, Final Year Project, Work Placement or Service-Learning and Civic Engagement modules
2. The Individual Project, Team Project, Final Year Project, Work Placement and Service-Learning and Civic Engagement modules cannot be used to compensate for any other module in stage 3.

6.15 Final Grade

The student’s final grade is calculated as the weighted average of the marks achieved for the modules taken at their award stage, where each module’s mark is weighted by the number of ECTS credits assigned to that module, as detailed in the General Assessment Regulations.

The award stage for students exiting with a Higher Certificate (Level 6) is stage 2.

The award stage for students exiting with a BSc Ordinary (Level 7) is

stage 3, which must be completed by taking all semester 1 modules, four specialist modules in semester 2 and the Individual Project.

The award stage for students exiting with a BSc Honours (Level 8) is stage 4.

The student's award will be classified according to the rules as set out in the General Assessment Regulations.

The awards of Higher Certificate and BSc (Ord) may be made with the classifications of Distinction, Merit or Pass in accordance with the schedule below:

Average Mark	Classification
>=70%	Distinction
60%- 69%	Merit Upper Division
50%-59%	Merit Lower Division
40% - 49%	Pass

The award of BSc (Honours) may be made with the classifications of First Class Honours, Second Class Honours or Pass in accordance with the schedule below:

Average Mark	Classification
>=70%	First Class Honours
60%- 69%	Second Class Honours, Upper
50%-59%	Second Class Honours Lower
40% - 49%	Pass

7 Programme Development Plan

Stage 1 of the revised programme, proposed in this review, will start for all students registering for the first time on the programme in September 2011

Stages 2, 3 and 4 of the revised programme will start for existing students of the programme registering for these stages in September 2011. One minor exception to this applies for students entering stage 2 in September 2011, who will complete the Microprocessor Systems 1 module at stage 2, as per the previous programme structure, instead of the Operating Systems 2 module, which will be offered for the first time in the academic year commencing September 2012.

For students on stage 4 in the academic year 2011/2012 and 2012/2013, the facility to calculate the BSc (Honours) award grade based on a combination of stage 2, 3 and 4 marks will remain available to them, as per the previous version of the programme document.

8 Syllabi

In this section, a module descriptor is provided for each of the modules of the BSc in Computer Science programme. Each descriptor specifies the module author, description, aim, learning outcomes, assessment strategy, module content and reading references. The assessment strategy provides a suggested assessment strategy for the learning outcomes. It is provided as a guideline only which is subject to change at the discretion of the lecturer and is not to be used as an indicator of either exam or continuous assessment content. The content listed is indicative of the type of content associated with the learning outcomes provided, and may be modified and developed over time.

8.1 Stage 1 Syllabi

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Algorithm Design and Problem Solving**
Program Design		5	CMPU1001	

8.1.1 Algorithm Design and Problem Solving

Module Author

Richard Lawlor

Module Description

This module is concerned with problem solving skills and with the concept of an algorithm and how an algorithm may be arrived at or designed from a problem specification. It relies on ideas introduced in the Program Design module. It introduces elementary algorithms in which the focus is on the workings of the algorithm rather than its implementation detail. To these ends a functional language will be used. A functional language being highly expressive and flexible, allows one to represent problem information and solution steps in such a way that the focus is on the problem domain rather than the details of language syntax as might be the case with an imperative language.

Some elementary sorting and searching algorithms will also be examined. Further use will be made of pseudocode and flowcharts. The use of top down design and stepwise refinement will be incorporated into algorithm design. Various puzzles/problems will be examined from a representation and algorithmic viewpoint. Notion of programming paradigms and levels of abstraction will be introduced.

Module Aims

The aims of this module are:

- To introduce the notion of an algorithm and to help the students see computer programs as algorithms albeit written in a format understandable to a computer or programmer.
- To extend the use of top down design and stepwise refinement techniques in algorithm design.
- To illustrate the use of flowcharts and pseudocode in expressing an algorithm and their mapping onto imperative code.
- To introduce the students to a variety of sorting and searching algorithms.
- To provide an intuitive feel for the workings of some algorithms by executing them on a functional language interpreter.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Algorithm Design and Problem Solving**
Program Design		5	CMPU1001	

- To explore the ideas of algorithm performance and efficiency.
- To encourage the students to think on how certain problems or puzzles may be represented and solved, first on paper and afterwards how this representation and solution could be mapped onto data structure and algorithm.
- To introduce and explore issues relating to state based search.

Learning Outcomes

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

1. Design algorithms for some simple problems.
2. Be well grounded in the use of top down design, stepwise refinement and the use of modularity.
3. Use structure charts in top down design and describe an algorithm with flowcharts and pseudocode.
4. Have knowledge of several sorting and searching algorithms.
5. Analyse some algorithms estimate their run time performance.
6. Implement some algorithms in a functional language. This will also result in a basic competency in an alternative programming paradigm.
7. Analyse some puzzles and demonstrate how state based search can be used to find a solution.
8. Have knowledge of two quite different computational paradigms.

Learning and Teaching Methods

The module will be delivered primarily through lectures, tutorials and laboratory work. The material will be developed in an informal way during lectures. It is envisaged that the students will assimilate much of the material through problem solving and exercises. Emphasis will be placed on allowing the students to examine the exercise problem descriptions and try possible solutions in the laboratory. Tutorials will be used to allow the students to get help in the more difficult areas and provide students with the opportunity for individual assistance from the supervisor.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Algorithm Design and Problem Solving**
Program Design		5	CMPU1001	

Module content

Notion of an algorithm, difference between an algorithm and a program.

More on top down design, stepwise refinement, structure charts. Algorithm description using pseudocode and flowcharts. Translating flowchart or pseudocode into imperative code.

Specify and design elementary sorting and searching algorithms, and introduce binary search trees. Implementation of these algorithms in a functional language. Performance analysis of simple algorithms. Imperative versus functional paradigm.

Puzzles, problem solving stages, eight-puzzle. State space, state representation and state space search.

Module Assessment

Assessment will be based on a two hour end of semester written exam and continuous assessment during the semester.

- Written exam – 60%
- Continuous assessment – 40%

Essential Reading

Course notes

Supplemental Reading

Fethi Rabhi and Guy Lapalme, Algorithms: A Functional Programming Approach, Addison-Wesley.

Web references, journals and other

As specified by the lecturer

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Communications for Computer Technologists**
		5	CMPU1005	

8.1.2 Communications for Computer Technologists

Module Author

School of Computing

Module Description

This module is designed to equip computing students with the necessary skills to function in a working environment. To do this, students must be good communicators and be able to clearly express their ideas to a range of stakeholders using a range of computer-based technologies. This part of the module covers both traditional media of communications (such as the writing of business documentation, particularly software specifications, graphical presentations and oral communications) and the use of network communication technologies (such as e-mail, threaded discussion boards and on-line chat), plus appropriate netiquette.

Additionally the module will cover the necessary study skills to cope with third-level content areas. To this end topics on this module will include time management, textbook studying, note-taking, library usage, using on-line resources, reducing test anxiety, improving concentration, learning memory strategies, and exam and assessment preparation.

Module Aims

The aim of the module is to help students develop an appreciation of the importance of communications in the computing discipline. In most work-based computing environments, the ideas being dealt with are a combination of organisational and technical concepts all of which are often very abstract and complex, therefore the need for clear and concise communication is vital.

Learning Outcomes

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

1. Apply effective note taking techniques in lectures
2. Describe a range of learning theories and learner types
3. Develop and apply an effective time management plan

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Communications for Computer Technologists**
		5	CMPU1005	

4. Apply effective test taking strategies to objective and essay tests
5. Assess personal level of test anxiety and select appropriate strategies for dealing with stress
6. Describe the importance of good communication and the problems with plagiarism
7. Prepare various documentation to an accurate format
8. Communicate effectively using oral, written and internet-based technologies
9. Work effectively in teams and groups.
10. Prepare and deliver a short presentation

Learning and Teaching Methods

The module will be delivered primarily through lectures and tutorials. The material will be developed in an informal way during lectures. It is envisaged that the students will assimilate much of the material through exercises. Much emphasis will be placed on worked examples and group discussion of exercises.

Because this in intended to be of practical use to the students, a large emphasis will be placed on allowing the students to tryout the approaches described in lectures. Each week, the students will be given a number of exercises that cover material relevant to the implementation of approaches discussed. These exercises get progressively more difficult and will incorporate material learned previously. Tutorials will be used to allow the students to get extra tuition in the more difficult areas. They will also be used to allow the students to ask for any extra help required. The tutorials will incorporate the delivery of additional exercises and examples and provide the student with the opportunity for one-to-one assistance from the supervisor.

Module content

Introduction to DIT

Importance of Communications

Learning - individually and in groups and teams

Study Skills- individually and in groups and teams

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Communications for Computer Technologists**
		5	CMPU1005	

Oral Communication and good presentation skills

Written Communication

Graphical communication

On-Line Communications

Module Assessment

Continuous Assessment - 100%

An example of the type of assignments is as follows:

- Assignment based on company case study
- Short research assignment
- Oral presentation

Essential Reading

Cottrell, S., 2003, The Study Skills Handbook, 2nd ed., UK: Palgrave Macmillan.

Paradis, J.G., Zimmerman, M., 1997, The MIT Guide to Science and Engineering Communication, MIT Press

Supplemental Reading

Cottrell, S., 2005, Critical Thinking Skills: Developing Effective Analysis and Argument, UK: Palgrave Macmillan

Tufte, E.R., 1997, Visual Explanations, Graphics Press

Web references, journals and other

College Life at DIT: http://www.dit.ie/DIT/prospective/welcome/college-life.html

Study Tips: Manage Your Own Learning - http://www.ul.ie/~library/studyskills.html

What are Key Skills About? http://www.schoolzone.co.uk/resources/articles/whatare_keyskills.asp

Key Skills Framework - http://www.action.ncca.ie/ga/key-skills/key-skills-framework

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Computer Architecture and Technology**
		5	CMPU1006	

8.1.3 Computer Architecture and Technology

Module Author

Art Sloan, John Kelleher

Module Description

This module presents theoretical aspects of computer science, supporting and enhancing other modules on the course. It also provides the student with an understanding of the operation of typical modern digital computers such as desktops, servers and mainframes.

Module Aims

The aim of this module is to provide the student with the theoretical foundations for other modules on the course by providing the student with the necessary background knowledge to understand modern computers – how they are physically structured and how they operate. It should expose the student to the internals of typical modern computers and provide a general overview of their operation in terms of both hardware and software, particularly the operating system.

Learning Outcomes

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

1. Demonstrate a knowledge of number systems, Boolean algebra, sets, logic, relations and functions
2. Identify and describe the major components of a typical general-purpose computer
3. Describe the operation of an idealised generic computer
4. Outline where a typical modern computer deviates from the idealised version
5. Use the course topics to solve computing problems
6. Use appropriate software and hardware to solve problems
7. Identify, and differentiate between, different types of computer systems
8. Identify, and describe the operation of, basic logic circuits
9. Convert between, and perform elementary arithmetic and other operations in, number systems including binary, octal, decimal

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Computer Architecture and Technology**
		5	CMPU1006	

and hexadecimal

Learning and Teaching Methods

Lectures, self-study, tutorials and any combination of discussion, case study, problem solving exercises, readings, seminars and computer-based learning.

Module content

Introduction and History: from valves to integrated circuits

Electrical fundamentals: Voltage, current, resistance, Ohm's law

Boolean Algebra: Basic laws, simplification of expressions, application to switching circuits Number Systems: Binary, octal, decimal, hexadecimal, simple binary arithmetic

Logic gates: AND, NAND, OR, NOR, XOR, NOT

Boolean algebra and combinational logic including half- and full-adders

Sequential logic: latches, flip-flops, shift-registers, and counters

Binary, octal, and hexadecimal representations and busses

Introduction and History: from Babbage to multi-core processors

Von Neumann architecture and stored program computing: ALU, memory, program counter (or instruction pointer), registers, instruction decoder, fetch-execute-store cycle, data and address busses

Components: interfaces, hardware, and operation of: disk, memory, serial, parallel, graphics, audio and network subsystems

Contemporary and historical removable media (e.g. floppy disk, tape, CD, DVD, flash key)

Contemporary and historical external interfaces (e.g. PC parallel port, PC serial port, PS/2 ports, USB, IEEE-1394 "Firewire", SCSI, Ethernet)

Contemporary and historical internal interfaces (e.g. AGP, SATA, PCIe, PATA, PCI, ISA, EISA, NuBus, MCA, AGP, VESA)

Module Assessment

Methods of assessment to be used to measure the learning outcomes stated above are to be: a written examination and continuous assessment, including one or more of assignment, essay, problem-solving exercise and/or class or lab tests.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Computer Architecture and Technology**
		5	CMPU1006	

Examination: 60%

Continuous Assessment: 40%

Essential Reading

Englander, I. (Latest Edition), The Architecture of Computer Hardware and System Software: An Information Technology Approach, John Wiley & Sons, New Jersey, USA

Supplemental Reading

Comer, D.E. Essentials of Computer Architecture, 1st Edition, Pearson Education, New Jersey, USA, 2004,

Web references, journals and other

http://www.informaworld.com

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**IT Fundamentals**
		5	CMPU1017	

8.1.4 IT Fundamentals

Module Author

Art Sloan, Patricia O'Byrne

Module Description

This module provides an overview of the discipline of Information Technology (IT) and describes how it relates to other computing disciplines.

Module Aims

The aim of this module is to help students understand the diverse contexts in which IT is used and the challenges inherent in the diffusion of this type of technology.

Learning Outcomes

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

1. Describe the role of the IT professional as the user advocate.
2. Explain data quality and systems security
3. Explain how the components of an IT system interrelate.
4. Understand the issues of management of complexity in an information technology environment by applying best practices
5. Illustrate the use of information and communication technologies to solve problems
6. Outline the history of computing technology, the Internet, and the World-Wide Web
7. Explain the relationship between IT and related and informing disciplines
8. Explain how and to what extent IT has changed various application domains.

Learning and Teaching Methods

Lectures, self-study, tutorials and any combination of discussion, case study, problem solving exercises, readings, seminars and computer-based learning.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**IT Fundamentals**
		5	CMPU1017	

Module content

User centeredness and advocacy

Information assurance and security

IT systems model

Management of complexity

Information and communication technologies: HCI, Networking

History of computing technology

Related disciplines: software Engineering, Mathematics and Statistics

Application domains: science, business, legal issues

Module Assessment

Methods of assessment to be used to measure the learning outcomes stated above are to be: a written examination and continuous assessment, including one or more of assignment, essay, problem-solving exercise and/or class or lab tests.

Examination: 70%

Continuous Assessment: 30%

Essential Reading

Parsons, J. and Oja, D. New Perspectives on Computer Concepts 2011, Course Technologies - Cengage Learning, Inc., Kentucky, USA, 2010

Supplemental Reading

Tajfar, A. Comprehensive Review of Information Technology Fundamentals, Virtualbookworm.com Publishing, Texas, USA

Web references, journals and other

http://www.informaworld.com

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mathematics 1**
		5	CMPU1018	

8.1.5 Mathematics 1

Module Author

Shane Mulligan

Module Description

This is an introductory module, to give the student a range of basic mathematical skills, and a knowledge of how they are applied in various areas of computing and computer science. A number of common mathematical structures and methods will be presented, and their application to represent and solve simple problems. Their application to various areas in computing will be demonstrated.

Module Aims

The aim of this module is to give the student the basic knowledge and competence to deal with mathematical concepts and problems that arise in computer science. It will give the student an understanding of discrete mathematics, and demonstrate the wide applicability of discrete mathematics to computing. It will present mathematics as an exact science, and train the student to think logically, and express themselves clearly.

Learning Outcomes

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

1. Describe number theory concepts and how they can be used in computing.
2. Perform the operations associated with sets, relations and functions, and relate practical examples to the appropriate set, function, or relation model. Describe symbolic logic and how it can be used to model real-life situations, e.g. represent sentences.
3. Define sequences and series, and their definition using iteration.
4. Explain and apply the rules for indices, and logs using base 2 and base 10.
5. Describe matrices, and their operations, and apply simple matrix algebra.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mathematics 1**
		5	CMPU1018	

6. Describe and compute basic statistics and their application to data presentation and analysis.

Learning and Teaching Methods

The learning and teaching methods will consist of lectures and tutorials. Exercises and continuous assessments will be given to ensure that the student understands and masters the material, and to give them practice at representing and solving simple problems. Computer based or online methods may also be used for assessment.

Module content

Number Theory: Division Algorithm, Euclidean algorithm, Prime numbers,

Fundamental Theorem of Arithmetic, Modular arithmetic, relevance to computing.

Indices and logs. Arithmetic of, application and manipulation rules.

Set Theory: Definition. Algebra of sets, set operations, subsets, power set, Venn diagrams. Cartesian product, Computer representation of sets.

Relations: Definition, Binary relations. Equivalence relation properties, and application to databases.

Sequences and series. Definitions, sum of integers 1 to N, sum of squares.

Functions: Onto and one-to-one functions. Composition of functions, inverse functions. The floor function and the ceiling function. Linear and quadratic functions. Arithmetic operators and operator precedence. Application of the concept of a function to computer programming and to the computational complexity of algorithms.

Boolean logic. Logic gates.

Propositional Logic. Propositions, operators, representing English sentences with propositions.

Matrices: Definition, Matrix algebra (addition, multiplication, inverse). Application to representing systems of equations. Storing large data sets. Applications in computing, e.g. computer graphics, and computer representation of relations.

Statistics: Data collection and presentation in tables, stem-plots and

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mathematics 1**
		5	CMPU1018	

histograms. Summarizing and describing numerical data. Measures of central tendency and spread of data, mean, mode, variance, standard deviation. Frequency distribution.

Module Assessment

The Module assessment will be by written examination (70%), and continuous assessment (30%).

Essential Reading

Kenneth H. Rosen, 2003, Discrete Mathematics and its Applications, 5th Edition, McGraw-Hill.

Seymour Lipschutz, Marc Lipson, 1997, Schaum’s outlines Discrete Mathematics 2nd Edition, McGraw-Hill.

Supplemental Reading

Seymour Lipschutz, Marc Lipson, 1997, Schaum’s outlines Discrete Mathematics 2nd Edition, McGraw-Hill.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Microprocessor Systems**
		5	CMPU1018	

8.1.6 Microprocessor Systems

Module Author

Frank Duignan, Richard Hayes

Module Description

This module introduces the students to hardware and software aspects of microprocessor and microcontroller system development.

Module Aims

The aim of this module is to provide students with an understanding of the inner operation of computers and the way in which they interact with external devices.

Learning Outcomes

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

1. Design, write and debug simple assembly-language programs for the 80x86 family of microprocessors.
2. Design, write and debug simple C-language programs for the 80x86 family of microprocessors and a microcontroller.
3. Perform arithmetic in the binary and hexadecimal number systems.
4. Explain the role played by (and limitations of) the stack in a high level programming language.
5. Explain the operation of simple circuits that interface microprocessors to external devices.
6. Outline the operation of microprocessor system elements such as RAM, ROM, Timers and communications peripherals.
7. Discuss the encoding of different types of computer data (e.g. ASCII characters, Unicode, floating point numbers).
8. Discuss interrupt handling in microprocessor systems.

Learning and Teaching Methods

Lectures, laboratory work, self-learning.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Microprocessor Systems**
		5	CMPU1018	

Module content

Analogue systems vs. Digital systems

The binary and hexadecimal number systems, signed and unsigned numbers.

Microprocessor core elements: Registers, flags, calculation units, buses, RAM, ROM, Parallel I/O ports.

Microcontroller programming : registers, memory map, I/O port mapping.

Interfacing with actuators and sensors

Analogue to digital and digital to analogue conversion

Introduction to interrupts.

Introduction to serial communications

Writing simple 80x86 assembler programs, addressing modes, decision making, looping.

Calling subroutines and the behaviour of the stack.

Relating simple C-programs to Assembler programs : Allocation of variables, assignment of values to variables, looping, decision making.

System calls.

The C-calling convention.

Mixed language programming

Code optimization

There are three sections in the laboratory programme:

1. Microcontroller labs (4 weeks)
2. x86 labs (4 weeks)
3. Mini project (4 weeks). The mini project is typically a microcontroller application which includes input/output and structured programming.

Module Assessment

There are two main sections to the module mark:

Section 1:

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Microprocessor Systems**
		5	CMPU1018	

- Two online/automated tests, one dealing with the 80x86 family of processors, the other with a microcontroller. Each of these accounts for 10% of the total module mark.
- Laboratory reports account for 20% of the module mark.
- Mini project work accounts for 20% of the module mark.

Section 2:

- The remaining 50% of the module marks are assigned to an end of module written exam which covers all of the course material. Students are expected to answer 3 questions out of a total of 4.

Section 1 accounts for 50% of the total module mark.

Similarly, Section 2 accounts for 50% of the total module mark.

Assessment of specific learning outcomes:

- (1),(2),(5),(8) Assessment based on performance in laboratory
- (3) to (8) Assessed using online tests and end of module test.

Essential Reading

On-line class notes

Supplemental Reading

Brey, Barry B., 1994 The Intel microprocessors : 8086/8088, 80186, 80286, 80386, and 80486 architecture, programming and interfacing. Prentice Hall.

Web references, journals and other

Microcontroller datasheets and programming guides will be available on module website.

Students will also be expected to download, test and modify various sample programs from the module website.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Operating Systems 1**
		5	CMPU1022	

8.1.7 Operating Systems 1

Module Author

Michael Gleeson

Module Description

This module will serve as an introduction to Operating Systems. It provides an overview of the major components of a computer system and their interaction with the systems software. The module provides a fundamental understanding of the concepts of operating systems. Students will also learn how and why operating systems have evolved over years and the impact this has had on modern operating systems. The concepts will be reinforced with practical laboratory exercises in operations systems functionality, user interaction and management. This will be further backed up by a focus on command line interaction with various operating systems. Practical assignments will be given to develop practical operating systems skills. The module will, at a basic level introduce networked, client-server and distributed operating systems to the student. The module will provide the fundamentals for Advanced Operating Systems and the groundwork for other modules in computer science that assume a general understanding of operating systems principals and practice.

Module Aims

The aims of this module are to:

- introduce the student to the principals of operating systems design
- give the students a working knowledge of a modern operating system
- provide the student with a sound knowledge of the various components and interactions of a modern operating system
- facilitate a competency in practical interaction with an operating system

Learning Outcomes

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

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Operating Systems 1**
		5	CMPU1022	

1. Explain the benefits of an operating system in a computing environment
2. List and describe the major components of an operating system and their basic functions
3. Discuss the fundamental trade-offs involved in the design of operating systems
4. Differentiate between the concept of processes and threads of control
5. Classify scheduling policies with examples from different operating systems
6. Appraise memory management techniques and virtual memory implementations
7. Examine various file systems and illustrate their relationship with the IOCS
8. Compare and contrast the strengths and weaknesses of different modern operating system
9. Discuss networked, client-server and distributed operating systems and how they differ from single user operating systems
10. Display and perform proficient command line interaction with various operating systems

Learning and Teaching Methods

In this module a number of teaching methods will be employed including lectures, practical sessions, tutorials and case studies. At least one industrial seminar may be arranged. Focus should be placed on empowering the students to develop their skills independently of the presence of a tutor or lecturer.

This can be aided by the introduction and use of VLE resources, examples including online student discussion groups, reflective blogs for use immediately after practical sessions and voluntary Q&A sections.

Module content

Introduction: Definition of an operating system, abstract views of an operating system, functions of an operating system, event-driven systems, efficiency & system performance goals, evolution of operating

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Operating Systems 1**
		5	CMPU1022	

system designs, classes of operating systems and examples of operating systems.

Process and Threads: Process and programs, programmers view of processes, operating systems view of processes, concurrency, process states, thread of control, interacting processes.

Scheduling: Non pre-emptive scheduling policies, pre-emptive scheduling policies, scheduling in practice, real-time scheduling, example scheduling in UNIX, Linux and Windows.

Memory Management: Memory hierarchy, address spaces, static and dynamic memory, memory allocation to a process, continuous memory allocation, non-continuous memory allocation, swapping and relocation, paging, segmentation, paging with segmentation. Virtual memory basics, demand paging, page replacement policies, memory allocation to a process, page faults.

File System & IOCS: Files and file operations, directories and directories operations, pathnames and filenames, multiple file systems, file types, file sharing, links and shortcuts, file locking, file attributes, disk structure, examples of UNIX, Linux and Windows file systems. Architecture of the IOCS, device drivers, types of devices, buffering, device driver structure.

Multiprocessor Systems: Multiprocessor systems, multicomputer systems, clients and servers, distributed file systems, distributed processing, introduction to thin client computing.

Laboratory Work: In addition to the lecture material studied in class, a weekly lab session focusing on Linux and UNIX-like operating systems will be scheduled. This session will be a hands-on approach to understanding and using the basics of Linux and UNIX-like operating systems. Topics covered include basic Linux commands, working with file systems, process management, proficient knowledge of the vi editor, working with shells, a brief introduction to shell scripting.

Module Assessment

This module should have a 60% weighting for the examination and a 40% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, it is equally important that they can demonstrate an understanding of the theoretical aspects of Operating Systems.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Operating Systems 1**
		5	CMPU1022	

Essential Reading

Ida M. Flynn & Anne McIver McHoes, 2008, Understanding Operating Systems, Thompson Leaning.

John English, 2005, Introduction to Operating Systems: Behind the Desktop, Palgrave McMillian.

D.M. Dhamdhere, 2007, Operating Systems: A Concept based Approach, McGraw Hill.

Supplemental Reading

William Stallings, 2009, Operating Systems: Internals and Design Principles, Prentice Hall.

Web references, journals and other

http://williamstallings.com/OS/OS6e.html

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Programming with Persistent Data**
		5	CMPU1028	

8.1.8 Programming with Persistent Data

Module Author

Programming and Algorithms Group

Module Description

This module introduces the fundamental programming techniques required for the efficient storage and retrieval of data.

Module Aims

The aims of this module are to:

- Teach the fundamentals of data storage methods and file formats
- Teach the procedural programming techniques to implement various storage methods and formats.

Learning Outcomes

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

1. Distinguish between various data storage methods and formats
2. Choose the most appropriate data storage method for a specified requirement
3. Design and write procedural programs to store and retrieve data in an efficient manner

Learning and Teaching Methods

The module will be delivered primarily through lectures, tutorials, self-directed learning and practical laboratory exercises. A Virtual Learning Environment (VLE) is used extensively in this module.

Module content

File Structure: bytes, fields, records,attributes, rows, columns.

File Types: ASCII, Binary, Relational

File Access: Serial, Sequential, Random, Indexed, Index Sequential.

Data Manipulation: Creating persistent data, Retrieving persistent data, Updating persistent data, Deleting persistent data.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Programming with Persistent Data**
		5	CMPU1028	

File Design: Efficient design of file structure and content.

Common File Techniques: Multiple-file matching, sorting, merging, filtering.

File Security: Reading and writing file permissions.

Programming with Common File Formats: Reading and writing using common propriety software formats, e.g. spreadsheet, database, HTML

Module Assessment

Continuous Assessment (30%):

- Individual assignment
- Lab test

Written examination (70%)

- One two hour, end of module examination.

Essential Reading

Depending on the procedural language used in this module, specific reading lists will be specified in advanced of the start of the module.

Supplemental Reading

Born, Gunter,1995, The File Formats Handbook, Wadsworth Publishing Company

Further Details

Class size is expected to be 80, which can be divided into smaller groups for labs and tutorials.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Program Design**
		5	CMPU1024	

8.1.9 Program Design

Module Author

Richard Lawlor and Shane Mulligan

Module Description

This module is concerned with program design skills, with particular reference to using flowcharts, pseudocode and programming language constructs to model and design computer programs. Consideration is given as to how problem information might be represented in code or on paper and what program design steps may be performed to arrive at a solution. Abstraction, modularity and top-down design are central to this module.

Module Aims

The aims of this module are:

- To introduce students to program design and the main techniques of program design. In particular, to introduce students to design strategies such as top-down and bottom-up and the techniques of stepwise refinement.
- To introduce the use of pseudocode and flowcharts in program design.
- To convey to students an understanding and appreciation of the power of abstraction whereby the essential information relating to a programming problem is abstracted and mapped onto programming constructs.
- To convey the importance of a well conceived design before rushing into code.

Learning Outcomes

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

1. Abstract problem information and represent it on paper or an appropriate computing environment.
2. Demonstrate a basic competence in the use of a program constructs to solve a problem

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Program Design**
		5	CMPU1024	

3. Develop solutions to some elementary program design problems using top down design and stepwise refinement.
4. Describe some simple program designs using pseudocode and flowcharts, and then implement the design.

Learning and Teaching Methods

The module will be delivered primarily through lectures, tutorials and laboratory work. The material will be developed in an informal way during lectures. It is envisaged that the students will assimilate much of the material through problem solving and exercises. Emphasis will be placed on worked examples and group discussion of exercises.

Practical work will consist of weekly laboratory sessions. This will also help the students understand how program design concepts can be mapped to a program language constructs.

A large emphasis will be placed on allowing the students to tryout the problem descriptions and possible solutions in the laboratory. Exercises will be provided that cover material relevant to the implementation of problem solutions. These exercises get progressively more difficult and will incorporate material learned previously.

Tutorials will be used to allow the students to get help in the more difficult areas and provide students with the opportunity for individual assistance from the supervisor.

Module content

Problem Solving, Stages in Problem Solving.

Data Types and Data Representation.

Program Constructs.

Abstraction, Problem Specification, Approaches to Problem Solving and Program Construction, Divide and Conquer, Stepwise Refinement, Top Down Design, Bottom Up Design.

Recursion. Greatest Common Divisor, Factorial and Fibonacci.

Pseudocode and Flowcharts in Program Design.

Linear Data Structures - arrays and lists.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Program Design**
		5	CMPU1024	

Module Assessment

Assessment will be based on a two hour end of semester written exam and continuous assessment during the semester.

- Written exam - 60%
- Continuous Assessment - 40%

Essential Reading

No specific textbook. Lecture notes and laboratory material as provided by the lecturer.

Supplemental Reading

Maureen Sprankle and Jim Hubbard - Problem Solving and Programming Concepts.

Web references, journals and other

As specified by the lecturer

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Programming**
		10	CMPU1025	

8.1.10 Programming

Module Author

Programming and Algorithms Group

Module Description

The module teaches the fundamental principles required to design, write, test and document structured procedural programs.

Module Aims

The aims of this module are to:

- Teach the fundamentals of procedural programming
- Teach the principles of good program design, implementation, documentation and testing.

Learning Outcomes

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

1. Apply basic problem solving techniques to design a program using appropriate modules and data structures to specified requirements.
2. Implement a program using previously a developed design
3. Use an Integrated Development Environment (IDE) proficiently to develop programs
4. To understand the compilation/link processes and interpret errors generated
5. Design appropriate test data to ensure module and program correctness and robustness
6. Debug a program using an IDE and by program tracing
7. Write documentation for a program.

Learning and Teaching Methods

This module will be taught using lectures, practical sessions in the laboratory and tutorials. All theoretical material will be taught in lecture class and this will include the rules and syntax of procedural

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Programming**
		10	CMPU1025	

programming. Each lecture will include many example programs to show the students how the material covered in the lecture is implemented.

Due to the practical nature of programming, a large emphasis will be placed on allowing the students to practice the development of programs in the laboratory. Each week, the students will be given a number of programming exercises that cover all material taught to them in their lecture class.

Tutorials will be used to allow the students to get extra tuition in the more difficult areas. They will also be used to allow the students to ask for any extra help required. The tutorial will incorporate additional exercises and the ability for the supervisor to provide more one-to-one assistance with a student.

Extensive use of a Virtual Learning Environment (VLE) is used in this module.

Module content

Introduction: What is a program? Source code. Machine code. Editing, Compiling Linking, Debugging. Use of an Integrated Development Environment (IDE). The command line.

Basic Data Types: integer, floating-point and character data and variables.

Basic Input-Output: Display data on a screen. Input data from the keyboard.

Programming Structures: Conditional statements: Boolean values and expressions, logical and relational operators, if-statement, case-statement, compound conditional statements.

Iterative constructs: while-statements, for-statements and nested control statements.

Introduction to Data Structures: Strings, single-dimensional arrays, two-dimensional arrays, dynamically allocated arrays, user-defined structures, abstract data types, and enumerated data types. Command line arguments.

Structured Programming: functions, parameter passing, returning values, global and local variables, nested functions, reusable code, library functions.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Programming**
		10	CMPU1025	

Implementing Basic Algorithms: Summation, counting, numeric operations, swapping, maximum and minimum, simple string and array manipulation.

Testing and debugging: Objectives and principles of testing. Choosing appropriate test data. Testing and debugging strategies. Debugging using an IDE. Debugging using a program trace.

Documentation: Writing user and technical documentation. Style guidelines.

Module Assessment

Assessment of the module is a combination of the following:

Continuous Assessment (50%)

- Individual assignments
- Lab tests
- On-line tests
- In-class written tests

Written examination (50%)

- One three hour, end of module examination.

Essential Reading

Depending on the procedural language used in this module, specific reading lists will be specified in advanced of the start of the module.

Supplemental Reading

Dependent on the procedural language used.

Web references, journals and other

Dependent on the procedural language used.

Further Details

Class size is expected to be 80, which can be divided into smaller groups for labs and tutorials.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Development 1**
		5	CMPU1031	

8.1.11 Web Development 1

Module Author

Ciarán O'Leary

Module Description

This module provides the student with the skills necessary to develop web sites with multiple pages, dynamic presentation and style independent from content throughout the site.

Module Aims

The aims of this module are to:

- Provide students with excellent skills for client side web development using popular markup languages
- Provide students with the skills necessary to design the appearance of a web resource independently of the content
- Provide students with the skills necessary to dynamically generate or alter the contents or appearance of a web resource using a client side scripting language such as JavaScript.

Learning Outcomes

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

1. Describe the underlying architecture of both the World-Wide-Web
2. Evaluate web sites according to well known criteria for effective web design.
3. Describe and employ effective design approaches when building web resources.
4. Implement web pages using HTML, XHTML and CSS
5. Generate web content dynamically and interact with users of web resources using a client side scripting language such as JavaScript.
6. Maintain state across a number of web resources using client side cookies.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Development 1**
		5	CMPU1031	

Learning and Teaching Methods

Web technologies, given their simplicity relative to other technologies, provide the student with the ideal mechanism to quickly observe how a software developer follows a design methodology to build a product. It is important that challenging projects be given to students so that they are encouraged to expand upon the basic knowledge presented in class. The most effective way to learn a web technology is to implement systems using those technologies, so substantial practical time should be devoted to observing how the student is progressing and directing them towards the myriad online resources that can be employed to advance their knowledge. It is suggested that the lecture time be used to provide both a high level explanation of a given technology, with some of its more powerful aspects treated in detail. Focus should be placed on empowering the students to develop their skills independently of the presence of a tutor or lecturer.

Module content

World-Wide-Web: Relationship to the Internet. Relationship to E-Mail, FTP, Telnet and other Internet technologies. Using search engines, particularly the advanced features of popular search engines. Using discussion boards, bulletin boards and other online collaboration or knowledge sharing resources. Overview of the full space of web technologies. Simple introduction to the potential future directions for the World-Wide-Web such as the Semantic Web and agent technology.

Client Side Markup Languages Introduction to HyperText. HTML and XHTML. Layout using tables, frames, layers. Incorporating images, imagemaps, Applets and other components. Accepting input using forms. Submitting forms. Embedding audio, video and other multimedia resources. Commenting, documentation, indentation style.

Separation of Style and Content: Cascading StyleSheets (CSS). Internal and external style sheets. Style local to elements in a web page. Introduction to XML. Simple XML DTD. XSL-T. XML languages such as RSS.

Dynamic HTML: Introduction to scripting. Interpreted languages as opposed to compiled languages. Basic scripting programming language structures: arrays, variables, functions. Generating output by writing code directly and by manipulating the Document Object Model. Capturing events. Taking input from HTML forms.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Development 1**
		5	CMPU1031	

Maintaining State: Client side cookies. Generating cookies for a session. Generating cookies for multiple sessions. Manipulating cookies. Receiving input from cookies. Deleting cookies. Overview of the role of cookies in web applications.

Module Assessment

This module should have a 50% weighting for the examination and a 50% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, it is equally important that they demonstrate an understanding of the theoretical aspects of both the World-Wide-Web and web design and development. This can be assessed in the examination, alongside the assessment of some technical content. It is suggested that the student not be required to write large amounts of code in the examination, with a preference for demonstrating their ability by identifying how code would behave, or how code could be modified.

Essential Reading

W3schools.org

Supplemental Reading

Jennifer Niederst Robbins, Aaron Gustafson (2007), Learning Web Design: A Beginner's Guide to (X)HTML, StyleSheets, and Web Graphics

Web references, journals and other

World-Wide-Web Consortium -http://www.w3.org/

HTML and CSS reference : http://www.htmlhelp.com/

JavaScript Reference : http://wp.netscape.com/eng/mozilla/3.0/handbook/javascript/

8.2 Stage 2 Syllabi

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Algorithms and Data Structures**
Programming Algorithm Design and Problem Solving		5	CMPU2001	

8.2.1 Algorithms and Data Structures

Module Author

Richard Lawlor

Module Description

This module is concerned with algorithms, data structures and their implementation. Standard algorithms & data structures and their associated applications will be presented. Other foundational aspects of computer science such as complexity will introduced. The theory and application of this subject area is a cornerstone of computer science. It is a well established and traditional area and has applications and implications in all areas of computing. The practical side of this subject involves complex coding and reinforces programming skills. Because of the importance of Abstract Data Types (ADT), some knowledge of object-oriented programming is necessary.

Module Aims

The aims of this module are to:

- introduce the students to the subtlety of algorithms and data structures.
- expose the students to a variety of standard algorithms and data structures
- convey the notion of an Abstract Data Type.
- convey to students some of the theory and application of algorithms and data structures.
- show how the concepts involved can be expressed in a programming language and at the same time expose the students to sophisticated programming.
- advance understanding of algorithm design

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Algorithms and Data Structures**
Programming Algorithm Design and Problem Solving		5	CMPU2001	

- show students how to establish an analysis of complexity and algorithm performance issues.
- give some appreciation of the theoretical aspects of computer science.

Learning Outcomes

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

1. Demonstrate an understanding of abstract data types and of the importance difference between interface and implementation.
2. Have been knowledge of a variety of data structures and algorithms.
3. Demonstrate knowledge of the intricacies of implementing complex data structures and their associated algorithms.
4. Discuss the practical application of various algorithms.
5. Have acquired some sophisticated programming skills.
6. Analyse algorithms and understand the issues involved in algorithm complexity and performance.
7. Have a solid understanding and practical experience of algorithm design.
8. Be aware of some issues that concern theoretical computer science.

Learning and Teaching Methods

This is a one semester module and it will be delivered primarily through lectures, tutorials and laboratory work. The material will be developed in an informal way during lectures. It is envisaged that the students will assimilate much of the material through paper based simulations of the algorithms and exercises. There will also be weekly laboratory work to give the students a more practical appreciation of algorithm implementation and to allow them compare algorithms for

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Algorithms and Data Structures**
Programming Algorithm Design and Problem Solving		5	CMPU2001	

efficiency.

Tutorials will be used to allow the students to get extra tuition in the more difficult areas. They will also be used to allow the students to ask for any extra help required and provide the student with the opportunity for one-to-one assistance from the supervisor.

Module content

Philosophy of Data Structures. Abstract Data Types and Data Structures. Algorithms and Programs.

Simple Data Structures, Lists, Stacks, and Queues. The Dictionary ADT. Priority Queues and Heaps.

Sorting, three O(n2) Sorting Algorithms plus others including Quicksort, Mergesort, Heapsort. An Empirical Comparison of Sorting Algorithms. External Sorting. Greedy approach, Divide and Conquer approach.

Searching, Binary Search, Hashing, Binary Search Trees, other Search Trees.

Data Compression, Huffman Trees and Huffman Coding.

Graph Algorithms, Simple, Euler and Hamiltonian Paths. Graph Representation, Dense and Sparse Graphs. Graph Search. Minimum Spanning Tree. Shortest Path.

Algorithm Analysis & Complexity. A Faster Computer or a Faster Algorithm? Asymptotic Analysis. Calculating the Running Time of a Program. Space Bounds. Some Practical Considerations. P, NP and NP-Complete. Halting Problem. Turing Machines. Related theoretical topics

Module Assessment

Assessment will be based on a two hour end of semester written exam and continuous assessment during the semester.

- Written exam – 70%
- Continuous assessment – 30%

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Algorithms and Data Structures**
Programming Algorithm Design and Problem Solving		5	CMPU2001	

The continuous assessment may consist of laboratory tests, practical assignments or a combination of these as deemed fit by the module lecturer.

Essential Reading

Course notes

Goodrich and Tamassia, Algorithm Design, Wiley

Supplemental Reading

Robert Sedgewick, Algorithms in C++, Addison-Wesley

Web references, journals and other

As specified by the lecturer

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Data Communications**
		5	CMPU2005	

8.2.2 Data Communications

Module Author

Damian Bourke

Module Description

This module is intended to be a precursor to network related modules to be undertaken in later stages. In particular it is a precursor for the Client-server Programming module to be undertaken in stage 3. Consequently it is essential that the student has a good understanding of how computing devices communicate with each other. This module explains data communications at a variety of levels corresponding to the lower layers of the ISO OSI and TCP/IP reference models namely, the Physical, Data Link (host-to-network) and Network layers.

Module Aims

The aim of this module is to provide the student with a fundamental understanding of lower-level computer communication concepts.

Learning Outcomes

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

1. Describe the physical properties and evaluate the performance characteristics of a variety of communication media
2. Explain bandwidth concepts and describe its impact on data communications
3. Explain the problems affecting effective computer communication and explain how they are addressed
4. Describe and evaluate digital and analogue data encoding techniques
5. Describe switching and multiplexing principles and differentiate between different technologies
6. Describe and evaluate the predominant LAN and WAN technologies
7. Describe network extendibility and interconnection technologies
8. Describe the components associated with internetworking

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Data Communications**
		5	CMPU2005	

architectures

9. Explain hardware and protocol addressing concepts and describe their operation
10. Describe the functionality associated with the Physical, Data Link (host-to-network) and Network layers of the ISO OSI model

Learning and Teaching Methods

Learning will be achieved through lectures and tutorials which will draw upon current theory content and case studies. In addition the student will be involved in a variety of class-room exercises to further extend their appreciation of the problems affecting effective communications.

Module content

Transmission Concepts

- The communications model
- Signal representation and analysis
- Bandwidth concepts and its impact on data rates
- Transmission systems

Transmission Media

- Characteristics of guided and unguided transmission media
- Transmission impairments and how they are addressed
- Channel Capacity

Synchronous/Asynchronous Transmission

- Timing problems
- Framing

Data Link Tasks

- Flow Control techniques
- Error Detection and Error Control techniques
- Predominant data link protocols

Communication Systems architecture

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Data Communications**
		5	CMPU2005	

- Multiplexing concepts
- Communication system building blocks
- System interface concepts and standards

Switching Networks

- Circuit and Packet switching concepts, operation and routing

LAN and WAN

- Concepts, topologies and operation
- Protocol architecture
- Medium Access Control techniques
- Extending and interconnecting LANs
- Addressing

Internetworking

- Universal interconnection
- Architecture components
- Hardware and Protocol addressing
- Data delivery concepts
- The ISO OSI model

Module Assessment

The student will be assessed on their knowledge attainment through class-room assessment. In addition the student will be assessed at the end of the module using a written paper.

The end-of-semester written examination accounts for 70% of the overall module marks.

The continuous assessment accounts for 30% of the overall module marks.

Essential Reading

Data and Computer Communications, 9th Edition. William Stallings. Prentice Hall, ISBN-10: 0131392050, 2010

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Data Communications**
		5	CMPU2005	

Supplemental Reading

Computer Networks, 4th Edition, Andrew Tanenbaum. Prentice Hall ISBN: 0130661023, 2002

Data Communications and Networking, 4th Edition, Behrouz A. Forouzan, McGraw-Hill, ISBN: 0072967757, 2007

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Databases 1**
		5	CMPU2007	

8.2.3 Databases 1

Module Author

Databases Group

Module Description

The student will be introduced to the concepts and rationale behind the relational database. Data modelling and database design will be explored and the student will learn how to create and manipulate data in a relational database, ensuring data and relational data integrity. The student will apply the concepts to a case study.

Module Aims

The aims of this module are to:

- Introduce the student to the concepts, justification of and rationale behind databases.
- Give the student the modelling skills to design and implement a robust data model.
- Equip the student with the ability to define and manipulate data in a relational database.

Learning Outcomes

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

1. Describe and justify the rationale behind the use of relational database management systems.
2. Demonstrate an understanding of the desirable features of a database management system and how they are achieved.
3. Identify and distinguish between data and meta-data, and the concepts of keys.
4. Design a data model suited to a business application and implement it in a relational database.
5. Define tables and views with appropriate constraints to ensure data integrity and relational integrity.
6. Manipulate the data in a relational database using DDL and DML

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Databases 1**
		5	CMPU2007	

aspects of SQL.

7. Apply knowledge of SQL to real-world query problems.

Learning and Teaching Methods

This module will be delivered over four hours per week for one Semester. This will include two hours laboratory session. During the lecture, new material on the theories and practices in relational databases will be disseminated. Techniques will be applied in the practical laboratory sessions, including the use of a case study. Students will work individually or in teams, as appropriate.

Module content

Concepts and rationale

- Discuss the content desirable features and rationale behind a database.
- Determine the data requirements for an application, judging whether given material is information, data or meta-data.

Data modelling

- Interpret, design and create Entity Relationship diagrams.
- Explain the relationship between functional dependencies and keys and give examples.
- Define entity integrity and referential integrity and give examples of user defined integrity constraints.
- Select appropriate business rules for a given scenario and apply them to the model.
- Describe the relationship between a logical model and a physical model and create both.
- Use a CASE tool to generate and reverse engineer logical and physical models.

SQL

- Create, Alter and drop single and related tables and other objects, using appropriate data types and constraints.
- Populate tables and modify and remove rows from tables, being cognizant of constraint issues.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Databases 1**
		5	CMPU2007	

- Formulate and test queries to return selection and projection, using functions to enhance returned data.
- Use relational operators and INNER and OUTER JOINs to query multiple tables.
- Use aggregation and sub-queries to return multi-dimensional data.
- Formulate and test simple transactions to update persistent data in the database, using COMMIT and ROLLBACK as appropriate.

Module Assessment

Module will be assessed by both non-exam and examination assessment. Students must pass the examination component. Students must pass the entire module. Practical work 50%, written examination 50%.

Essential Reading

Elmasri & Navathe, 2010, ‘Fundamentals of Database Systems’ 6th Ed., Addison Wesley.

Supplemental Reading

Fehily, C. 2008 ‘SQL: Visual Quick Start Guide’ 3rd Ed., Peachpit Press

Watson & Ramklass, 2008, ‘SQL Fundamentals 1 Exam Guide’, Oracle Press

Web references, journals and other

Some references will vary, depending on the database being used.

www.oracle.com

www.sqlzoo.net

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Human Computer Interaction**
		5	CMPU2008	

8.2.4 Human Computer Interaction

Module Author

Andrea Curley and Dave Carroll

Module Description

Human Computer Interaction (HCI) emphasises the significance of good interfaces and the relationship of interface design to successful human interaction with computer systems. This module provides the theoretical knowledge and practical experiences in the fundamental aspects of analysing, designing, prototyping, and evaluating user interfaces based on sound HCI principles.

Module Aims

The aims of this module are to:

- Enable students to understand human behaviour with interactive objects;
- Provide students with the skills to develop, and evaluate interactive software using a human-centred approach;
- Provide students with an appreciation of human diversity and the need for accessibility and its implications for HCI design.

Learning Outcomes

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

1. Discuss the background of HCI with its underpinnings from psychology and cognitive science;
2. Select appropriate I/O devices and interaction styles to support users and their tasks;
3. Employ a range of user centred design techniques which use established design principles and methodologies to solve HCI problems;
4. Employ a range of low-fidelity and high-fidelity prototyping techniques;
5. Employ a range of evaluation techniques;

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Human Computer Interaction**
		5	CMPU2008	

6. Discuss accessibility issues and their implications for user interface design.

Learning and Teaching Methods

This module will be delivered using a combination of lectures and practical laboratory sessions. The main theoretical material will be delivered in lectures. The practical work will analyse design, prototype and evaluate a user interface will be carried out in the laboratory sessions.

Module content

Fundamentals of Human Computer Interaction: Scope and interdisciplinary nature of HCI; human capabilities; user groups.

Cognitive Foundations Metaphors: mental models; perception; attention; memory; learning.

User interface technology: Input and output devices, interaction styles.

User-centred design process: HCI design process vs. software engineering design process; usability guidelines, principles and theories

User interface design and prototyping: User centered design; participatory design; screen design; low-fidelity, medium-fidelity and high-fidelity prototyping design principles and rules.

User interface evaluation Usability specifications. Measurement criteria. Usability evaluation; cognitive walkthroughs, heuristic analysis, expert review, think aloud methods.

Usability issues and accessibility: Ease of use. Principles of universal design. Usability standards. HCI standards. Accessibility requirements. Implications for HCI. Legal imperative for accessibility.

Major Application Areas: Safe vs. hostile user environments: end-user interfaces, computer supported co-operative work, e-commerce, mobile computing, ubiquitous computing.

Module Assessment

60% - written examination

40% - continuous assessment

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Human Computer Interaction**
		5	CMPU2008	

Essential Reading

Preece, J. et al (2002), Interaction Design, John Wiley & Sons.

Norman, D. (2002,) The Design of Everyday Things, Basic Books.

Supplemental Reading

Benyon, D. (2010), Designing Interactive Systems, Pearson.

Shneiderman, B. & Plaisant, C. (2010), Designing the User Interface, Pearson.

Dix, A., Finlay,J., Abowd, G., & Beale, R. (2003), Human Computer Interaction, 3rd Ed, Prentice Hall.

Web references, journals and other

Jakob Nielsen's Website http://www.useit.com/

Donald Norman's Website http://www.jnd.org/

Ben Shneiderman's Website http://www.cs.umd.edu/~ben/

ACM SIGCHI http://www.sigchi.org/

BCS HCI http://www.bcs.org/server.php?show=nav.14296

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Legal and Professional Issues**
		5	CMPU2011	

8.2.5 Legal and Professional Issues

Module Author

Paul Bourke

Module Description

This module describes the relationship between technological change, society and the law, emphasising the powerful role that computers and computer professionals play within the IT industry. The learner will be introduced to the module content from the perspective of a highly placed computer professional or manager involved in the strategic development of an organisation. In particular it considers the legal areas relevant to the discipline of computer science namely intellectual property, liability for defective software, computer misuse, etc. The learner will also be introduced to ethical concepts that are important for an IT professional.

In preparation for the Work Placement module in 3rd year, this module will introduce students to the factors involved in career decision making as well as giving them the practitioner experience necessary for seeking out employment.

Module Aims

The aims of this module are:

- To introduce the law regulating the use of computers.
- To encourage awareness and critical thinking regarding the social impact of information and communication technology (ICT).
- To identify the impact of national, international and EU regulation and other legislative processes in defining IT strategy.
- To become familiar with Codes of Conduct, Professionalism and Ethical theories.
- To give students the necessary skills to enable them to begin the process of taking control of their career development.
- To understand the business environment of the IS professional

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Legal and Professional Issues**
		5	CMPU2011	

Learning Outcomes

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

1. demonstrate a broad knowledge and integrated understanding of the structure, sources and substantive laws and regulation governing the ICT environment.
2. examine the ongoing development of law and regulation and its impact in the strategic development of IT within an organisation.
3. apply the substantive law and regulation to a range of complex practical problems in ICT working environment using real case studies.
4. analyse legal problems, classifying issues in terms of relevance and importance, and propose solutions.
5. analyse the ethical issues that may arise for a IT Professional within the work environment and in relation to the implementation of an ICT solution.
6. decide what actions they must take in order to pursue their chosen profession
7. construct a networking plan/target list and outline the broad approaches to job search and getting the job they want
8. apply a project and business management skill-set.

Learning and Teaching Methods

Lectures, self-study, PBL tutorials, and any combination of discussion, case study, problem-solving exercises, readings, seminars, and computer-based learning.

Module content

Sources of Irish Law: Bunreacht na hEireann - separation of powers, fundamental rights, organs of state. Legislation –superior and delegated. Common Law – doctrine of precedent. E.U. Law – Community law and national law, Directives, Regulation, actions in the Court of Justice.

Law of Tort: Nature of a tort. Liability issues. Defences, Remedies: damages, injunction.

Employment Law: Contracts of employment, legislation on dismissal,

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Legal and Professional Issues**
		5	CMPU2011	

equality, working time and safety.

Information Technology Law: Ownership issues – data protection, copyright protection of computer programs and databases, patents. Data Protection Act 1988, Data protection Directive, the Europol Convention. Copyright Act 1963, 1993 Regulations, 1996 Pirated Goods Regulation, Database Directive 1996, Patents Act 1992 and European Patent Convention. Liability issues – contractual and tortious. Liability for defective products and services, breaches of security, breach of privacy, professional negligence.

IT Criminal Issues: computer misuse, hacking, viruses, fraud, Criminal Damage Act 1991. Control of content, pornography, defamation, privacy and electronic commerce.

Contract Law: Defining a contract, formation of a contract, written/oral. Essential elements – capacity, agreement, consent, consideration, intention, legality. Remedies for breach of contract – damages, rescission, injunction. Sale of Goods Act 1982, EC Directive on Unfair Terms.

Ethics: This course also explores issues on the interface between information technology and society, with a special focus on ethical issues.

Professionalism and codes of practice: identify the roles and responsibilities of IT personnel towards their employers, clients and society at large. Likewise, an examination of compliance requirements for professionals e.g. under the Sarbanes-Oxley Act.

Project planning and management. Quality assurance measures for software and data

Appreciation of business practices and systems. Appreciation of risk analysis and techniques for security. Systems and programming documentation standards

Career Development: Self Assessment, Occupational Research, Opportunity Search, Applications, Interviews and Re-Evaluation.

Business continuity Management & Evolving technologies.

Module Assessment

Continuous Assessment: 50%

Written Examination: 50%

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Legal and Professional Issues**
		5	CMPU2011	

Essential Reading

Frank Bott, 2005, Professional Issues in Information Technology, BCS.

Brian Doolan, 2003, Principles of Irish Law 6th Ed., Gill and Macmillan

Jon Fell (Editor), John Antell, Jonathan Exell, Vivian Picton, Adrian Roberts-Walsh, Louise Townsend, 2007, IT Law, BCS

Jeremy Holt, Jeremy Newton, 2011, A Manager's Guide to IT Law, BCS

Deborah G. Johnson, 1994, Computer Ethics, Prentice Hall.

Supplemental Reading

Kellegher & Murray, 2008, Information Technology Law in Ireland, 2008, Tottel

Web references, journals and other

Computers and Law Journal

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mathematics 2**
Mathematics 1		10	CMPU2012	

8.2.6 Mathematics 2

Module Author

Shane Mulligan

Module Description

This is a second module in mathematics, to give the student a broad range of basic mathematical skills, and a good knowledge of how they are applied in various areas of computing and computer science. Their application to various areas in computing such as data security, computer modelling and artificial intelligence will be demonstrated.

Module Aims

The aim of this module is to give the student the necessary knowledge and competence to deal with mathematical concepts and problems that arise in computer science.

An emphasis will be given to relating the mathematical concepts to their particular application areas.

It will give the student an understanding of discrete mathematics, and demonstrate the applicability of the various branches of discrete mathematics to computing. It will present mathematics as an exact science, and train the student to think logically, and express themselves clearly.

Learning Outcomes

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

1. Perform number theory calculations, and relate them to data security techniques.
2. Demonstrate the application of set theory to formal specifications.
3. Use predicate logic to represent real-life situations, e.g. represent sentences, and draw logical conclusions.
4. Define functions and demonstrate their applications in computing.
5. Calculate probabilities of events, end expectations of random variables. Apply the tools of probability to solve problems in

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mathematics 2**
Mathematics 1		10	CMPU2012	

computing, e.g. analysis of algorithms.

6. Perform computations in 2d and 3d geometry, using vector and matrix algebra.
7. Define graphs, their properties and their various applications in computing.

Learning and Teaching Methods

The learning and teaching methods will consist of lectures and tutorials. Exercises and continuous assessments will be given to ensure that the student understands and masters the material, and to give them practice at representing and solving simple problems. Computer based methods may also be used for assessment.

Module content

More Number Theory: Modular arithmetic, discrete logarithms, application to encryption.

Predicate logic. Predicates, quantifiers, use in data representation and deductive reasoning.

Set Theory: Formal specifications using a formal language.

Functions: Hash functions, random numbers, iteration.

Probability and statistics. Probability spaces, independence and dependence, repeated trials, random variables, expectation, probability distribution.

Coordinate geometry. In 2- and 3-dimensions, matrix operations.

Graph theory. Paths, cycles, special graphs, graph algorithms.

Module Assessment

The Module assessment will be by written Examination (70%), and Continuous Assessment (30%).

Essential Reading

Kenneth H. Rosen, 2003, Discrete Mathematics and its Applications, 5th Edition, McGraw-Hill.

Seymour Lipschutz, Marc Lipson, 1997, Schaum’s outlines Discrete Mathematics 2nd Edition, McGraw-Hill.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mathematics 2**
Mathematics 1		10	CMPU2012	

Supplemental Reading

Murray R Spiegel, Larry J Stephens, 1999, Schaum's outlines Statistics 3rd Edition, McGraw-Hill.

Web references, journals and other

Lecturer's web page

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Object Oriented Programming**
Programming		10	CMPU2016	

8.2.7 Object-Oriented Programming

Module Author

Programming and Algorithms Group

Module Description

This module develops the students' programming and problem solving skills, progressing from procedural to object-oriented programming.

Topics include the principles, practices, and applications of programming in object-oriented environment with applications to business and scientifically oriented problems. The techniques and language features of object-oriented design are implemented in programming projects. Emphasis is placed upon development of well-designed, efficient, maintainable object-oriented software.

Module Aims

The aim of this module is that the student become familiar with Object Oriented Programming concepts and implement these concepts in elementary object oriented programs.

Learning Outcomes

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

1. Demonstrate an understanding of the underlying principles and concepts of Object-Oriented Programming
2. Use UML in the design of OOP programs
3. Write programs using Object-Oriented programming language
4. Extensively testing using debugging tools to remove runtime errors from a program
5. Document an Object-Oriented program
6. Demonstrate an understanding of the advanced principles and concepts of Object Oriented Programming
7. Design and implement object oriented programs using advanced Object Oriented constructs and design patterns
8. Persistent objects: Formatted file input and output, direct file input and output

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Object Oriented Programming**
Programming		10	CMPU2016	

9. Construct Program Libraries.

Learning and Teaching Methods

Lectures with demonstrations

Tutorials

Laboratory practicals based on lectures and tutorials

A Virtual Learning Environment (VLE) is used extensively in this module.

Module content

Comparison of Procedural Programming with OOP

Review of procedural programming. Problems with procedural programming and the need for OOP

Object-Oriented Principles and Concepts

Introduction to class diagrams and OOP concepts.

Relationships, Inheritance, Multiple Inheritance, Abstract Classes.

Object Oriented Programming Language Constructs

Abstract data types, classes, objects, messages, Instance variables, methods, encapsulation, private and public access, class variables, constructors, class interface, class implementation.

Classes and objects, private and public class members, constructors, initialisation list, static data members, overloading, inline, separation of interface and implementation.

Function overloading. Operator overloading.

Destructors.

Virtual functions and friend functions.

Composition.

Inheritance: Types of Inheritance, Construction, Destruction, Multiple Inheritance.

Polymorphism.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Object Oriented Programming**
Programming		10	CMPU2016	

Abstract Classes.

String class and character arrays. Pointers and dynamic memory.

Generic Types, Static and Dynamic Binding, Polymorphism, Overloading.

Exception handling.

Streams and files.

Templates: functions and classes. Genericity.

Collections frameworks.

Program Libraries.

Testing techniques for object oriented programs.

Module Assessment

Continuous Assessment (60%):

- Individual assignments
- Lab tests
- On-line tests
- In-class written tests

Written examination (40%):

- One end of module examination.

Essential Reading

Depending on the language used in this module, specific reading lists will be specified in advanced of the start of the module.

Supplemental Reading

Dependent on the language used in this module.

Further Details

Class size is expected to be 80, which can be divided into smaller groups for labs and tutorials.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Operating Systems 2**
Operating Systems 1		5	CMPU2017	

8.2.8 Operating Systems 2

Module Author

Michael Gleeson

Module Description

This module serves to extend the learner's knowledge of operating systems design and operation. It builds on knowledge gained in Operating Systems I and examines in further detail more advanced aspects of an operating system. It introduces students to the principles and practice of systems security and distributed operating systems. The concepts are re-enforced with practical laboratory exercises in operating system configuration and manipulation. Practical shell programming assignments are also given to develop advanced practical operating systems skills.

Module Aims

The aim of this module is to extend the students knowledge of the principles of operating system design and to develop in them an advanced working knowledge of operating systems. It also servers to introduce to the student to systems security, distributed operating systems and new emerging and developing technologies.

Learning Outcomes

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

1. describe in detail the functions of the major components of an operating system
2. describe in detail the interactions between these components
3. describe in detail different types of operating system
4. discuss what the kernel is and differentiate between user and kernel modes
5. control the behaviour of an operating system through a command interface
6. understand the goals of security and protection of an operating system
7. discuss encryption and authentication

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Operating Systems 2**
Operating Systems 1		5	CMPU2017	

8. synthesize UNIX and Linux user, group and systems permissions in a lab environment
9. compare and contrast networked and distributed operating systems
10. appraise and integrate developing technologies such as hypervisors

Learning and Teaching Methods

In this module a number of teaching methods will be employed including lectures, practical sessions, tutorials and case studies. At least one industrial seminar may be arranged. Focus should be placed on empowering the students to develop their skills independently of the presence of a tutor or lecturer. Practical assignments will be given throughout to allow students to gain experience of operating systems shell programming.

This can be aided by the introduction and use of VLE resources, examples including online student discussion groups, reflective blogs for use immediately after practical sessions and voluntary Q&A sections.

Module content

Introduction: Definition of an operating system, functions of an operating system, event-driven systems, protection mechanisms, virtual resources, evolution of operating system designs, examples of operating systems.

Using an Operating System: The programmer’s view, event-driven programming, exceptions, the shell, graphical shells.

The Kernel: Kernel responsibilities, implementing threads, kernel objects, process management, scheduling, exception handling, memory management, kernel processes, daemons.

Process Synchronization: Critical selections, race conditions, concurrent systems, lock-based mechanisms, monitors, semaphores, classic process synchronising problems.

Deadlocks: Resource allocation, resource handling, mutual exclusion, handling deadlocks, detection and resolution, avoidance, prevention, deadlock handling in practice.

Hardware Support: Computer operation, user and kernel modes, kernel

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Operating Systems 2**
Operating Systems 1		5	CMPU2017	

memory, system calls, memory management hardware, caching, power management, hardware failure.

File Systems: I/O devices, device level I/O, access methods, unified disk cache, disk scheduling, disk block buffering.

Systems Security: Overview of security, security attacks, encryption, authentication, access control and protection structures, UNIX/Linux and windows security mechanisms.

Distributed Operating Systems: Distributed operating systems, networked operating system, model of a distributed system, design issues in distributed computing, distributed file systems.

Developing Technologies: Virtualisation and hypervisors, native and host based systems, Virtual Box, VMWare and Xen technologies.

Laboratory Work: In addition to the lecture material studied in class, a weekly lab session focusing on Linux and UNIX-like operating systems will be scheduled. This session will be a hands-on approach to understanding and utilize advanced features of Linux and UNIX-like operating systems. Topics covered include advanced shell commands, manipulating permissions and file systems, designing security configurations, differences with distributed systems and advanced shell scripting. Implementing some form of virtualisation technologies to serve as a hand on introduction to the area will also be addressed.

Module Assessment

This module should have a 60% weighting for the examination and a 40% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, it is equally important that they can demonstrate an understanding of the theoretical aspects of Operating Systems.

Essential Reading

Ida M. Flynn & Anne McIver McHoes, 2008, Understanding Operating Systems (5th Edition), Thompson Leaning.

John English, 2005, Introduction to Operating Systems: Behind the Desktop, Palgrave McMillian.

D.M. Dhamdhere, 2007, Operating Systems: A Concept based Approach, McGraw Hill.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Operating Systems 2**
Operating Systems 1		5	CMPU2017	

Supplemental Reading

William Stallings, 2009, Operating Systems: Internals and Design Principles, Prentice Hall.

Web references, journals and other

http://williamstallings.com/OS/OS6e.html

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Engineering 1**
		5	CMPU2019	

8.2.9 Software Engineering 1

Module Author

June Barrett, Oliver O'Connor, Andrea Curley

Module Description

This module develops the student theoretical and practical understanding of the software development lifecycle as an engineering discipline. It gives the student an overview of the development of a project from its inception right through to its realisation. Using a relatively simple case study, the student will analyse and specify requirements and design a partial solution.

Module Aims

The aim of this module is to introduce student to the basic principles of software engineering. Specifically, it aims to provide the student with the skills and knowledge to identify and employ appropriate practices for the specification of an object oriented system . Also, to enhance the students team working skills through the use of a group Case Study.

Learning Outcomes

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

1. Demonstrate an appreciation for the discipline of software engineering and the role of a software engineer.
2. Gather and specify system requirements, from a description of a system and problems inherent in it.
3. Understand the benefits derived from the use of Computer Aided Systems Engineering (CASE) tool.
4. Create UML diagrams to represent a system
5. Use appropriate UML diagrams to aid in the requirements elicitation process.
6. To create a complete user requirements specification model

Learning and Teaching Methods

Lectures with tutorials and laboratory sessions to give the student experience in developing models and specifications of a variety of

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Engineering 1**
		5	CMPU2019	

systems with lecturer feedback. The use of a CASE tool will provide standards, rigour and guidance for the specifications. The Object Oriented concepts will be reinforced by integrating applications with the programming module. Extensive use of a Virtual Learning Environment (VLE)

Module content

Introduction to Software Engineering Software Engineering as an engineering discipline; Software processes.

Requirements elicitation, questionnaires, interviews, observation and prototyping.

Object oriented principles and concepts.

Object Oriented Development Rational Unified Process; Principles and concepts;

Visual Modelling (use-case diagrams, class diagrams and sequence diagrams.

Requirements specification

Module Assessment

This module will be based on 30% continuous assessment and 70% written examination

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one assignment which will consist of modelling and enhancing a system using a CASE tool in a laboratory environment

Essential Reading

Software Engineering", Ian. Sommerville, 9th ed, Pearson, ISBN 0-13-705346-0

"Object-Oriented Systems Development", Carol Britton & Jill Doake, McGraw Hill, ISBN 0-077-09544-8

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Engineering 2**
Software Engineering 1		5	CMPU2020	

8.2.10 Software Engineering 2

Module Author

June Barrett, Oliver O'Connor, Andrea Curley

Module Description

Building on the analysis skills from software Engineering I, the module studies elaboration and design stages of a software engineering project. It introduces testing, and forward engineering of the specification to create class templates and a data model.

Module Aims

The aim of this module is to expand the software engineering techniques from software Engineering I module, and apply them to developing a design specification for a small system.

Learning Outcomes

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

1. Produce a detailed set of Object Oriented Design specifications
2. Design Data Management Objects
3. Implement a test process and develop test cases
4. Identify and apply simple Design patterns in software development
5. Relate and apply Design Models for object system development
6. Evaluate and use the various tools available to the software engineer

Learning and Teaching Methods

Lectures with tutorials and laboratory sessions to give the student experience in developing models and specifications of a variety of systems with lecturer feedback. The use of a CASE tool will provide standards, rigour and guidance for the specifications. The Object Oriented concepts will be reinforced by integrating applications with the programming module. Extensive use of a Virtual Learning Environment (VLE)

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Engineering 2**
Software Engineering 1		5	CMPU2020	

Module content

Refining the Requirements model

Class Diagram with attributes, operations and their specification

Object interaction and collaboration

Representing software behaviour, Sequence Diagram with message modelling, Collaboration Diagram

Detailed Design , associations, boundary and control classes

Software design principles, decomposition, decoupling, cohesion, reuse, and portability

Design patterns

Testing and project management

Tiered architectures

Module Assessment

This module will be based on 30% continuous assessment and 70% written examination

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one assignment which will consist of modelling and enhancing a system using a CASE tool in a laboratory environment

Essential Reading

Software Engineering", Ian. Sommerville, 9th ed, Pearson, ISBN 0-13-705346-0

"Object-Oriented Systems Development", Carol Britton & Jill Doake, McGraw Hill, ISBN 0-077-09544-8

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Development 2**
Web Development 1		5	CMPU2022	

8.2.11 Web Development 2

Module Author

Cindy Liu

Module Description

To enhance the web skills developed in Web Development I. To further develop skills needed to implement more advanced Web applications that use server-side technologies.

In addition, the foundation on the Internet architecture treated in the Web Development I module is extended by examining the core protocols of the Internet within the TCP/IP model.

Module Aims

The aims of this module are:

- To expand the student's knowledge of the infrastructure of the Internet by examining the TCP/IP protocols and their relationship to application layer protocols such as HTTP.
- To introduce the student to the HTTP protocol as a simple but scalable, text based application layer protocol, and to examine the evolution and capabilities of the protocol.
- To provide the student with the server side skills to integrate their existing skills with client side technologies and databases to develop three tier web applications.
- To examine important issues for web applications, such as scalability, security and extensibility.
- To examine the role of the web server and develop the skills required to install and administer web servers upon which web applications are deployed.

Learning Outcomes

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

1. Describe the underlying architecture of both the World-Wide-Web and related Internet technologies.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Development 2**
Web Development 1		5	CMPU2022	

2. Implement web pages and generate web content dynamically and interact with users of web resources.
3. Maintain state across a number of web resources using cookies.
4. Evaluate and be familiar with the newest advances in Web technology.
5. Express an understanding of the strengths and weaknesses of competing technologies and select the appropriate technology or technologies for a given task.

Learning and Teaching Methods

Learning will be encouraged and supported through a combination of strategies such as lectures, lab work, projects, presentations and service-learning.

As with its prerequisite Web Development I module, it is important that students taking this module learn by doing. Sufficient time should be facilitated in the laboratory to give students the opportunity to experiment with the HTTP protocol, web servers and server side technologies. It is also important that the underlying theory of the World-Wide-Web is treated during the lecture time, although it is felt that this should be supported by structured experiments in the lab, where students examine the role of the components and protocols of a web environment.

Module content

Internet Infrastructure and Protocols: TCP/IP and The HTTP Protocol

Tiered Architectures : n-Tier architectures. Client, server, database. Roles of layers. Interaction between layers. Client server architectures. Management of database. Role of application servers.

Web Servers: Role of the web server. Different web server implementations. Basic operation of web server in implementing HTTP protocol. Authentication on web servers. Security on web servers. Installation of web servers. Configuration of ports.

Web Application Development: Technologies available. Common Gateway Interface. Perl. PHP, JSP, ASP, ColdFusion. Detailed examination of one technology.

Server Issues: Security. Authentication. Passwords. Scalability. Testing.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Development 2**
Web Development 1		5	CMPU2022	

Comparison of web server functions and operation.

Module Assessment

50% - written examination

50% - continuous assessment.

Essential Reading

Simon Brown. (2005), Pro JSP 2, Fourth Edition

Ben Laurie, Peter Laurie (2009), Apache: The Definitive Guide. Third Ed. O'Reilly

Supplemental Reading

Leon Shklar, Richard Rosen (2009), Web Application Architecture: Principles, Protocols and Practices, John Wiley and Sons Ltd

Web references, journals and other

World-Wide-Web Consortium -http://www.w3.org/

JSP reference : http://www.jsptut.com/

The Apache Software Foundation http://apache.org/

8.3 Stage 3 Syllabi

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Applied Intelligence**
		5	CMPU3004	

8.3.1 Applied Intelligence

Module Author

Brian MacNamee and John Kelleher

Module Description

This module introduces sudents to the power of artifical intelligence (AI) and machine learning (ML) by developing and exploring a series of ML and AI based applications primarily focused on using online data sources. While the module will cover AI and ML echniques these will not be covered in great depth, but rather the module will focus on the application of these techniques.

Module Aims

The aims of this module are to:

- Introduce students key techniques from artificial intelligence and machine learning
- Develop in students a deep understanding of how AI and ML techniques can be applied to real world problems.
- Engage sudents in the development of a range of real solutions using ML and AI techniques.

Learning Outcomes

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

1. Demonstrate an understanding of a range of machine learning and artificial intelligence techniques
2. Select appropriate ML and AI techniques to use to solve specific problems
3. Compare the suitability of different AI and ML approaches to solving the same problem
4. Develop a range of AI and ML solutions to real problems
5. Critique the limitations of AI and ML approaches

Learning and Teaching Methods

Lectures. tutorials, lab demonstrations and discussions. This module

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Applied Intelligence**
		5	CMPU3004	

will be primarily practical and while material will be presented in lectures a large amount of time will be spent developing solutions to real problems based on the material covered in lectures. A virtual Learning Environment is extensively used in this module.

Module content

- Introduction to Applied Intelligence
  - What Is Artificial Intelligence (AI)?
  - What Is Machine Learning (ML)?
  - Limits of ML and AI
  - Applications of AI and ML
- Making Recommendations
  - Collaborative Filtering
  - Collecting Preferences
  - Finding Similar Users
  - Recommending Items
  - User-Based or Item-Based Filtering?
  - Applications of Recommender Systems: Online Retail Recommendation System, Social Network Friend Recommendation
- Discovering Groups
  - Supervised versus Unsupervised Learning
  - Hierarchical Clustering
  - K-Means Clustering
  - Clusters of Preferences
  - Applications of Clustering: Customer Segmentation, Player Segmentation
- Searching and Ranking
  - What's in a Search Engine?
  - A Simple Crawler

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Applied Intelligence**
		5	CMPU3004	

  - Building the Index
  - Querying
  - Content-Based Ranking
  - Applications of Search: Building the next Google!
- Optimization
  - Representing Solutions
  - The Cost Function
  - Random Searching
  - Hill Climbing
  - Simulated Annealing
  - Genetic Algorithms
  - Optimizing for Preferences
  - Network Visualization
  - Applications of Optimization: Price Modelling, Parameter Selection
- Building Classification Models
  - Documents and Words
  - Modelling with Decision Trees
  - Modelling with Naïve Bayes Classifier
  - Modelling with k-Nearest Neighbours Classifier
  - Applications of Classification Models: Filtering Spam, Filtering Blog Feeds, Building Price Models, Social Network Friend Matching
- Evolving Intelligence
  - What Is Genetic Programming?
  - Genetic Programming Approaches
  - Applications of Genetic Programming: Games, Opimisation, Search
- APIs and Languages

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Applied Intelligence**
		5	CMPU3004	

- Suitable APIs (e.g. Google API, Facebook API) an programming languages (e.g. Python) and techniques will be used to develop the sample applications described above.

Module Assessment

Continuous assessment - 100%

The module will be evaluated based entirely on continuous assessment. Students will be asked to develop a range of solutions to real problems using the approaches discussed in lectures. Students will also be tasked with presenting their opinions of the state of the art of AI and ML applications and he limitations of ML and AI based approaches in a series of in-class presentations.

Essential Reading

Toby Segaran, 2007, Programming Collective Intelligence: Building Smart Web 2.0 Applications, O'Reilly Media.

Supplemental Reading

Russell & Norvig, 2009, Artificial Intelligence – A Modern Approach (3rd ed), Prentice Hall

Ethem Alpaydin, 2004, Introduction to Machine Learning, MIT Press

Tom Mitchell, 1997, Machine Learning, McGraw Hill

Web references, journals and other

Machine Learning, Springer Science + Business Media

ACM Journal of Machine Learning Research, ACM

IEEE Transactions on Pattern Analysis and Machine Intelligence

IEEE Intelligent Systems

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Business and Enterprise**
		5	CMPU3005	

8.3.2 Business and Enterprise

Module Author

Hugh McAtamney

Module Description

In this module the students will study the business environment, the nature of enterprise, the generation of potential business idea and how to package this knowledge in a professional formal, realistic and professional business plan. It also introduces students to the concepts of thinking laterally, creating new ideas and how to develop business ideas for starting up a new business.

Module Aims

The aims of this module are to:

- Provide Idea Generation Tools
- Provide an introduction to the business environment and particularly the role of the small to medium sized business,
- Help students generate potentially commercial ideas from their computer studies, to evaluate this idea and package it as a product, process, service or concept in the form of a formal professional business plan.
- Present this plan as a project in seeking finance

Learning Outcomes

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

1. Assess project viability and determining what constitutes a ‘good’ idea
2. Describe the increased importance of enterprise and innovation for the Irish economy
3. Attempt to identify potential commercial opportunities from their knowledge base, research skills and course
4. Understand the nature and role of the entrepreneur in the business set up and corporate intrapreneurship

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Business and Enterprise**
		5	CMPU3005	

5. Research a market and to measure its business potential
6. Critically evaluate the business set up planning process
7. Develop and produce a comprehensive business plan
8. Identify and source the financial resources to set up a business
9. Present a critical evaluation of good business practice

Learning and Teaching Methods

Lectures, case studies, workshops, guest lectures.

Module content

Lateral thinking & thinking outside the box, Applied idea generation techniques , Morphological analysis,

Random Association, Thinking Backwards

The growing importance of Small to Medium sized Business (SMEs).

The Irish and European business economic structure.

The Barriers to innovation in SMEs and their potential for growth.

Characteristics of Entrepreneurs, corporate entrepreneurship and the entrepreneurial process.

Idea generation, evaluation and protection. (Registration, Register Design, Trade Marks Patents, copyright etc.).

Business planning process and preparation of a business plan.

Marketing research, management, production and finance planning

Business support initiatives

Strategic planning for the SME

Presentation and evaluation of the business plan

Module Assessment

Examination: 50%

Continuous Assessment: 50%

- Business Plan (teams): 70% for plan and 30% for presentation

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Business and Enterprise**
		5	CMPU3005	

Essential Reading

Starting your own Business, Brian O'Kane, Latest Ed. Oak Tree Press.

Annual Competitiveness Reports. Forfas

Global Entrepreneurship Monitor reports

Supplemental Reading

Magazines: Business and Finance, Running Your Business, The Economist

Business newspapers and current affairs.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Client-Server Programming**
		5	CMPU3006	

8.3.3 Client-Server Programming

Module Author

Damian Bourke

Module Description

Client-Server applications are of critical importance to modern enterprise systems. This module introduces the student to the concepts behind the development and deployment of such applications. Essentially it builds upon the student's understanding of data communications and network technologies and introduces the student to the higher layer functionality associated with the ISO OSI and TCP/IP reference models. It also introduces the techniques for installing and configuring applications. The student is expected to write a significant amount of code and to use their problem solving skills to develop complex client and server applications.

Module Aims

The aim of the module is extend the student's software development skills to a networked environment and to extend the student's knowledge of common network issues. The student will also gain experience working in a command-line mode for the development and roll-out of application software.

Learning Outcomes

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

1. Describe the client-server computing paradigm
2. Explain the difference between protocols and services
3. Describe the purpose of the transport layer and the services provided by this layer
4. Compare and contrast the functionality associated with the transport and network layers
5. Describe the functionality associated with common network applications.
6. Produce iterative and concurrent networked applications
7. Demonstrate the installation and configuration of networked

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Client-Server Programming**
		5	CMPU3006	

applications

8. Demonstrate and evaluate the operation of the transport layer using networked applications
9. Demonstrate the use of standard network monitoring tools
10. Demonstrate and explain the use of standard UNIX tools to configure the starting and stopping of services in different UNIX run-levels

Learning and Teaching Methods

Learning will be achieved through a combination of lectures and laboratory sessions. Lectures will draw upon current theory content and case studies. Weekly laboratory sessions will be used to reinforce lectures and will facilitate the student's self-directed learning.

In addition there will be at least two assignments covering theoretical and practical content. These will facilitate the student's self-assessment.

Module content

Introduction

- The Client-server model
- Concepts
- Concurrent server-class machines/programs

The Transport Layer

- Services and protocols
- Transport addresses
- Transport connections
- Transport layer operational concepts

The Application Layer

- Address resolution
- Common networked applications

The Berkeley Sockets API

- The Socket Abstraction

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Client-Server Programming**
		5	CMPU3006	

- The Socket interface/primitives
- Addressing
- Remote and local process-to-process and inter-process communications
- Iterative and concurrent applications

Software Management and Deployment

- Control Scripts
- Management of Services
- Management of software installation and maintenance
- UNIX man pages – writing and maintenance

Module Assessment

The student will be assessed on their practical and theoretical attainment throughout the module through a combination of laboratory and class-room and/or written assessment. In addition the student will also be assessed at the end of the module using a written paper.

The end-of-semester written examination accounts for 60% of the overall module marks.

The continuous assessment accounts for 40% of the overall module marks.

Essential Reading

Data and Computer Communications, 9th Edition. William Stallings. Prentice Hall, ISBN-10: 0131392050, 2010

Edward C. Bailey, 1997, Maximum RPM, (Imprint: Sams) Macmillan Computer Publishing, Indiana. ISBN: 0-672-31105-4

Supplemental Reading

Computer Networks, 4th Edition, Andrew Tanenbaum. Prentice Hall ISBN: 0130661023, 2002

TCP/IP Network Administration 3rd edition. Craig Hunt, O'Reilly Publishers 2002.

Internetworking With TCP/IP Volume 1: Principles, Protocols and

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Client-Server Programming**
		5	CMPU3006	

Architecture, 5th edition, Douglas E. Comer, Prentice Hall, ISBN 0-13-187671-6, 2006

Computer Networks and Internets, 5th Edition, Douglas E. Comer, Prentice Hall, ISBN 0-13-606127-3, 2009

Hands-on Networking with Internet Applications, 2nd Edition, Douglas E. Comer, Prentice Hall, ISBN 0-13-144310-0, 2004.

Web references, journals and other

Multimedia Communications : Applications, Networks, Protocols and Standards, Fred Halsall, Addison Wesley, UK, 2000, 0-201-39818-4.

Donnie Barnes, 1999, The RPM How-To Guide (http://www.rpm.org/RPM-HOWTO/), Red Hat Inc. Accessed 7th March 2006

Article: What You Need to Write Man Pages. By Peter Seebach

http://www.linux.com/article.pl?sid=04/02/05/1651203 (Accessed: 12th March)

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Cloud Computing**
		5	CMPU3007	

8.3.4 Cloud Computing

Module Author

Paul Doyle

Module Description

This module looks at the history and evolution behind cloud computing followed by a review of the latest technologies within it. This module is designed to provide the student with both a practical and theoretical understanding of existing cloud systems and their underlying technologies. The technologies focused on will include grid computing, virtualisation, distributed computing, cloud storage, security within the cloud in addition to reviewing a number of existing cloud environments.

Module Aims

The aim of this module is to provide the learner with both a theoretical and practical understanding of the technologies used to enable Cloud Computing.

Learning Outcomes

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

1. Demonstrate an understanding of the fundamentals of Cloud Computing
2. Demonstrate an understanding of the evolution of Cloud Computing technologies
3. Demonstrate a practical understanding of cloud technologies within a laboratory environment.
4. Configure basic infrastructural components used within the cloud
5. Critically analyse different methods for implementing Cloud solutions

Learning and Teaching Methods

The course delivery involves a combination of lectures and labs. Students will be expected to put into practice some of the module concepts by interacting with existing public cloud infrastructures.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Cloud Computing**
		5	CMPU3007	

Module content

What is cloud computing?

Classifications of Clouds

Evolution of Clouds

Technologies within the cloud

Virtualisation

Storage and Security

Hadoop

Case Studies of Commercial Clouds (AZURE, Google APPS, Amazon EC2, OpenStack)

Module Assessment

Assessment will be through a combination of continuous assessment and a written exam.

Marks will be allocated as follows

- Written Exam (70%): The written exam will be conducted under normal DIT Exam regulations and will be based on the theory covered during lectures.
- Continuous Assessment (30%): The CA elements of the course will be based on the implementations of basic cloud technologies.

Essential Reading

Velte/Elsenpeter 2009, Cloud Computing A Practical Approach, McGrawHill

Rosenberg/Mateos 2010, The Cloud at your service, Manning

Mather/Kumaraswamy/Latif Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice) 2009, O'Reilly Media

George Reese, 2009, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, O'Reilly

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Cloud Computing**
		5	CMPU3007	

Supplemental Reading

Armbrust, 2009, Above the clouds: A Berkeley view of Cloud Computing,UCB/EECS

Kesselman/Foster, 1998, The Grid: Blueprint for a new computing infrastructure, Morgan Kaufmann Publishers

Web references, journals and other

http://aws.amazon.com/ec2/

http://www.ibm.com/ibm/cloud/

http://code.google.com/appengine/

http://www.microsoft.com/windowsazure/

http://www.grid.ie/

http://www.openstack.org/

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Computational Mathematics**
Mathematics 2		5	CMPU3008	

8.3.5 Computational Mathematics

Module Author

John Gilligan, Shane Mulligan

Module Description

This is a an advanced module in Mathematics, which will give the student the necessary mathematical skills and tools, to deal with specialised areas in computing, such as Graphics, Image processing, Games development and Cryptography.

Module Aims

The aim of this module is to give the student the necessary knowledge and competence to deal with specialised applications in computer science, such as graphics, image processing and cryptography. It will present the necessary mathematical tools and how they can be applied to various problem areas. It will present mathematics as an exact science, and a powerful tool to model and analyse problems.

Learning Outcomes

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

1. Define and use linear algebra techniques in geometry.
2. Explain the set-up and application of linear transformation, and matrix techniques in Graphics.
3. Apply coordinate geometry algorithms in 2 and 3 dimensions.
4. Develop and apply computational techniques which have application in topics such as image processing and intelligent analysis. These methods include, for example, numerical analysis and soft computing approaches and algorithms from number theory, and their application to cryptography.

Learning and Teaching Methods

This module will consist of lectures and tutorials. Exercises and continuous assessments will be given to ensure that the student understands and masters the material, and to give them practice at representing and solving simple problems.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Computational Mathematics**
Mathematics 2		5	CMPU3008	

Module content

Vectors and Matrices: Vectors, vector operations, cross product, normal vector. Matrices, multiplication, transpose, determinants.

Geometry. Co-ordinate systems, homogeneous coordinates. Line, circle, plane equations, parametric forms. Normal vector. Geometrical transformations and matrices, scaling, translation, rotations, and their composition. Projections and projection matrices.

Geometric algorithms: Euclidean distance, areas, intersection of line and plane, point-in-triangle test, area of a polygon.

Computational methods Numerical Algorithms, Methods from Number Theory, Soft Computing Methods, Applications in Computer Science

Module Assessment

Written Examination 70%

Practical 30%

Essential Reading

Foley J. D., A. van Dam, S. K. Feiner, J. F. Hughes, 1990, Computer Graphics – Principles and Practice, 2nd Edition, Addison-Wesley.

Supplemental Reading

Bishop Cryptography with Java Applets Jones and Bartlett Comp Sci 2003

Karray and De Silva Soft Computing and Intelligent Systems Design Addison Wesley 2004

Web references, journals and other

www.mathworks.com

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Databases 2**
Databases 1 Programming		5	CMPU3010	

8.3.6 Databases 2

Module Author

Databases Group

Module Description

The material covered will build on student's knowledge and skills to include design and transaction processing on a multi-user environment. The student will use a procedural language extension to allow for sequential, selective and iterative processing. Triggers, functions and procedures will enrich the student's ability to store and retrieve data securely. The student will apply the concepts to a case study.

Module Aims

The aims of this module are to:

- Enhance design skills using Data Normalisation and Extended Entity Relational Modelling.
- Expand the student's ability to ensure data integrity using further database objects, such as triggers, functions and procedures.
- Furnish the student with the skills to undertake multi-layer transactions using a procedural language extension to the relational database, and a third-generation language.

Learning Outcomes

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

1. Apply ER Modelling skills to a medium-sized database application area.
2. Normalise data to third normal form.
3. Demonstrate an understanding of ACID properties.
4. Design and implement a multi-statement transaction using a procedural extension to the relational database.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Databases 2**
Databases 1 Programming		5	CMPU3010	

5. Write and use functions, procedures and triggers where appropriate.
6. Configure user access and transactions to allow for multiple users with different needs to access the database.
7. Practice multi-layer transaction processing.

Learning and Teaching Methods

This module will be delivered over four hours per week for one Semester. This will consist of one hour lecture, one hour tutorial and one two-hour laboratory session. During the lecture, new material on modelling, transaction processing and programming with relational databases will be disseminated. Techniques will be applied in the practical laboratory sessions, including the use of a case study. Where appropriate, students will work in teams. The tutorials will review theory and practice, to prepare them for practical and examination work, in accordance with the requirements of the students.

Module content

Techniques of Data Normalisation and Extended Entity Relationship Modelling.

Architecture and layering of transactions in a relational database.

Concepts of procedural programming as applied in the language extension.

I/O to the client from the language extension and from a 3GL.

Stored Functions and procedures.

Trapping and propagating errors from the database to a client process.

Allocating privileges to schema users, depending on the user requirements.

ACID properties, concurrency and locking.

Audit and constraint triggers.

Module Assessment

Module will be assessed by both non-exam and examination

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Databases 2**
Databases 1 Programming		5	CMPU3010	

assessment. Students must pass the examination component. Students must pass the entire module. Continuous assessment 40%, examination 60%.

Essential Reading

Elmasri & Navathe, 2010, 'Fundamentals of Database Systems' 6th Ed., Addison Wesley.

Depending on database being used, examples are:

Sunderraman, R., 2007, 'Oracle 10g Programming: A Primer, Addison Wesley

Sack, J., 2008, SQL Server 2008 Transact-SQL Recipes: A Problem-Solution Approach, APress

Geschwinde, E., 2001, PostgreSQL Developer's Handbook, 2nd Ed., Sams

Supplemental Reading

Connolly & Begg, 2009, 'A practical approach to design, implementation and management' 5th ed. Addison Wesley

Web references, journals and other

References to specific vendors will depend on the DBMS being used:

www.oracle.com

www.postgresql.org

http://msdn.microsoft.com/en-us/library/bb510741(SQL.100).aspx

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Logic and Design**
		5	CMPU3015	

8.3.7 Games Logic and Design

Module Author

Hugh McAtamney

Module Description

The aim of this module is to introduce students to theory and practice of game development. This will be achieved by enabling students to develop computer game prototypes. The course will introduce students to the fundamental concepts of game development such as game design principals, game engines, computer graphics, sound and artificial intelligence.

Module Aims

This module aims to expose students to the important considerations in game logic and design by enabling students to create 2D computer games. Students will an industry standard game engine that they can enhance and reuse to build a variety of game prototypes.

Students will learn the basics of game logic and design by understanding how to create a reusable game engine; capture keyboard, mouse, and joystick input; animate graphical objects with sprites; detect collisions between sprites; play digital sound effects and music and give game characters intelligent behavior.

Learning Outcomes

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

1. Demonstrate an understanding of the history, context and ethical issues of modern computer games
2. Demonstrate an understanding of the main concepts involved in developing computer games
3. Use mathematics and algorithms to develop 2D games
4. Analyse and define the rules and logic of a game
5. Develop game prototypes integrating OOP concepts and industry standard API’s
6. Discuss game theory, sprite animation, sprite artificial intelligence

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Logic and Design**
		5	CMPU3015	

7. Program path finding and vector based collision detection algorithms

Learning and Teaching Methods

Class time is split into a series of interactive “studio classroom” based lectures and practical problem solving in labs. In lectures, students have access to a PC, with appropriate software and development kits, so that material and examples can be examined in a live environment. In addition, students will be expected to proactively and independently seek out resources on the internet and from the library to supplement their own learning.

A Virtual Learning Environment (VLE) will be employed to distribute all teaching materials and to support student interaction with both other students and academic staff. Where new material is not presented in lectures, such material will be made available through the VLE and students are expected to proactively use this resource.

Module content

Game Logic and design

- History of computer games, technology, game theory, gameplay, game balance, design documentation, ethics and societal issues in games.

Game Engines

- Introduction to the game loop and the role of object orientated concepts in game development.

Game Logic

- Mechanics, game logic, balance, progression

Graphics & Animation

- Co-ordinate systems, color, device contexts, pens, brushes, bitmaps, fonts, graphics primitives, sprites, managing sprites, animation.

Input

- Accessing keyboard, accessing mouse, interacting with joysticks and joypads,

AI

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Logic and Design**
		5	CMPU3015	

- Rule based AI, roaming, chasing and evading.

Module Assessment

The module will be assessed through a combination of a written exam and continuous assessment. The continuous assessment aspect of the course will consist of the development of a game prototype. The final grade will be made up of 50% continuous assessment and 50% examination

Essential Reading

Michael Morrison , XNA programming, Published by Sams , ISBN:0672326590; 2004

Supplemental Reading

The Game Programming Gems series published by Charles River Media

Wendy Stahler, 2004, Beginning Math and Physics for Game Programmers , New Riders

Web references, journals and other

Gamasutra – www.gamasutra.com

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Graphical User Interface Programming**
Object-Oriented Programming		5	CMPU3017	

8.3.8 Graphical User Interface Programming

Module Author

Programming and Algorithms group

Module Description

This module addresses the need among software developers to build graphic user interfaces using a variety of user interface components. It introduces the students to event driven programming and provides the skills necessary to design and develop user interface applications. In addition it extends the student's object oriented programming skills developed in the pre-requisite Object Oriented Programming modules.

Module Aims

The aims of this module are to:

- equip the student with knowledge and programming skills to build GUIs using user interface components and implement event programming techniques
- expand the student's knowledge of more advanced programming concepts e.g. inheritance, interfaces, polymorphism
- enable students to discriminate between alternative user interface programming approaches
- demonstrate good GUI design principles

Learning Outcomes

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

1. demonstrate thorough knowledge of and ability to apply event driven programming
2. implement a user interface with user interface components and event handling
3. incorporate basic graphics within an application
4. manage persistent data in tandem with appropriate user interface components

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Graphical User Interface Programming**
Object-Oriented Programming		5	CMPU3017	

5. understand concurrency issues in event driven programming

Learning and Teaching Methods

This is a practical module, largely driven by learning through doing. The lectures will be used to present concepts, discuss sample code and look at practical examples.

The key learning method is via the programming labs where students will be set programming lab exercises to implement the concepts covered in lectures. Tutorial sessions will be used to discuss problems that students had with the lab exercises, and to discuss any particular aspect of the module that the students are having difficulty with. In addition, students will be asked to undertake practical assignment(s) where they will complete programming tasks in their own time as part of the continuous assessment.

Module content

If not previously studied, an introduction to the programming language and environment to be used;

Overview of application programming interface for the relevant GUI environment;

Creating a GUI, including components, windows, layouts ;

Implementing event driven behaviour;

Implementing the GUI in typical environments used, such as web;

Use of persistent data (time allowing, linking to a database for query retrieval and data display);

Module Assessment

The assessment for this module will be 50% continuous assessment and 50% written exam. The continuous assessment will focus on the practical learning outcomes, with assessments such as practical programming assignments, use of the lab outputs as part of the assessment (at the discretion of the lecturer) and lab based exam(s).

The written exam will be used to test the student’s understanding of the concepts and their implementation, as learnt on the module. The exam will allow the students to demonstrate their knowledge by both

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Graphical User Interface Programming**
Object-Oriented Programming		5	CMPU3017	

descriptive explanations and via written coding examples.

Essential Reading

This depends upon the programming environment used and will be specified by the lecturer at the start of the module.

Supplemental Reading

This depends upon the programming environment used and will be specified by the lecturer at the start of the module.

Web references, journals and other

This depends upon the programming environment used and will be specified by the lecturer at the start of the module.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Knowledge Engineering**
		5	CMPU3024	

8.3.9 Knowledge Engineering

Module Author

Richard Lawlor

Module Description

This module will introduce students to the issues and factors involved in building modern complex knowledge-based systems. Expert systems will be considered in detail. Issues of knowledge representation and uncertainty will be explored. There is a strong practical emphasis in the module to allow students to gain experience of tools and techniques to assist in this process.

Module Aims

The aims of this module are:

- To introduce the student to the role of knowledge in an organisation and the function of knowledge based systems.
- To introduce the student to the nature of knowledge and the particular challenges involved in developing knowledge based systems.
- To provide students with an awareness of good software design principles for developing modern knowledge systems.
- To provide a technical grounding in expert system development and knowledge representation.

Learning Outcomes

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

1. Describe the components and categories of knowledge and the role of knowledge in an organisation.
2. Explain the nature and purpose of knowledge based systems.
3. Explain the stages in knowledge based systems development.
4. Program using an appropriate knowledge related language and have basic knowledge in applying a different development paradigm.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Knowledge Engineering**
		5	CMPU3024	

5. Extract knowledge from a domain expert description and represent it in a production system.
6. Describe different approaches to knowledge representation and inference.
7. Outline how uncertainty can be handed in knowledge-based system.

Learning and Teaching Methods

The module will be delivered primarily through lectures and laboratory work. The material will be developed in an informal way during lectures. It is envisaged that the students will assimilate much of the material through problem solving and exercises. Knowledge engineering is best learned hands-on, therefore it is important that students be able to link the concepts and techniques learnt in this module to real world problems through practical laboratory work.

Module content

Knowledge based systems, components of knowledge, categories of knowledge.

Expert systems, production systems, architecture of an expert system.

Expert system shell programming.

Declarative versus procedural programming. Logic programming.

Inference, forward and backward chaining, Rete algorithm.

Knowledge representation, frames, semantic networks.

Uncertainty, certainty factors, fuzzy logic.

Knowledge and data mining.

Knowledge engineering, stages in knowledge based systems development.

Module Assessment

Assessment will be based on a two hour end of semester written exam and continuous assessment during the semester. This module should have a 60% weighting for the examination and a 40% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, it is more

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Knowledge Engineering**
		5	CMPU3024	

important that they demonstrate an understanding of the theoretical aspects of knowledge systems.

Essential Reading

Negnevitsky, 2011, Artificial Intelligence: A Guide to Intelligent Systems, Addison Wesley.

Supplemental Reading

Luger, 2008, Artificial Intelligence: International Version: Structures and Strategies for Complex Problem Solving, Addison Wesley.

Web references, journals and other

As specified by lecturer

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mobile Robotics**
		5	CMPU3025	

8.3.10 Mobile Robotics

Module Author

Brian MacNamee and John Kelleher

Module Description

An introduction to mobile robotics including underlying theory and applications. The module will have a large practical component.

Module Aims

This module provides a broad introduction to the field of mobile robotics, from a software perspective. It covers the basics of mobile robot control, software architectures, navigation and localization algorithms, and surveys common application areas. The practical aspects of the course will give students experience in developing real robotic systems.

Learning Outcomes

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

1. Critique the appropriateness of using robotic solutions for various applications
2. Describe the main components of a mobile robotic system, including sensors, actuators, control architectures and power options
3. Differentiate between and compare different robotic sensor and actuator approaches
4. Compare different approaches to robot locomotion
5. Compare different approaches to robot localisation and mapping
6. Develop mobile robotic solutions to sample problems

Learning and Teaching Methods

The module will be delivered through a mixture of lectures and lab sessions with a stronger emphasis on lab work towards the latter half of the course. Lab work will take the form of a robot workshop in which students will be divided into teams to develop mobile robot solutions to appropriate problems. A virtual Learning Environment will be used in

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mobile Robotics**
		5	CMPU3025	

this module.

Module content

- Introduction
  - Key questions for mobile robots
  - Application areas
  - Key components of mobile robotic systems
- Sensors
  - Basic concepts
  - Contact sensors
  - Internal sensors
  - Range sensors
  - Beacon based sensing
  - Data fusion
  - Biological sensing
  - Visual sensors and algorithms
- Actuators
  - Electronics basics
  - Motors
  - Artificial muscles
  - Pneumatics
  - New materials
- Mobile Robot Locomotion
  - Wheeled robot topologies
  - Legged robot topologies
  - Legged robot gaits
  - Basic mobile robot kinematics
  - Representing robot positioning

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mobile Robotics**
		5	CMPU3025	

  - Forward kinematic model
  - Mobile robot manoeuvrability
- Robot Control System Architectures
  - General Control Structure for Mobile Robots
  - Perception Action Models
  - Subsumption Hierarchies
  - Layered Models
  - Deliberative Agents
- Mapping and Localisation
  - Topologic maps
  - Metric maps
  - Construction of occupancy grids
  - Localisation approaches
  - Introduction to SLAM (simultaneous localisation and mapping)
- Planning and Navigation
  - Path-planning
  - Collision Avoidance
- Advances in Robotics
- Implementing Robotic Systems
  - An appropriate robotic API (e.g. ROS, Lejos)
  - An appropriate mobile robot hardware platform (e.g. Lego Mindstorms, iRobot Roombas)

Module Assessment

Written examination - 50%

Continuous assessment - 50%

Essential Reading

Roland Siegwart and Illah R. Nourbakhsh, 2004, “Introduction to

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mobile Robotics**
		5	CMPU3025	

Autonomous Mobile Robots”, Bradford Books.

Supplemental Reading

Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun, 2005, “Principles of Robot Motion : Theory, Algorithms, and Implementations”, The MIT Press.

Gregory Dudek and Michael Jenkin, 2000, “Computational Principles of Mobile Robotics”, Cambridge University Press.

Siciliano, Bruno; Khatib, Oussama (Eds.), 2008 “Springer Handbook of Robotics”, Springer.

Web references, journals and other

Singularity Hub (www.singulaityhub.com)

NASA robotics (robotics.nasa.com)

Robots.net (www.robots.net)

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mobile Software Development**
Object-Oriented Programming		5	CMPU3026	

8.3.11 Mobile Software Development

Module Author

Programming and Algorithms Group

Module Description

This modules covers the knowledge and practical skills to design and implement basic mobile phone applications, in a specific mobile development environment. In order to set mobile software development in context, it also covers background mobile communication technologies such as GSM, GPRS, EDGE, 3G, Bluetooth, and examines leading mobile development platforms and typical mobile data applications.

Module Aims

The aim of this module is to enable the learner to develop mobile phone applications in a leading mobile phone development platform, whilst also providing an understanding of relevant mobile phone technologies in order to set the application development in context.

Learning Outcomes

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

1. Develop mobile phone applications in a leading mobile development platform;
2. Understand and apply good design guidelines for mobile application development;
3. Discuss and compare the leading mobile development platforms such as but not limited to: Android, Qt, Windows Mobile, and iOS (Apple);
4. Discuss and compare the mobile phone communication technologies such as, but not limited to: GSM, GPRS, EDGE, 3G, Bluetooth;

Learning and Teaching Methods

Classroom based lectures will be used to explore and explain concepts,

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mobile Software Development**
Object-Oriented Programming		5	CMPU3026	

technologies, environment features, code examples and any other material relevant to the course that requires in class coverage.

A critical part of the module is the practical lab sessions where students will implement development concepts covered in lectures. Practical exercises will be completed within the lab , with practical assignments completed outside of class time, in the students own time. The students will use an Integrated Development Environment ni the lab to support their programming work if appropriate for the platform selected .

A tutorial session will also be included to provide a discussion forum where students can explain the problems that have had in lab sessions or with concepts covered in lectures. The tutorial session will also be used to introduce new material as needed to supplement material covered in lecture time.

Module content

Technologies for mobile software development have and continue to develop and change rapidly. Therefore, no specific mobile platform is prescribed. The content is described generically, and will be covered for the particular platform used when the module is taught:

Mobile phone development: typical applications, technologies, usage;

Mobile communications technologies: local (e.g. Bluetooth) and network technologies (e.g. GSM, 3G);

Use of the relevant IDE for application development;

Mobile GUI components such as widgets, containers, menus;

Mobile GUI Layouts;

Mobile communications via the Internet;

Using persistent data with mobile applications;

Design consideration for mobile application development.

Module Assessment

This module will be based on 50% continuous assessment and 50% written examination. The continuous assessment will be based upon practical assignments that will reinforce and assess the practical

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Mobile Software Development**
Object-Oriented Programming		5	CMPU3026	

learning outcome of application development. The written exam will test the student's understanding of the concepts covered of application development but will also assess the learning outcomes related to wider knowledge of mobile phone technologies.

Essential Reading

This will be prescribed by the lecturer at the beginning of the mobile as it will be specific to the mobile platform used.

Supplemental Reading

This will be prescribed by the lecturer at the beginning of the mobile as it will be specific to the mobile platform used.

Web references, journals and other

A series of relevant content references will be provided to students to support their learning activities.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Individual Project**
		10	CMPU3018	

8.3.12 Individual Project

Module Author

Damian Bourke

Module Description

This module requires the student to propose, design, implement, test, document and present a software project to demonstrate the level of knowledge gained over the course of their studies. The project must be an independent piece of work which is both coherent and well structured. The student is expected to be self-motivated so as to drive this work through to completion. They are expected to identify the key areas of the project and to make real decisions that will ultimately affect the end deliverable. The project deliverable will be an assessable, independent body of work that will demonstrate the student's ability to work on their own and their ability to communicate key aspects of the project.

Module Aims

The aim of this module for the student is to complete a software system implementation. This includes all aspects of software development including, but not limited to; analysis, design, development, implementation, integration, and documentation. In addition the student is required to produce an academic quality report outlining the key steps undertaken, the key decisions that were made, an evaluation of the outcomes, and the student's conclusions about the success or failure of the delivered system.

Learning Outcomes

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

1. Define a problem area and write a project proposal
2. Evaluate similar systems to their proposal identifying a set of clear user requirements
3. Undertake research of the problem area to determine the boundaries and scope of the project
4. Undertake research of the proposed solution to identify appropriate technologies to use

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Individual Project**
		10	CMPU3018	

5. Select and implement a formal design methodology
6. Write a project plan and project manage the project to completion
7. Develop a fully operational software/hardware system
8. Produce a report (in English) of academic quality with appropriate referencing
9. Present and defend their findings

Learning and Teaching Methods

The project module involves a combination of seminars, self-paced work, research, and weekly meetings with their assigned supervisor. This is primarily a self-learning module with the supervisor providing guidance and feedback to the student throughout the project. The student is expected to state at the outset the objectives for the project and to meet these objectives throughout the year. The student is also given explicit verbal feedback at the Interim report stage which is a presentation by the student on the work completed so far and the work planned in the future.

In addition to the dedicated supervisory resources the School will also provide a second reader for the purposes of assessing the student's work.

Module content

Students are provided with detailed guidelines on the project process. These guidelines detail the requirements of the project in terms of the project content and deliverables and they also provide guidance on the content and format of the Interim Report and Final Project Manual. The guidelines also explain the roles of the key stakeholders including the student, the supervisor, second reader and project co-ordinator.

Module Assessment

The module assessment will comprise 100% Continuous Assessment.

There are two key milestones within the project lifecycle at which the student is assessed. At the Interim Report stage the student is assessed on their research findings, system design, report writing and project management. The assessment is based on a key set of criteria set-out in the guidelines document.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Individual Project**
		10	CMPU3018	

At the Final Report stage the student is assessed on any additional research undertaken post-design, the system implementation phase, testing and evaluation phase, report writing, communication skills, project management, and the quality, complexity and usability of the delivered system. The assessment is based on a key set of criteria set-out in the guidelines document.

Essential Reading

Damian Bourke, 2010. The Project Guidelines

Kathy Schwalbe, 2005, A Guide to the Project Management Body of Knowledge: Course Technology

Strunk, W. and White, E. B., 2000, The Elements of Style (4th ed.) Longman

Booth, W., Colomb, G.C., Williams, J.M., 2003, The Craft of Research, University of Chicago Press

Dawson, C., 2005, "A Practical Guide to Research Methods: A User-friendly Manual for Mastering Research Techniques and Projects", How To Books

Supplemental Reading

Evaluating Software Architectures Methods Studies

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Quantitative Methods and Tools for Data Analysis**
Mathematics 2		5	CMPU3033	

8.3.13 Quantitative Methods and Tools for Data Analysis

Module Author

Svetlana Hensman

Module Description

This module introduces the students to the basics of the probability theory and statistical methods. It provides a background and variety of examples of the application of the different methods for collecting and presenting data, as well as the statistical techniques used in practice. It also provides knowledge of some of the currently available statistical packages.

Module Aims

The aims of this module are to:

- Provide the student with a comprehensive foundation in probability theory and statistics
- Enable the students to scientifically collect, interpret and analyse data and familiarise them with advantages of using statistical packages
- Introduce the theoretical and practical aspects of the applications of the linear models in statistical data analysis and illustrate tests for differences between groups by applying analysis of variance

Learning Outcomes

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

1. Calculate probabilities of events and expectations of random variables for elementary problems, such as games of chance
2. Understand and apply a variety of methods for collecting, representing, visualising and analysing data and recognise the distribution of collected data
3. Apply statistical models and methods to solve wide range of practical problems and evaluate the suitability of a chosen statistical model
4. Interpret the results of a statistical analysis

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Quantitative Methods and Tools for Data Analysis**
Mathematics 2		5	CMPU3033	

5. Use hypothesis tests to make decisions
6. Perform analysis of variance (ANOVA)
7. Confidently use statistical software

Learning and Teaching Methods

The module is delivered in the form of one lecture, one tutorial and one lab session weekly over one semester. This will allow the use of a combination of variety of learning and teaching methods, including lectures, discussions, problem-solving exercises, computer-based learning, self-directed learning and problem-solving.

Module content

Probability Theory

- Review of the probability rules. Bayes' theorem. Random variables – discrete and continuous. Expected values and variances. Chebyshev's inequality. Binomial, Poisson and Normal distributions.

Estimation and Hypothesis testing

- Point and interval estimates of means and proportions. Large and small samples. Z and t-tests. Chi-squared tests.

Linear Regression Simple linear regression.

- Method of the least squares. Appropriateness of a regression model. Coefficient of determination and correlation coefficient. Transformation to linear form.

Analysis of Variance

- One way completely randomized design. F-test for equality of treatment means. Estimation of model parameters.

Module Assessment

Class test will comprise 40% of the marks for this module, and examination, the remaining 60%.

Essential Reading

Neter J., Wasserman W. and Whitemore G.A., 1993, Applied Statistics, Prentice-Hall, Inc.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Quantitative Methods and Tools for Data Analysis**
Mathematics 2		5	CMPU3033	

Walpole R. and Myers R.,1989, Probability for engineers and Scientists, Macmillan Publishing.

Supplemental Reading

Levine D., Ramsey P. and Smidt R. 2001, Applied statistics for engineers and scientists: using Microsoft Excel and MINITAB, Prentice-Hall, Inc.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Service-Learning and Civic Engagement**
		30	CMPU3035	

8.3.14 Service-Learning and Civic Engagement

Module Author

Ciarán O'Leary

Module Description

This module is designed to help learners develop their understanding of how their behaviour as graduates and professionals can impact upon the world around them, including how their skills and abilities can and should be used with a sense of responsibility to improve the lives of others. As such, it meets the requirements of a modern liberal education, and is informed by the skills requirements of employers and professional accreditation bodies.

Learners will be required to participate actively and meaningfully in a six month long project, whereby they are immersed in an environment different to their typical environment. The learner must acquire sufficient knowledge and skills during this period to complete a project which meets a specific community need related to the learner's domain of study.

The precise details of the project are defined through negotiation with the module coordinator and a partner organisation, based in a disadvantaged community, a community in a developing country, or some similar environment. The partner organisation will be sourced by the module coordinator, and will typically be a voluntary or non-governmental organisation or charity.

Depending on their domain of study, the student may participate in a variety of activities during this time. Computing students, for example, may spend this time teaching basic computing skills to pupils or teachers. Others may teach English, for example, or use their domain skills for some activity agreed through negotiation with the module coordinator and the partner organisation.

The learner uses the experience to discover and assess a particular need of the partner organisation or community. The need must be related to the learner's domain of study, and will lead to the implementation of a project to address this need before the completion of the module.

The module is divided into four components, which can be completed

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Service-Learning and Civic Engagement**
		30	CMPU3035	

in series or in parallel, or a combination of both, depending on the nature of the project and partner organisation. Similarly, the time periods allocated to the individual components will vary depending on the nature of the project. Crucially, all decisions relating to the design of the project will be made through negotiation between the learner, the partner organisation and the module coordinator.

The learner's personal development and ongoing reflection on their experience is intended to contribute to the development of their world-view and understanding of their current and future role in society. As such, the learner is required to negotiate a personal development plan and complete reflection tasks which inform and demonstrate their learning.

The key transferable skills of problem solving, written and oral communication, critical and creative thinking, teamwork, negotiation, and civic responsibility and engagement represent the core of this module. These skills are identified in numerous reports and investigations as the key skills expected of graduates by employers.

Module Aims

The aims of this module are to:

- Develop the learner’s critical thinking, creative thinking and problem solving by presenting them with real problems in the real world.
- Require the learner to communicate effectively and with impact while interfacing with professionals and others.
- Encourage the learner to broaden their world-view and develop their understanding of their current and future role in society and their community.
- Facilitate the learner in developing their understanding of the potential role for the skills acquired from their specific domain of study in disadvantaged communities and societies.
- Develop the learner’s organisational abilities and sense of responsibility, as required of all effective professionals.
- Encourage personal reflection on the learner’s values, motivation and contribution to society.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Service-Learning and Civic Engagement**
		30	CMPU3035	

Learning Outcomes

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

1. Effectively communicate with others, using verbal, written and other means as required.
2. Develop strategies for identifying needs, assessing and defining problems, critically analysing potential solutions, and solving real world problems related to their domain of study.
3. Fully appreciate the importance of good timekeeping and time management, commitment and responsibility, personal presentation and professionalism in the work environment.
4. Work closely with others in a variety of roles and appreciate the value of working with, as well as for, others.
5. Identify and describe the role that their domain of study plays in the world, and how they, as professionals, can help in addressing the problems of disadvantage and exclusion in the modern world.

Learning and Teaching Methods

- Class-based seminars and tutor-led group reflection sessions during the latter two stages.
- Self-directed site-based research and project work.
- Web-based research and literature analysis.
- Supervision and feedback from the module coordinator and partner organisation.

Module content

The module is divided into four components, which can be completed in series or in parallel, or a combination of both, depending on the nature of the project and partner organisation. The time periods allocated to the individual components will vary depending on the nature of the project. All decisions relating to the design of the project will be made through negotiation between the learner, the partner organisation and the module coordinator.

Prior to the commencement of the project, interested learners must apply to participate in the module. Participants will be selected based on a number of criteria, including their existing commitment to

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Service-Learning and Civic Engagement**
		30	CMPU3035	

voluntary work and social engagement, if any, their proven reliability and ability, and references. If selected, the learner usually will begin negotiating with the module coordinator regarding how they will meet the learning outcomes of the module. If required, the learner usually will begin fundraising at this point.

Preparation: Starting in semester 2 of the academic year (usually at the end of January or start of February) the learner will formally begin the module. The learner will complete negotiation of the learning contract and personal development plan with the module coordinator and partner organisation, and will commence a period of formal preparation. Preparation may include collection or design of learning and teaching materials, development of websites, accumulation of contacts, continuation of fundraising, organisation of travel, negotiation of services, learning of a language, completion of short courses, search of relevant literature etc.

Immersion: The learner will be based in a disadvantaged community or society for an agreed period of time. The learner will be supervised by members of a support organisation or network, approved by the module coordinator. The learner will carry out the work required of them as agreed in the learning contract. During this period the learner will assess the needs of the community with a view to suggesting a role for their domain of study in that community. As an example, electronic engineering students may assess the need for satellite or mobile devices, computing students may assess the need for improved selection of software in schools or colleges, other learners may assess the need for improved access to learning materials in schools or colleges. The precise details of the needs will not be known until the learner experiences the environment during the immersion component of the module.

Project: The learner will complete a project based on and informed by their experience during the immersion component. The learner may, for example, develop software, compile learning materials, complete documentation or anything else as agreed in their learning contract.

Completion: Coming at the end of the module, the learner may revisit the community to implement their project, or may hand over their project with required documentation to someone else who will continue the work. The learner will complete their submission of learning materials including, for example, reflection on critical incidents which occurred during the module.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Service-Learning and Civic Engagement**
		30	CMPU3035	

Module Assessment

The learner is to participate in a personal development process, beginning with the development of a personal development plan. As part of this, the learner is required to select and employ an appropriate reflection model, to address their personal, academic and societal reflection which demonstrates their learning. This will be supplemented with the submission of a final report, documenting the learner's experience and personal development.

The quality of the project and feedback from the organisation with which they were working will contribute significantly to their assessment.

Specific details on the assessment weighting will be agreed with the learner as part of the negotiation of their learning contract.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Engineering 3**
Software Engineering 2		5	CMPU3038	

8.3.15 Software Engineering 3

Module Author

Oliver O'Connor

Module Description

This module builds on the Object Oriented Software Engineering material from stage 2 by a covering advanced Analysis and Design. It extends Implementation and reviews typical Technologies. Testing throughout the process is emphasised. Object Oriented Methodologies and Project Management are incorporated.

Module Aims

The aims of this module are to:

- Reinforce and extend the skills and knowledge gained in Stage 2 into a comprehensive Object Oriented approach.
- Provide the student with the skills and knowledge to document and manage their Final Year Project and apply this training to the development of software systems project in the workplace.
- Integrate and support the related modules in Web Development, Database Technology and User Interface Programming.

Learning Outcomes

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

1. Apply concepts and skills learnt to conduct an analysis and design and implementation of a system using Object Oriented techniques and languages.
2. Identify the Design Patterns and choices of Software Technologies used to implement a system.
3. Design the software architecture for a system and identify the software technologies suitable to implement the various layers and the interfaces between the layers.
4. Apply CASE tools to support the development process and apply reverse and forward engineering where appropriate.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Engineering 3**
Software Engineering 2		5	CMPU3038	

5. Describe in detail ancillary activities such as Project Management and analyse a typical project using Network Analysis.
6. Design Test Strategies and Test Cases and have a sound appreciation of the importance and role of testing throughout the development process.

Learning and Teaching Methods

Lectures with tutorials and laboratory sessions to give the student experience in developing models and specifications of a variety of suitable systems with lecturer feedback. A team-based case study with an assessment component will reinforce group learning and enable larger applications to be considered. The use of a CASE tool will provide standards, rigour and guidance to the specifications. Collaboration with the associated software development courses to reinforce the skills.

Module content

Review the nature of an object, Use Case Diagrams, Class Diagrams, Sequence Diagrams and State Machine Diagrams.

Advanced Analysis: Class modelling, Class Layers, Generalisation, Inheritance, Aggregation and Delegation modelling.

Design: System Architecture, Design Patterns, Technologies, Mapping Analysis Classes to the Relational Database Model. Reusable Design Patterns. The Presentation Layer, User Interface guidelines and implementation techniques.

Implementation: Component Diagrams, Deployment Diagrams, Strategies, Reviews.

Code Generation and Reverse Engineering tools and techniques.

Continuous Testing: Types of Tests, Strategies, Test Driven Development. Automated Tests, and Change Management.

Project Management: Computer Project Organisation: (Teams, Goals, Deliverables).

Management Techniques: Network Analysis/PERT, Gantt Charts, Sensitivity Analysis, Resource Smoothing and Allocation, Optimization Techniques.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Engineering 3**
Software Engineering 2		5	CMPU3038	

Module Assessment

Formative assessment is via continuous assessment using practical assignments. Summative assessment is via an end of module written examination.

There is a 30% weighting for the continuous assessment and a 70% weighting for the examination.

Essential Reading

Mark Priestley, Practical Object-Oriented Design with UML, McGraw Hill, 2003.

S. Bennett, S. McRobb and R. Farmer, Object-Oriented Systems Analysis and Design using UML, 4rd Ed , McGraw-Hill, 2010

Supplemental Reading

R. Pressman, Software Engineering: A Practitioner’s Approach., 6th Ed.,McGraw Hill 2005

Relevant supplemental references will be indicated during the teaching of the module.

Web references, journals and other

Up to date Web references placed in the course notes web page on specific topics

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**System Administration**
Operating Systems 2		5	CMPU3049	

8.3.16 System Administration

Module Author

Ronan Bradley

Module Description

This module provides the student with knowledge of the fundamentals of UNIX and Windows system administration including common network services for authentication, naming and file sharing.

Module Aims

The aim of the module is to equip the student with a sufficient understanding of and skills in the configuration and administration of a mixed network of UNIX and Windows based computers. The students will acquire an understanding of how the core services such as naming and file sharing operate from both theoretical and practical perspectives.

Learning Outcomes

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

1. Configure the user and group management mechanisms on Windows and UNIX systems
2. Modify the start-and shutdown order of services on a UNIX system.
3. Utilise performance analysis tools on the UNIX platform
4. Configure authentication mechanisms on the UNIX platform and between Windows and UNIX Platforms
5. Configure file sharing mechanisms on the UNIX platform and between Windows and UNIX platforms
6. Configure host name resolution mechanisms on the UNIX platform and between Windows and UNIX platforms
7. Design a systems integration plan to support file sharing, identity sharing and other resource sharing between UNIX and Windows systems.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**System Administration**
Operating Systems 2		5	CMPU3049	

Learning and Teaching Methods

This module will be taught through a combination of lecture, tutorials and practical lab-based sessions.

Supervised practical sessions will be used to complement material presented in lectures. Concepts from lectures will be applied via supervised lab based exercises and practical assignments. It will be expected that all students complete their exercises each week and be pro-active in their learning approach.

Module content

UNIX System start-up and shutdown control including configuration of run-levels

User Management on UNIX and Windows Systems including adding and removing users and groups

Operation of common performance analysis tools to analyse the current performance characteristics of the system and identify potential problems

Configuration and integration of authentication mechanisms on UNIX (e.g. Portable Authentication Module and directory services supporting Network information Service and LDAP used for authentication)

Selection of appropriate file sharing services for different networks

Configuring of file sharing services on UNIX such as SaMBa and nfs

Operation of the Domain Naming System and configuration using bind on a UNIX system

Overview of other network services commonly used (such as NTP)

Module Assessment

Assessment by a combination of Continuous Assessment and Exam

Exam 60%, Continuous Assessment 40%

Essential Reading

Marty Poniatowski, 2002, UNIX User’s Handbook, 2nd ed., Prentice Hall PTR

Steve Shah, Wale Soyinka. 2005, Linux Administration: A Beginner’s Guide. Osborne/McGraw-Hill

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**System Administration**
Operating Systems 2		5	CMPU3049	

Supplemental Reading

Relevant supplemental references will be indicated during the teaching of the module.

Web references, journals and other

Relevant web references, journals and other will be indicated during the teaching of the module.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**System Security**
		5	CMPU3042	

8.3.17 System Security

Module Author

Fredrick Mtenzi

Module Description

Recent developments in the computing technology, mobile device technologies and the increasing broadband availability globally are shaping our lives in ways unimaginable a few years ago. While these developments have provided an unprecedented access to information and connectivity, they have also led to increased security concerns regarding the safety of our systems, data and wellbeing. The rush to market software which forms the trend of most of the modern technology has enhanced the release of insecure software.

This module provides an overview of the fundamental concepts of security. It will introduce students to the basic principles of securely using computers, wired and wireless networks and the internet. Emphasis will be placed on understanding issues which will lead to sensible security and an appreciation that security is the weakest link problem. The user security social responsibility which is becoming more important now will be covered in the legal and ethical aspects of security. Relevant concepts covered in the module will be experimented by students in the labs.

Module Aims

The aims of this module are to:

- introduce the students to the security principles,
- give the students a thorough understanding of the network security issues,
- provide them with sound practical knowledge of security tools, Operating Systems and Network security,
- provide them with an in-depth practical security knowledge in real-life.

Learning Outcomes

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

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**System Security**
		5	CMPU3042	

- Define and describe network security,
- Discuss and relate the fundamental concepts of Security,
- Classify and analyse the nature and threat from malware,
- Design and deploy best practice techniques for securing Operating Systems and Networks,
- Compare and contrast different security tools and determine their appropriate use in a specific situation,
- Examine the trade-off of legal and ethical security issues in an organisation
- Asses and rank different systems security approaches according to the protection they provide in real-life applications.
- Recognise and deploy the different hacking phases in systems.

Learning and Teaching Methods

In this module a number of teaching methods may be employed including lectures, practical sessions, tutorials and case studies. At least one industrial seminar may be arranged. Focus should be placed on empowering the students to develop their skills independently of the presence of a tutor or lecturer.

Module content

Introduction to security

Classical encryption techniques and stream ciphers

User Authentication

Access Control

Physical and Infrastructure Security

Operating Systems Security

Network Security

Security Tools

Introduction to ethical hacking

The role of people in security

Legal and ethical aspects of security

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**System Security**
		5	CMPU3042	

Module Assessment

This module should have a 50% weighting for the examination and a 50% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, it is equally important that they demonstrate an understanding of the theoretical aspects of security.

Essential Reading

William Stallings, 2005, Cryptography and Network Security: Principles and Practice, 5th ed, Prentice Hall

Supplemental Reading

Seymour Bosworth and M.E. Kabay, 2009, Computer Security Handbook, John Wiley & Sons. Inc.

William Stallings and Lawrie Brown, 2008, Computer Security: Principles and Practice, Prentice Hall

Markus Jakobsson, Zulfikar Ramzan, 2008, Crimeware: Understanding New Attacks and Defences, Symantec Press.

Charles P. Pfleeger and Shari Lawrence Pfleeger, 2007, Security in Computing, Prentice hall

Michael Gregg, 2008, Certified Ethical Hacker, Que Publishing

Vincent Nestler, Gregory White, Wm. Arthur Conklin and Corey Schou , 2010, Principles of Computer Security CompTIA Security+ and Beyond Lab Manual, McGraw-Hill Companies

Web references, journals and other

IEEE security & privacy

ACM Transactions on Information and System Security (TISSEC)

http://www.sans.org/rr/

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Team Project**
		10	CMPU3045	

8.3.18 Team Project

Module Author

Ronan Bradley

Module Description

This module offers the students the opportunity to use the knowledge and skills they have developed over the previous stages in the development of a complex software system. The project is a team project requiring the students to collaborate and organise themselves in a group to achieve the module objectives.

Module Aims

This module aims to group students into small teams and familiarise them with a range of skills and knowledge necessary, to take a software application through its life cycle.

Learning Outcomes

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

1. Demonstrate group interaction
2. Demonstrate ability to deliver individual objectives within a team structure
3. Demonstrate leadership and team collaboration skills
4. Demonstrate an ability to follow development processes from initiation through design, implementation, test and delivery
5. Implement a project design in an appropriate format from a requirements specification
6. Demonstrate a choice of appropriate technology
7. Implement role functionality
8. Determine group integration.
9. Develop presentation skills

Learning and Teaching Methods

This module will be taught through a combination of lecture, tutorials

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Team Project**
		10	CMPU3045	

and practical lab-based sessions.

The supervised practical sessions will be used to allow lecturers to review and assess the progress of the whole team and the individual team members. It will also be used for team demonstrations and presentations at least twice during the semester.

Module content

Students will be presented with a requirements specification, from which they must develop a software system in a group of between three and five students. These groups will be pre-assigned.

Topics which will be covered as part of the module include

- Project planning and management (including documentation requirements)
- Test planning and test implementation
- Application of software engineering concepts to team development projects

Module Assessment

The module is assessed entirely by continuous assessment. The continuous assessment mark is determined through a combination of report submissions, submission of the student's project journal, presentations and interviews which take place throughout the semester.

These are as follows:

- Group submission of project documentation artefacts (such as design documents, test plan, testing logs and project plans)
- Group presentation and defence of
  - the design (prior to implementation)
  - the final deliverable (including demonstration)
- Individual submission of a project journal
- Individual interview of each student within a group by the lecturer

A student's final mark will be composed of 40% of their group mark and 60% of their individual mark.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Team Project**
		10	CMPU3045	

Essential Reading

To be provided at the start of the module.

Supplemental Reading

To be provided at the start of the module.

Web references, journals and other

To be provided at the start of the module.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Universal Design and Assistive ICT**
		5	CMPU3046	

8.3.19 Universal Design and Assistive ICT

Module Author

John Gilligan, Dave Carroll, Damian Gordon, Ciaran O'Leary

Module Description

This module embraces the paradigm of Assistive Technology. Its subject matter ranges from Disability Awareness, an overview of technology, the legal social and economic imperative of Accessible IT, Human Computer Interaction, Universal Design, Accessible Software and Accessible Web Design. It is intended to provide the learner with the necessary understanding and skills to develop Accessible User interfaces to Application Software and Accessible Web pages. The learner should understand the challenges of developing Inclusive IT solutions and be able to critique available solutions.

Module Aims

The aims of this module are to:

- Describe the context of Disability and explore the issues involved in Assistive Technology.
- Investigate the specific challenges of an inclusive Information Technology society.
- Demonstrate the relevance of the learners own IT skills to facilitating an inclusive Information Technology society.

Learning Outcomes

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

1. Describe the social and legal imperatives with regard to Assistive Technology.
2. Appreciate the environmental constraints of people with disability.
3. Describe the various technologies and specialised interfaces to assist people with disability.
4. Describe principles of user interaction and identify various disabilities that impact the Human Computer Interface.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Universal Design and Assistive ICT**
		5	CMPU3046	

5. Assess the accessibility of software within a universal design framework.
6. Design and implement software interfaces with improved accessibility.
7. Design and construct inclusive Web sites.

Learning and Teaching Methods

This module will be delivered using a combination of Lectures, Practicals, Case Studies, Role Playing, Discussions, and Thinking Skills. The main theoretical material will be delivered in lectures. The practical work to assess the accessibility of software, and design and implement accessible software interfaces evaluate accessibility, and inclusivity of Web sites will be carried out in laboratory sessions.

Module content

Disability Awareness: Overview of different conditions. Analysis of requirements imposed by these conditions. Social, Legal and Economic Models of Disability. Service Delivery Models.

Technical Overview: Relevant Technologies. Product Review. Software Support Systems. Hardware Overview. Specialized Interfaces. Augmentative Alternative Communication: Switch Access: Voice Recognition: Text to Speech Computer Vision: Environmental Control and Independent Living.

Specialized Interface design: Analysis of Requirements. Technical Constraints. Design Issues. Universal Design, Challenging the HCI user Model. Enhanced models of HCI.

Inclusive IT Universal Design Access issues, Matching Person and Technologies, Frameworks, Accessible Databases, Web Access. Usability, Technical Support.

The Education Paradigm Connecting to Learn, Education and Technology, Learning Styles, Individual Education Plans.

Module Assessment

Written Examination 70%

Practical 30%

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Universal Design and Assistive ICT**
		5	CMPU3046	

Essential Reading

Cook and Hussey (2007), Assistive Technology Principles and Practice, (3rd Ed), Mosby.

Supplemental Reading

Scherer, Marcia J. (2003) Connecting To Learn, Educational and Assistive Technology for People With Disabilities, American Psychology Association

Web references, journals and other

National Disability Authority WWW.NDA.IE

Web Access Initiative www.w3.org/wai

Higher Education Disability Site www.Ahead.ie

Central Remedial Clinic www.crc.ie

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Work Placement**
		30	CMPU3048	

8.3.20 Work Placement

Module Author

Paul Doyle

Module Description

This module is a 6 to 7 month full time industrial placement which provides students with the opportunity to obtain professional industry experience.

Module Aims

The aim is to give students the opportunity to gain a structured learning experience within a professional / industrial environment so that they may consolidate and broaden the knowledge that they have gained in their academic studies to date. Students should gain a deep understanding of the business within which they are placed.

Learning Outcomes

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

1. Work in a professional environment
2. Demonstrate extensive understanding of a particular project or area of work that they have been involved in
3. Reflect on the learning experience and outcomes of their work experience
4. Produce a professional report describing the details and experiences of their placement
5. Evaluate their contribution to the company
6. Critically assess the relationship between their academic knowledge and industrial experience
7. Demonstrate an understanding of the company’s business model

Learning and Teaching Methods

During the time in industry students may be working as part of a team or individually. Students are assigned a DIT industrial placement monitor for the duration of their work placement and deliver reflective

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Work Placement**
		30	CMPU3048	

documents demonstrating their learning as part of continuous assessment.

Module content

Professional Development

Structured reporting on work performed

Teamwork within a professional organisation

Reflective writing and reporting

Presentation of work

IT skills specific to the work placement

Module Assessment

Assessment will be based entirely on continuous assessment.

The continuous assessment element of this course is comprised of regularly submitted reports by students on work placement, an end of year written report and an end of year presentation. The end of year presentation and report is jointly assessed by the Industrial placement monitor and the industrial placement coordinator.

Essential Reading

Jennifer A. Moon, 2004 A Handbook of Reflective and Experiential Learning: Theory and Practice ,

Lydia E. Anderson, 2007, Professionalism: Real Skills for Workplace Success, Prentice Hall

Paul Doyle, DIT School of Computing Industrial Placement Handbook

Supplemental Reading

David Boud, Reflection : 1985, Turning Experience into Learning, Routledge

Dannelle D. Stevens , Joanne E. Cooper, 2009, Journal Keeping: How to Use Reflective Writing for Learning, Teaching, Professional Insight and Positive Change , Stylus Publishing

Web references, journals and other

http://www.comp.dit.ie/industrialplacement

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Work Placement**
		30	CMPU3048	

http://www.dit.ie/careers/

8.4 Stage 4 Syllabi

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Databases**
		5	CMPU4003	

8.4.1 Advanced Databases

Module Author

Susan McKeever, June Barrett, Bryan Duggan

Module Description

This module provides knowledge and practical application of advanced database concepts, building on the experience and knowledge gained on core database modules from previous years. The module covers advanced and complete database concepts relating to storage, management and architecture of data. It also covers topics in advanced database design, focussing on both logical and physical design aspects. Emerging database technologies and themes will also be studied.

Module Aims

The aim of this module is to study and practice advanced database features and techniques for relational and other database models; and to understand alternate database architectures in the context of applications and information systems.

Learning Outcomes

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

1. Employ fundamental and advanced database modelling techniques for conceptual, logical and physical design;
2. Understand and apply application versus database considerations for integrity and security
3. Understand and implement optimisation and performance tuning, concurrency control, recovery and integrity on a relational database;
4. Critically analyse the limitations of the relational model;
5. Analyse the strengths and weaknesses of alternate database models;
6. Discuss the database design issue and application scenarios of distributed databases;
7. Compare the role and features of object oriented databases to

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Databases**
		5	CMPU4003	

relational ;

8. Discuss recent development and emerging trends in database technologies;
9. Discuss and discriminate between different approaches to data integration, both semantic and physical;

Learning and Teaching Methods

This module will be taught through a combination of lecture, seminars and practical lab-based sessions.

Lectures will include case studies, industry examples and practical examples in order to support the learning outcomes. Students will be encouraged to be pro-active in their approach to learning through the use of case studies and exercises, working independently and in groups.

Supervised practical sessions will be use database design and a leading database system, so that students will be familiar with industry standard tools. Concepts from lectures will be applied via supervised lab based exercises and practical assignments that will be completed in the students' own time.

Module content

Advanced database design, covering and distinguishing between conceptual, logical and physical design;

Query optimisation and performance tuning techniques;

Application versus database-level considerations; data integrity and security;

Data Management: transactions, concurrency control, recovery

Distributed databases; architectures, design , fragmentation and replication, data integrity management, performance optimisation;

Relational database model features, strengths, weaknesses versus other models;

Object oriented databases: features, examples, strengths and weaknesses;

Data and database integration ;

Emerging database technologies and application areas.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Databases**
		5	CMPU4003	

Module Assessment

Continuous assessment – 40%

Written Examination – 60%.

Continuous assessment will take the form of exercises and assignments related to the exploration and application of course content. The work will involve investigating advanced database topics relevant to information systems using suitable case studies, research papers and other peer reviewed sources. This may be done individually or as part of a group. Where possible and appropriate, contact practical sessions will be used to provide the student with time and assistant to complete continuous assessment work.

Essential Reading

Ramaz Elmasri, 2010, Fundamentals of Database Systems 6th Edition, Addison Wesley

Connolly and Begg, 2010, Database Systems: A Practical Approach to Design, Implementation and Management, 5/e, Addison Wesley.

Supplemental Reading

Relevant supplemental references will be indicated during the teaching of the module.

Web references, journals and other

www.sqlzoo.net; www.w3schools.com;

Additional web references, journals and other reading materials will be indicated during the teaching of the module.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**App Development and Commercialisation**
		5	CMPU4009	

8.4.2 App Development and Commercialisation

Module Author

Susan McKeever and Bryan Duggan

Module Description

Recently, the number of apps downloaded by iPhone/iPad/iPod touch owners exceeded ten billion (Jan 2011), while the number of apps downloaded by users of Google's Android Operating System exceeded two billion. The worldwide mobile app market is projected to be worth $15 billion by 2013. There is a low barrier to entry for developers who wish to develop and sell apps and a number of indigenous Irish companies have been successful in targeting the app market worldwide. The purpose of this course will be to enable students to target this lucrative market by learning how to develop their app ideas into commercial products. The course will cover an introduction to app development on platforms including (but not restricted to) iOS, Android, Symbian and Windows Phone 7. In addition to technical considerations, the course will also cover commercial considerations such as project lifecycle/management, funding/investment, business models, preparing a business plan and marketing plan.

Module Aims

The aim of this module is to provide students with the technical skills and commercial knowledge required to take their app concepts to market.

Learning Outcomes

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

1. Demonstrate an understanding of the capabilities of a range of mobile devices and operating systems such as iOS, Android, Symbian, RIM and Windows Phone 7
2. Demonstrate an understanding of the types of apps available on these platforms
3. Understand the certification process involved in registering for an app marketplace
4. Use a development environment to develop apps

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**App Development and Commercialisation**
		5	CMPU4009	

5. Design and develop a touch optimised user interface
6. Develop network aware, database powered apps
7. Make use of geo-coding capabilities of mobile devices
8. Make use of touch screens, accelerometers and cameras
9. Make use of graphics and audio in apps
10. Submit an app to a marketplace for approval
11. Write a business plan leading to app commercialization

Learning and Teaching Methods

Students will learn through instructor led, interactive “studio classroom” sessions. In the studio classroom, students have access to hardware, software and development kits, so that material and examples can be examined in a live environment. In addition, students will be expected to proactively and independently seek out resources on the internet and from the library to supplement their own learning. Students will have access to SDK’s, toolkits and hardware to install on their own laptops. Guest lecturers from industry and the DIT Hothouse will be invited to present successful app case studies.

A Virtual Learning Environment (VLE) will be employed to distribute all teaching materials and to support student interaction with both other students and academic staff.

Module content

Mobile devices, apps, marketplaces and trends

Agile app development methodologies & the app lifecycle

Software development environments for app development – including but not limited to: XCODE iOS SDK, Eclipse and the Android SDK, Visual Studio and Windows Phone 7 SDK.

Setting up hardware for development.

Languages, frameworks, collections and memory management

Touch user interface design and implementation

Mobile database development

Networking and XML parsing

Graphics and audio programming

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**App Development and Commercialisation**
		5	CMPU4009	

Geo and movement APIs

Publishing an App: Business models, funding, marketing, PR, legal and financial issues

App case studies

Module Assessment

This module has a 60% weighting for the examination and a 40% weighting for the continuous assessment. Continuous assessment will consist of a single significant assignment (with multiple deliverables), which may be individual or team based. The assignment will involve the planning, development and optional publishing of an app. Students will be required to document their progress on the assignment in the form of a blog or ePortfolio. Students will be encouraged to enter their apps into national and international competitions such as the XNA Ireland Challenge, the Imagine Cup or the Appys. Students will be expected to prepare a business plan as part of their assignment outlining potential revenue streams. They will also be encouraged and supported to commercialize their assignment work by submitting to an app store at the end of the project.

Essential Reading

Alasdair Allan, Learning iPhone Programming: From Xcode to App Store, 1st ed. (O'Reilly Media, 2010).

Meier, Reto, Professional Android 2 Application Development (Wrox Programmer to Programmer), (March 1, 2010)

Randolph, Nick, Professional Windows Phone 7 Application Development: Building Applications and Games Using Visual Studio, Silverlight, and XNA (Wrox Programmer to Programmer), (November 9, 2010)

Web references, journals and other

Additional web resources and more specific book references will be provided subject to the environments that are being specialised in in this fast changing subject area.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Security 1**
		5	CMPU4007	

8.4.3 Advanced Security 1

Module Author

Fredrick Mtenzi

Module Description

Cryptography is the science of providing security for information through the reversible transformation of data. It can also be defined as the study and practice of scrambling information in a manner that is difficult to unscramble, and making scrambled information unintelligible. It is used as the basis of much computer security, in that it can be used to keep information confidential, and also preserve the integrity of data, particularly when being stored or being transmitted. The development of digital computing revolutionized the use cryptography, and has allowed its usage in most computing activities.

Apart from government, military and organisations, users need security provided in cryptography. For example, they will need this kind of security because they may be designing new software, discussing a marketing strategy, or planning a hostile business takeover. The aim of this course is to provide a detailed understanding of the issues involved in Steganography and Cryptography algorithms and techniques used in implementation. The course will introduce and make use of the relevant mathematical concepts such as number theory and finite fields. Practical applications of Steganography and Cryptography will be demonstrated and experimented by students as part of the laboratory exercises or assignments.

Module Aims

The aims of this module are to:

- introduce the students to the principles of cryptography and steganography,
- give the students a thorough understanding of cryptography and steganography algorithms,
- provide them with an in-depth understanding of how cryptography and steganography provides security in real-life applications.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Security 1**
		5	CMPU4007	

Learning Outcomes

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

1. Describe the underlying principles of different cryptography and steganography algorithms.
2. Design and implement simple cryptography algorithms using any high level programming language.
3. Evaluate the effectiveness of cryptography algorithms according to well known security requirements.
4. Apply relevant/appropriate mathematical results in the design and implementation of cryptography algorithms.
5. Describe the principles and deployment of steganography techniques as applied in real-life security.
6. Compare and contrast on the effectives and efficiency of cryptography and steganography algorithms.
7. Recognise and justify the different scenarios of deploying cryptography and steganography algorithms.
8. Select and combine different cryptography algorithms in order to achieve highly secure algorithms.
9. Assess and rank cryptography and steganography algorithms according to the protection they provide in real-life applications.
10. Explain the principles behind and usage rationale of various encryption techniques,

Learning and Teaching Methods

In this module a number of teaching methods may be employed including lectures, practical sessions, tutorials and case studies. At least one industrial seminar may be arranged. Focus should be placed on empowering the students to develop their skills independently of the presence of a tutor or lecturer.

Module content

Introduction to cryptography

Number theory, Discrete logarithms and Elliptic Curves

Steganography

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Security 1**
		5	CMPU4007	

Block Ciphers and Advanced Encryption Standard, Confidentiality Using Conventional Encryption.

Public-Key Cryptography and RSA, Key management, Message Authentication and Hash Functions, Hash and Mac Algorithms, Digital Signatures and Authentication Protocols.

Module Assessment

This module should have a 60% weighting for the examination and a 40% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, it is equally important that they demonstrate an understanding of the theoretical aspects of cryptography.

Essential Reading

William Stallings, 2005, Cryptography and Network Security: Principles and Practice, 5th ed, Prentice Hall

Supplemental Reading

Niels Ferguson, Bruce Schneier, Tadayoshi Kohno, 2010, Cryptography Engineering: Design Principles and Practical Applications, John Wiley & Sons. Inc.

Behrouz A. Forouzan, 2008, Cryptography and Network Security, McGraw-Hill International Edition.

Wade Trappe, Lawrance C. Washington, 2006, Introduction to Cryptography with Coding Theory, Prentice Hall.

Web references, journals and other

IEEE security & privacy

ACM Transactions on Information and System Security (TISSEC)

http://www.sans.org/rr/

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Security 2**
		5	CMPU4008	

8.4.4 Advanced Security 2

Module Author

Fredrick Mtenzi

Module Description

In the last few decades, computers, mobile devices, networks and the Internet have become an integral part of our social fabric. As computer, networks and the Internet are used for communication and for varieties of online interactions and transactions, security has become the key issue in today's information technology world. Information security is required at all levels – the personal level, corporate level, and country level. There is a steady rise in the occurrence of security attacks. Security is a broad issue which is becoming increasingly important as computer networks become more widespread. It encompasses computer- and network-related crime, privacy issues, trust and confidence, and dependability of critical infrastructures. The evidence of the security threat is growing and new vulnerabilities are found each day. Those interested in exploiting these vulnerabilities are becoming a well-organized. Security threats are taking a variety of forms, including espionage, hacking, identity theft, crime, and terrorism. The level of sophistication and speed of development of the tools being used to create security breaches and attacks are growing exponentially.

With the increasing concern for safety and integrity of information against security attacks, it has become mandatory that organizations follow strict guidelines and security framework to assure the safety and protection of data and systems. To address these needs this module has will provide a detailed understanding of the issues involved in providing systems security. Further, the module will examine and critically analyze the security challenges faced by the society and the computing industry. It will elaborate and demonstrate the role of security in design, implementation and deployment of secure systems.

Module Aims

The aims of this module are to:

- introduce the students to the security principles,

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Security 2**
		5	CMPU4008	

- give the students a thorough understanding of the network security issues,
- provide them with sound knowledge of provision of an enterprise secure systems, security audit and compliance,
- provide them with an in-depth practical enterprise security.

Learning Outcomes

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

1. Define and explain the components of network security,
2. Discuss, relate and organise the fundamental concepts of Security,
3. Classify and analyse the nature and threat from malware,
4. Design and develop best practice techniques of mitigating security threats,
5. Critically analyze different aspects of security such as security testing, evaluation, auditing and policies,
6. Assess security compliance of an organisation
7. Compare and contrast international and regional security standards, legislations and laws.
8. Asses and rank different systems security approaches according to the protection they provide in real-life applications.
9. Recognise and justify the different scenarios of deploying systems security approaches.

Learning and Teaching Methods

In this module a number of teaching methods may be employed including lectures, practical sessions, tutorials and case studies. At least one industrial seminar may be arranged. Focus should be placed on empowering the students to develop their skills independently of the presence of a tutor or lecturer.

Module content

Authentication Applications,

Electronic Mail Security,

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Security 2**
		5	CMPU4008	

Internet Protocol Security,

Web security,

Intruders, Crimeware, Firewalls

Security Policies, Standards, Compliance

Security Metrics and Auditing

Penetration Testing

Defences to security attacks

The effect of Technological developments on Security

Module Assessment

This module should have a 50% weighting for the examination and a 50% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, it is equally important that they demonstrate an understanding of the theoretical aspects of security.

Essential Reading

William Stallings, 2005, Cryptography and Network Security: Principles and Practice, 5th ed, Prentice Hall

Supplemental Reading

Seymour Bosworth and M.E. Kabay, 2009, Computer Security Handbook, John Wiley & Sons. Inc.

Andrew Lockhart, 2004, Network Security Hacks 100 Industrial-Strength Tips & Tools, O'Reilly

Markus Jakobsson, Zulfikar Ramzan, 2008, Crimeware: Understanding New Attacks and Defences, Symantec Press.

Ed Skoudis and Tom Liston, 2006, Counter Hack Reloaded: A step-by-step Guide to Computer Attacks and Effective Defences, Prentice hall

Bruce Schneier, 2004, Secrets and Lies: Digital Security in a Networked World, Hungry Minds Inc

Web references, journals and other

IEEE security & privacy

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Advanced Security 2**
		5	CMPU4008	

ACM Transactions on Information and System Security (TISSEC)

http://www.sans.org/rr/

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Artificial Intelligence 1**
		5	CMPU4010	

8.4.5 Artificial Intelligence 1

Module Author

John Kelleher and Brian Mac Namee

Module Description

The most important concepts in introducing artificial intelligence are representation and search. This course introduces the student to the problems of knowledge representation and to the design of search strategies.

Module Aims

The module aim is to provide the student with an understanding of the problems surrounding knowledge representation, the range of structures available for knowledge representation, the relationship between representation structures and search strategies; and the ability to compare search strategies relative to their suitability to a given domain and their time and space complexity.

Learning Outcomes

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

1. Demonstrate a broad understanding of the roots and scope of Artificial Intelligence
2. Describe and compare a range of knowledge representation structures
3. Develop a knowledge based system, such as a planner or logical inference system
4. Describe and illustrate the functioning of a range of search algorithms
5. Demonstrate an appreciation of complexity issues and compare a range of search algorithms based on time and space complexity

Learning and Teaching Methods

Lectures will deliver theory and techniques. During practical sessions the students will implement the systems that utilise the techniques

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Artificial Intelligence 1**
		5	CMPU4010	

presented in the lectures and enable them to directly evaluate the different approaches.

Module content

Artificial Intelligence: its roots and its scope

- Historical Foundations, Overview of Themes and Applications

Structures and strategies for state space search

- Trees versus Graphs, complexity, states space search, data-driven and goal-driven search, fundamental search algorithms (breadth-first, depth-first, depth-first iterative deepening), heuristic search (A*, iterative deepening A*, beam search, two-person games, mini-max and alpha beta), simulated annealing, genetic algorithms, constraint satisfaction problems.

Knowledge Representation

- Issues in knowledge representation, representation languages, propositional calculus, predicate calculus, semantics networks, conceptual dependencies, frames, scripts.

Inference in First-Order Logic

- Inference, Soundness, Completeness, Unification, Proof by Refuation, Forward Chaining, Backward Chaining, Resolution.

Planning

- Representing Actions, Deriving plans by means-end analysis, Protecting goals, Goal regression, Heurtistics for Planning

Module Assessment

This module should have a 70% weighting for the examination and a 30% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, the course contains a large amount of background and theoretical data and it is important that the student illustrates their understanding of this information. The students' deep understanding of this information is best assessed through examination. Notwithstanding this it is important that the student can apply their knowledge in a practical manner and this ability is best assessed through course work.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Artificial Intelligence 1**
		5	CMPU4010	

Essential Reading

Artificial Intelligence: A Modern Approach, (3rd Ed), 2010, Stuart Russell and Peter Norvig, Pearson

Supplemental Reading

Artificial Intelligence: Structures And Strategies for Complex Problem Solving, (5th Ed), 2005, George F Luger, Addison Wesley

Artificial Intellgience, 2003, Robert Callan, Palgrave.

Prolog Programming for Artificial Intelligence (3rd Edition), 2000, Ivan Bratko, Pearson.

Web references, journals and other

Artifical Intelligence Journal (Elsevier)

Journal of Artificial Intelligence Research (www.jair.org)

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Artificial Intelligence 2**
		5	CMPU4011	

8.4.6 Artificial Intelligence 2

Module Author

John Kelleher and Brian Mac Namee

Module Description

Building on and comparing with the logic-based techniques presented in Artificial Intelligence 1 this module introduces the student to probabilistic techniques and machine learning.

Module Aims

The module aim is provide the student with an understanding of stochastic techniques for reasoning under uncertainty and symbolic and sub-symbolic machine learning techniques. It will provide the students with the fundamental knowledge and skills to work in areas such as data analytics, information retrieval and business systems intelligence.

Learning Outcomes

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

1. Demonstrate an understanding of appropriate techniques for autmoated reasoning in situations of uncertainty.
2. Demonstrate an understanding of the core concepts of machine learning under the headings of (0) represnetations of data, (1) learning using information, (2) memory/instance/analogy based learning, (3) learning using probability and (4) learning by reducing error.
3. Demonstrate and understanding of best practice in evaluating machine learning approaches.
4. Select appropriate machine learning techniques to solve problems in different problem domains.
5. Implement machine learning-based solutions to problems.
6. Discuss the application areas for which machine learning based solutions are appropriate and the limitations of this style of approach.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Artificial Intelligence 2**
		5	CMPU4011	

Learning and Teaching Methods

Lectures will deliver theory and techniques. During practical sessions the students will implement the systems that utilise the techniques presented in the lectures and enable them to directly evaluate the different approaches.

Module content

Reasoning in situations of uncertainty

- Review of counting principles, elements of probability, applications of stochastic technology, Baye's theorem and its use, bayesian belief networks.

Introduction to Machine Learning

- Introduce supervised inductive learning, the curse of dimensionality and feature selection, distinguish between lazy versus eager learners, overview approaches to learning : Probability, Reducing Error, Analogy, Information

Learning using information (entropy based approaches)

- Review information theory and entropy and show how these concepts can be used to develop decision tree classification models using the ID3 algorithm. Introduce students to the concepts and techniques underpinning symbol based learning, including: version space search. Introduce the concepts of inductive bias and learnability. Review the applications domains that information based approaches are suitable to.

Memory/instance/analogy based learning

- Review a range of distance metrics between instances (euclidean/hamming distance, dot product, cosine similarity). Introduce the concept of natural categories. Introduce density estimation and clustering techniques. Introduce K-nn classification. Introduce information retrieval and the PageRank algorithm.

Learning using Probability

- Introduce a range of classification techniques the are based on probability, including: a Bayes optimal classifier, a Maximum A Posterior classifier and a Naïve Bayes classifier. Review the types of applications domains that probability

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Artificial Intelligence 2**
		5	CMPU4011	

based approaches are suitable to.

Learning by reducing error

- Review a range of error functions, including the sum of squared errors. Introduce regression as a technqiues for developing classification models. Introduce the concept linear separability. Review a range of classification techniques based on reducing an error function, including neural networks and support vector machines. Review the types of applications domains that error based approaches are suitable to.

Evaluation

- Introduce concepts relevant to the evaluation of classificaiton models, for example over-fitting. Introduce the best practice in classifier evaluation, including: treatment of danger, avoiding peeking, cross-validation.

Module Assessment

This module should have a 70% weighting for the examination and a 30% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, the course contains a large amount of background and theoretical data and it is important that the student illustrates their understanding of this information. The students' deep understanding of this information is best assessed through examination. Notwithstanding this it is important that the student can apply their knowledge in a practical manner and this ability is best assessed through course work.

Essential Reading

Artificial Intelligence: A Modern Approach, (3rd Ed), 2010, Stuart Russell and Peter Norvig, Pearson

Supplemental Reading

Introduction to Machine Learning (2nd Edition), 2010, Ethem Alpaydin. MIT Press.

Machine Learning, Tom Mitchell, 1997, McGraw-Hill

Artificial Intelligence: Structures And Strategies for Complex Problem Solving, (5th Ed), 2005, George F Luger, Addison Wesley

Pattern Recognition and Machine Learning, 2006, C Bishop, Springer.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Artificial Intelligence 2**
		5	CMPU4011	

Web references, journals and other

Artifical Intelligence Journal (Elsevier)

Journal of Artificial Intelligence Research (www.jair.org)

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Bioinformatics**
		5	CMPU4012	

8.4.7 Bioinformatics

Module Author

Bioinformatics group

Module Description

This course introduces the student Bioinformatics, which uses computer databases to store, retrieve and assist in understanding biological information. Genome-scale sequencing projects have led to an explosion of genetic sequences available for automated analysis. These gene sequences are the codes, which direct the production of proteins that in turn regulate all life processes such as normal organism development and many pathological conditions. The student will be shown how gene expressions lead to the creation of proteins the building blocks of cells which in turn are the foundation of all organisms. Analysis these sequences by micro-array data lead to a much fuller understanding of many biological processes. Students will be introduced to the basic concepts behind Bioinformatics and Computational Biology tools.

Module Aims

The aim is to introduce computer science students a background and better understanding of the process of how genes through the process of regulation result after a number of intermediate stages in the formation of an amino acid sequence which has a direct relationship to the underlying gene base pair sequence. To show how this relationship leads to the inheritance of characteristics and the evolutionary process in general. Emphasise the significance amino acid sequences in the formation of proteins and specifically the regions involved in protein functionality. To develop the theory of how these basic building blocks lead to cell functionality and different biological processes e.g. organism development. To familiarise the student with how data can be generated and obtained, which can then used to develop a further understanding of the aforementioned biological processes.

Learning Outcomes

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

1. Discuss the process of genes, hereditary, gene regulation and

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Bioinformatics**
		5	CMPU4012	

evolution of the genome.

2. Discuss the importance of gene sequences in the genomic process and how it relates to the formation of amino acid sequences.
3. Analyse an amino acid sequence or gene sequence and discuss how it can result in the formation of the primary, secondary and tertiary structures of proteins the basic building blocks of all organisms.
4. Discuss the fundamental principles of cell biology
5. Describe and analyse the principles of micro-arrays and their importance to the field of bioinformatics.
6. Discuss how to obtain and use the freely available micro-array data
7. Develop an understanding of the underlying computational techniques used in bioinformatics.

Learning and Teaching Methods

This module will employ teaching methods and learning situations such as lectures, seminars and tutorials, as well as case studies

Module content

What is meant by the genome, what are genes and how is their expression regulated, what are the main types of genomes and their properties and how the genes establish different physical traits and how mutations result in the evolution of organisms.

How gene sequence produce basic amino acid sequences : the transcription of DNA and the translation of RNA.

How are the basic amino acid sequences converted into function proteins ..

What are cells and their different components. What are the functions of the different components of a cell. How to these components interact to produce cell functionality.

Evaluate the principles of micro-array analysis and discuss how such techniques allow for the observation of the expression of thousands at the same time.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Bioinformatics**
		5	CMPU4012	

What techniques can be applied to micro-array data to remove background noise and thus ensure the micro-array data is more accurate for analysis.

Discuss probability theory and its role in Statistical Inference.

Module Assessment

Continuous assessment (30%)

A report discussing how biological data can be utilised to evaluate biological systems.

Written examination (70%)

A two hour end of semester written exam.

Essential Reading

Alberts Bruce and Martin Raff, 2003 Essential Cell Biology: An introduction to the molecular biology of the cell. Garland Science Textbooks.

Clark, David P. 2005. Molecular Biology, Elsevier Academic Press

Microarray Gene Expression Data Analysis: A Beginner's Guide, Helen C. Causton, John Quackenbush, Alvis Brazma, 2003, Blackwell Publishers

Markley Scott and Leon Darryll Sequence Analysis in a Nutshell: A Guide to Tools A Guide to Common Tools and Databases 2003

Bower, James M and Bolouri Hamid, 2001, Computational modelling of genetic and biochemical networks Cambridge, Massachusetts; London, England: The MIT Press.

Supplemental Reading

To be supplied before commencement of the module

Web references, journals and other

Genetics Virtual Library

Microbiology Resources

Biology related search engine

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Bioinformatics**
		5	CMPU4012	

Gene Expression and Micro-array analysis resources

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Business Systems Intelligence**
		5	CMPU4013	

8.4.8 Business Systems Intelligence

Module Author

Brendan Tierney

Module Description

Business systems intelligence is an area of increasing importance and interest to organisations involved in knowledge management. Technologies such as data warehousing and data mining offer huge potential in the creation of new knowledge products and services and the enhancement of existing products and services. The module builds on the student's previous experiences of working with databases, knowledge tools, techniques and data analysis. This module covers topics in business systems intelligence relating to the formulation of data and business models for understanding data, construction of data warehouses and the application of data mining techniques.

Module Aims

The aim of this module is to study and practise advanced data modelling techniques and to understand and practice techniques of data warehousing and data mining in the context of business systems.

Learning Outcomes

1. On completion of this module, the successful learner will be able to:
2. Discuss how to build a business data model
3. Build a dimensional data model
4. Discuss the role of data warehousing and data mining in an organisation
5. Analyse and evaluate the suitability of different data warehouse architectures
6. Develop dimensional models for a data warehouse
7. Analyse and evaluate the issues involved in extracting and loading data into a data warehouse
8. Discuss the suitability of different data mining techniques

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Business Systems Intelligence**
		5	CMPU4013	

9. Identify the requirements of developing a model for data mining
10. Develop a model for a data mining application
11. Perform different data mining techniques to data
12. Discuss and evaluate the outcomes from a data mining process

Learning and Teaching Methods

This module will employ teaching methods and learning situations in the traditional roles such as lectures, seminars and tutorials, as well as more innovative, student-based learning methods such as problem solving in groups for both theoretical and practical situations.

Students will be encouraged to be pro-active in their approach to learning through the use of case studies and simulation exercises, working independently and in groups. In some cases students will be expected to use computer-based learning material to supplement studies.

The practical element of the module will be supported through the medium of supervised and independent practical sessions. Students will be able to explore the characteristics, advantages and limitations of approaches learnt through their application to suitable case studies and simulation exercises. Where appropriate, students will provide feedback from group research through cascading the knowledge to peers and through presentations. In-class discussions, review of leading research papers in each topic covered will also contribute towards the practical content.

Module content

Business Data Modelling

- Data, Information, Knowledge
- Modelling an activity
- Framing a business model
- Developing a model
- Deploying a model

Data Warehousing

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Business Systems Intelligence**
		5	CMPU4013	

- Introduction to data warehousing
- Characteristics of a data warehouse and how it differs to operational DBs etc
- Extracting and loading data into a data warehouse
- Dimensional modelling
- Data aggregation

Data Mining

- Introduction to data mining and applications of data mining
- Data mining lifecycles
- Data preparation
- Data association techniques
- Data classification techniques
- Data clustering techniques
- Data visualisation
- Data evaluation

Module Assessment

Continuous assessment will comprise 30% of the marks for this module. An end of module examination will comprise the remaining 70%.

Continuous assessment will take the form of exercises and assignments of varying difficulty. Those within the teaching weeks will be designed to consolidate the material delivered during the teaching programme supplemented by self-study. This work may involve investigating business systems topics relevant to business systems intelligence applications using suitable case studies, research papers, simulation exercises or a mixture of these. Such work may be undertaken individually or as part of a group. Where possible and appropriate, contact practical sessions will be used to provide the student with time and assistance in completing continuous assessment work.

The student will undertake a independent practical tasks consisting of one or more of the following, writing a research paper on an

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Business Systems Intelligence**
		5	CMPU4013	

appropriate use of business systems intelligence in a business context, applying techniques covered to a given data set and problem definition, etc..

Essential Reading

Kimball & Ross, 2002, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition) ,WileyHan & Kamber, 2001, Data Mining: Concepts and Techniques, Morgan Kaufmann

Supplemental Reading

Inmon, 2005, Building the Data Warehouse, Hungry Minds.

Michael J. A. Berry, Gordon Linoff, 1997,Data Mining Techniques: For Marketing, Sales, and Customer Support, Wiley.

Richard J. Roiger, Michael Gaetz, 2002, Data mining : a tutorial-based primer, Addison Wesley.

Olivia Parr Rud, 2000, Data mining cookbook : modeling data for marketing, risk and customer relationship management, Wiley.

Web references, journals and other

Oracle Technology Network, http://www.Otn.oracle.com

Database Trends and Applications website, http://www.dbta.com/index.html

Data Mining & Knowledge Discovery, http://www.kdnuggets.com

Kimball, 1997, Dimensional Modelling Manifesto, DBMS Magazine

Fayyad, 1996, From Data Mining to Knowledge Discovery in Data, AI Magazine

DBMS magazine, http://www.ienterprise.com

DMReview, www.dmreview.com

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Compilers and Language Design**
		5	CMPU4016	

8.4.9 Compilers and Language Design

Module Author

Ronan Bradley

Module Description

This module introduces the student to the architecture of a compilers, syntax and semantics analysis, language grammars, automata and regular expression pattern matching, and to compiler implementation techniques.

Module Aims

The aim of this module is to give an introduction of compiler theory and relevant compiler implementation techniques.

Learning Outcomes

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

1. Have a detailed understanding of architecture of a compiler.
2. Have a detailed understanding of Grammar, and Automata.
3. Appreciate the general compiling techniques and algorithms.
4. Gained practical experience in the applying the compiling techniques.

Learning and Teaching Methods

Lectures, self-study, labs, tutorials, and any combination of discussion, case study, problem-solving exercises, readings, seminars, and computer-based learning.

Module content

Complier Theory Overview, Compilers, Translators, Interpreters, Compilation Phases, Lexical, Syntactic, Semantic, Intermediate Representations, Code Generation, Optimisation, Language Issues.

Data Structures, Decision Constructs, Looping Constructs, Procedure calling, Parameter Passing, Object-Oriented Constructs.

Lexical Analysis, Finite State Machines, Regular Expressions. Syntax

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Compilers and Language Design**
		5	CMPU4016	

Analysis, Grammars, Chomsky Hierarchy, Top-Down and Bottom-Up Parsing, Recursive Descent Parsers, Pushdown Parsers, Attributed Grammars and Syntax Directed Translation.

Compiler Tools, Case Studies and Applications

Syntax-Directed Editing, Automated Code Generation.

Turing Machines.

Module Assessment

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one or more of assignment, essay, problem-solving exercise, oral presentation, and class or lab tests.

Continuous Assessment 30%

Examination 70%

Essential Reading

Aho, Sethi and Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1986. ISBN 0-201-10194-7, (better known as The Dragon Book).

Supplemental Reading

Michael L. Scott, Programming Language Pragmatics, Morgan Kaufmann Publishers, 2000.

Andrew W. Appel, Modern Compiler Implementation in C/Java/ML, Cambridge University Press, New York, 1998.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Computer Graphics**
		5	CMPU4017	

8.4.10 Computer Graphics

Module Author

Paul Kelly and Brian Mac Namee.

Module Description

This is an introductory module in computer graphics. This module introduces the learner to the branch of computer science which deals with the theory and technology for computerised image synthesis. The module examines the mathematical and algorithmic basis of graphics concepts and their implementation in a representative interactive graphics system. A major emphasis is placed on the structure and use of a device-independent application programmer's interface (API) to create graphical applications.

Module Aims

The aim of this module is to teach the fundamentals of computer graphics. Its theoretical spectrum is to cover the complete graphics pipeline and include major aspects of computer graphics such as geometrical transformations, projections, surface modelling, illumination and shading, hardware considerations and issues in visual realism. Students should also be made aware of the diverse application areas for graphics technologies. The theoretical content of the course is to be supported with a strong practical element to include both programming and the use of suitable supplementary packages.

Learning Outcomes

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

1. Know the fundamental techniques and algorithms used in computer graphics
2. Apply the mathematics and algorithms behind computer graphics
3. Write software that uses a graphics API
4. Implement some of the techniques taught in the course using appropriate packages
5. Critically analyse the impact of technological developments on the area of visualisation

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Computer Graphics**
		5	CMPU4017	

6. Display an awareness the application areas of computer graphics technologies
7. Select appropriate graphics techniques to meet application requirements (e.g. performance)

Learning and Teaching Methods

Lectures, tutorials, lab demonstrations and discussions. Students will be expected to put into practice some of the module concepts by developing small sample applications. A Virtual Learning Environment is used extensively in this module.

Module content

Graphics I/O and Storage Devices. Scan Conversion. Plotting points, lines, shapes. Filling. Boundary block transfer.

Mathematics of Transformations, Geometric, Coordination, Composite, Instance Transformations, Windows and viewports. Viewing transformations, clipping and shielding. The synthetic camera. Coordinate systems. Hierarchy models. Fractals.

Perspectives and Parallel Projections. 2D and 3D viewing pipelines. Hidden Surfaces. Depth comparisons. Depth buffer algorithm. Scan line algorithms. Painters algorithm. Subdivision algorithms. Hidden line elimination.

Animation. 3-D dynamics, occlusion, solid objects and interference, scene generation. Surface Treatment. Ray tracing, radiosity, rendering. Colour theory.

Visual realism: Illumination. Textures. Shadows. Fog. Reflections. Gourad and Phong shading models. Illumination and reflectance models.

Image File Formats.

Splines, B-splines, Cubic splines.

New and current topics in computer graphics.

Module Assessment

Written examination - 70%

Continuous assessment - 30%

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Computer Graphics**
		5	CMPU4017	

Essential Reading

Francis S. Hill and Stephen M. Kelley,2007, Computer Graphics Using openGL (3rd edition), Prentice-Hall, 2006., or

Donald Hearn, M. Pauline Baker and Warren Carithers, 2010, Computer Graphics with Open GL 4/e, Prentice Hall.

Supplemental Reading

Mason Woo, Jackie Neider, Tom Davis, OpenGL Architecture Review Board , 2008,

OpenGL Programming Guide: The Official Guide To Learning Opengl, Addison Wesley

James Foley, 2005, Computer Graphics: Principles and Practice, Addison Wesley

I. Angel, 2005, Interactive Computer Graphics: A Top-Down Approach Using OpenGL (5th Edition)

Web references, journals and other

OpenGL Home: http://www.opengl.org

OpenGL Programming Guide (The Red Book) :http://www.glprogramming.com/red/

OpenGL Reference :http://www.talisman.org/opengl-1.1/Reference.html

GLUT Reference:http://krow.net/dict/spec3/spec3.html

Nate Robbins' OpenGL tutorials:http://www.xmission.com/%7Enate/tutors.html

NEHE Tutorials: http://nehe.gamedev.net

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Designing and Building Semantic Web Applications**
		5	CMPU4019	

8.4.11 Designing and Building Semantic Web Applications

Module Author

Patrick Browne

Module Description

The Semantic Web is an effort to bring additional meaning (or semantics) to very large body of information that is available on the World Wide Web. The added meaning together with machine reasoning will lead to increased interoperability and integration, better search and improved navigation. Current web technologies support weakly structured information. HTML and CSS focus on presentation rather than the meaning of the content. Semantic integration provides a global view of diverse terms in different data sources. Interoperability provides the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units. It is envisaged that the semantic web will bring technical, social and economical benefits. The course is largely self-contained. However some familiarity with basic programming concepts, database management systems and elementary mathematic would be useful.

Module Aims

The module will provide a theoretical and practical understanding of leading edge solutions for the Semantic Web.

Learning Outcomes

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

1. The technical architecture of the Semantic Web, and its integration with the World Wide Web;
2. Be able to understand and use frameworks for building semantic web applications (e.g. Jena).
3. The underlying knowledge representation formalisms in use on the Semantic Web;
4. Common ontology design patterns;
5. Design and development techniques use for Semantic Web

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Designing and Building Semantic Web Applications**
		5	CMPU4019	

applications.

6. Critically evaluate recent research on the Semantic Web.
7. Apply the conceptual and theoretical elements to Semantic Web applications
8. Relate methodologies, technologies and techniques to a range of practical applications.
9. Use common Semantic Web tools to design, document and verify ontologies.
10. How the semantic relates to the more general IT and social environment.
11. How the SW relates to the smart economy.

Learning and Teaching Methods

Lectures, self-study, labs, tutorials, and any combination of discussion, case study, problem-solving exercises, readings, seminars, and computer-based learning.

Module content

Mathematical background: Elementary concepts about sets, relations, functions, set operations.

Propositional logic. Theory, language, models, validity and satisfiability, inference rules, soundness and completeness, reasoning methods: truth tables, proof techniques.

Predicate logic: First order logic formulae, their meaning, validity and satisfiability, useful normal forms, inference calculus, undecidability and semi-decidability, translating between natural language and first-order logic.

Elementary knowledge representation and formalisms: How symbolic formal systems can be used to represent the real world and our conceptualisation of the real world, including: theories, classes, types, categories, hierarchies, taxonomies, inheritance, lattices, networks, mereology, and frame-based systems.

Modal Logic: Representation and reasoning on the semantic level, possible worlds semantics, model checking, satisfiability and validity, correspondence theory.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Designing and Building Semantic Web Applications**
		5	CMPU4019	

Language of description logics, meaning of description logic statements, reasoning calculi, introduction to the semantic web and ontologies using description logics.

Ontologies: basic concepts, editors forontology construction (e.g. Protégé).

Web services: basic concepts.

How the semantic web relates to databases and legacy systems.

How the semantic web relates to society and the smart economy.

Possible languages include, but are not restricted to, general purposes languages (e.g. Java) and languages specifically designed fro the semantic web (e.g. OWL-DL). Possible software include, but not restricted to, semantic reasoners (e.g. Pellet), ontology editors (e.g. Protégé), semantic APIs (e.g. Jena).

Module Assessment

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one or more of assignment, essay, problem-solving exercise, oral presentation, and class or lab tests.

Continuous Assessment 50%

Examination 50%

Essential Reading

A Semantic Web Primer (2nd Edition), Antoniou and van Harmelen MIT Press, 2009

Supplemental Reading

Knowledge Representation, Sowa, Brooks/Cole 2000.

Foundations of Semantic Web Technologies, Hitzler, Krötzsch, Rudolph, Chapman & Hall/CRC, 2009.

Ontology Alignment, Ehrig, Springer 2007

Web references, journals and other

The Semantic Web. Tim Berners-Lee, James Hendler and Ora Lassila.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Designing and Building Semantic Web Applications**
		5	CMPU4019	

Scientific American, May, 2001.

Standards W3C-OWL-2 (2009). OWL 2 Web Ontology Language Document Overview, W3C.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Digital Audio**
		5	CMPU4020	

8.4.12 Digital Audio

Module Author

Bryan Duggan and Dan Barry

Module Description

Computers have been used for decades to generate process and analyse musical sounds. Their ability to perform these tasks has grown so great that a single computer with appropriate hardware and software can provide the functionality of a traditional recording studio. Digital audio and computer music are now ubiquitous, whether part of a multi-media application, a computer game, the latest singles in the charts or streaming audio over the Internet. The aim of this module is to introduce students to digital audio techniques. This course has a strong practical and technical focus.

Module Aims

The aim of this module is for students to learn how audio is digitised and processed.

Learning Outcomes

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

1. Demonstrate an understanding of trends and advances in digital audio and sampling technology.
2. Demonstrate an understanding of how issues such as sample rate, resolution, the Nyquist frequency and so on affect the quality of sampled audio.
3. Program effects such as vibrato, tremolo and ADSR envelopes using additive synthesis.
4. Analyse sampled audio using a range of tools such as the Fourier Transform.
5. Use a patch diagram to program audio.
6. Demonstrate a basic understanding of music theory.
7. Program simulations of sounds and instruments.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Digital Audio**
		5	CMPU4020	

Learning and Teaching Methods

Class time is split into a series of interactive “studio classroom” based lectures and practical problem solving in labs. In lectures, students have access to a PC, with appropriate software and development kits, so that material and examples can be examined in a live environment. In addition, students will be expected to proactively and independently seek out resources on the internet and from the library to supplement their own learning.

A Virtual Learning Environment (VLE) will be employed to distribute all teaching materials and to support student interaction with both other students and academic staff. Where new material is not presented in lectures, such material will be made available through the VLE and students are expected to proactively use this resource.

Module content

Digital Audio Concepts

- Phasers, basics of sound signals, sampling, resolution, frequency, the Nyquist frequency, compression artefacts.

Introduction to Psychoacoustics in Computer Music

- Perception of intensity, the human ear, perception of temporal features, perception of frequency, perception of noise, fusion and the perception of timbre.

Sound synthesis

- Soundcards and sound production, digital oscillators, FM & AM Synthesis, wavetables, soundcard architectures.

Programming effects

- Tremolo, vibrato, ADSR envelopes, additive synthesis, understanding patch diagrams, analysing signals, Fourier analysis.

Music Theory

- The well tempered scale, major and minor keys, sharps, flats and naturals, the relationship between notes and frequency, time signatures.

Digital audio formats

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Digital Audio**
		5	CMPU4020	

- Compression Principles, Sub-band Coding, Transform Coding, Compression Formats: MPEG Layer I/II/III/2 ACC. MP3, PCM, AU, WAVE, AIFF. Streaming Media.

Module Assessment

This module has a 70% weighting for the examination and a 30% weighting for the continuous assessment. Continuous assessment will consist of assignments and in class examinations. While it is important that students can demonstrate their technical ability through continuous assessment, it is equally important that they demonstrate an understanding of the concepts. This will be assessed in the examination.

Essential Reading

Curtis Roads, (2000), The Computer Music Tutorial, MIT Press

Supplemental Reading

Eduardo Reck Miranda, (2001), Composing Music with Computers, Focal Press.

Richard Boulanger (2000) The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing,and Programming (Paperback)

Web references, journals and other

http://csounds.com

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Distributed Systems**
		5	CMPU4021	

8.4.13 Distributed Systems

Module Author

Edina Hatunic-Webster

Module Description

This module introduces the student to the distributed systems fundamentals and the issues facing their design and implementation. It provides the student with the skills necessary to develop distributed applications. The module also analyses advanced distributed systems issues such as: processes and scheduling, time, synchronisation.

Module Aims

The aims of this module are to:

The module is designed to provide the students with a thorough understanding of issues involved in designing distributed systems. The aim of this module is to give students an understanding of the fundamentals and advanced aspects of distributed systems, and to provide students with the skills necessary to develop distributed systems.

Learning Outcomes

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

1. Discuss authoritatively the fundamental characteristics of distributed systems.
2. Demonstrate an understanding of various distributed system architectures.
3. Discuss how the characteristics of the underlying networks impact the behaviour and design of distributed systems.
4. Compare and contrast the requirements for distributed system development with development in a localised environment.
5. Develop advanced distributed applications using sockets.
6. Develop advanced distributed applications using remote method invocation.
7. Discuss advanced distributed systems issues (e.g. processes and

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Distributed Systems**
		5	CMPU4021	

scheduling, time, synchronisation).

8. Demonstrate an understanding of importance of middleware for distributed systems.
9. Demonstrate an understanding of advanced distributed systems issues.

Learning and Teaching Methods

This module will be taught through a combination of lecture, seminars and practical lab-based sessions.

Supervised practical sessions will be used to complement material presented in lectures. Concepts from lectures will be applied via supervised lab based exercises and practical assignments. It will be expected that all students complete their exercises each week and be pro-active in their learning approach.

Module content

Distributed System Fundamentals: Architectural models for distributed systems; concurrency, failure models, replication, naming and directory services, impact of underlying networks characteristics, internetworking, middleware.

Advanced Distributed Systems Issues: Processes and scheduling, time, security issues and securing distributed systems. Concurrency: synchronization, transaction management.

Distributed objects and remote invocation: Communication between distributed objects, distributed object model, Remote Method Invocation (RMI), events and notifications, web services.

Middleware: Purpose of middleware in distributed systems, middleware platforms.

Java in Distributed Systems: Java interface definitions, Java Remote Method Invocation (RMI), concurrency and multithreading, Java security manager, serialisation.

Module Assessment

Continuous assessment will comprise 30% of the marks for this module. An end of module examination will comprise the remaining 70%.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Distributed Systems**
		5	CMPU4021	

Continuous assessment will take the form of exercises and assignments of varying difficulty. Where possible and appropriate, contact practical sessions will be used to provide the student with time and assistance in completing continuous assessment work.

Essential Reading

George Coulouris, Jean Dollimore and Tim Kindberg (2004), Distributed Systems: Concepts and Designs (4th Edition), Prentice Hall.

M. L. Liu (2003) Distributed Computing: Principles and Applications, Addison Wesley

Supplemental Reading

Relevant supplemental references will be indicated during the teaching of the module.

Web references, journals and other

Relevant web references, journals and other will be indicated during the teaching of the module.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Enterprise Application Development**
		5	CMPU4023	

8.4.14 Enterprise Application Development

Module Author

Michael Collins

Module Description

This module shows how to architect enterprise applications and how to use enterprise-level Java technologies to develop very sophisticated applications. Core design fundamentals such as concurrency, scalability and redundancy will be taught to students and how to incorporate these and other important factors in a multi-tiered web application.

Module Aims

The aims of this module are to:

- Teach the student how to design and develop sophisticated enterprise applications using several Java related technologies.
- Teach correct enterprise application design principles and how to incorporate these in developing a web application.
- Provide the student with sufficient knowledge in recognising common problems in web applications.
- Teach the student how to incorporate persistent data in a multi-tiered enterprise application model.

Learning Outcomes

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

1. Design and develop sophisticated enterprise applications using a variety of Java related technologies.
2. Incorporate efficient design principles in the development of a complex web application.
3. Distinguish the need for specific components in order to implement a design.
4. Implement a sophisticated Graphical User Interface (GUI) of a web application using enterprise-level Java components that are independent of the server-side technologies.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Enterprise Application Development**
		5	CMPU4023	

5. Maintain state in a sophisticated application.
6. Incorporate authentication in a multi-tiered application.
7. Implement session tracking using a variety of techniques and be able to select the most appropriate method based on the design.

Learning and Teaching Methods

This module will teach how to design and develop sophisticated enterprise applications using a mixture of theory and practical work. Lectures will be used to deliver the theoretical material to the student and will include sample code to show the student how the material is implemented. References, documentation and other resources will also be provided to supplement all material delivered. It will be expected that the student conducts self-study outside of normal contact hours to complement this.

Due to the strong practical nature of this module, a large emphasis will be placed on the development of web applications in the laboratory. Each week, the student will be given a practical exercise based on material delivered in their lecture. This will compliment their overall understanding in the respective area.

Module content

Java Servlets

- Review of Servlet APIs, processing HTTP requests, HTTP Authentication with Servlets, Session management, using cookies with Servlets, using the Session API, Database management with Servlets and JDBC, comparing Servlets with other server-side technologies.

JavaServer Faces

- Introduction to JSF, What is JSF, JSF Features, How JSF fits for web applications, the JSF Architecture and design pattern, JSF Components, JSF Tags, JSF validation process, JSF lifecycle.

Enterprise Architectures

- The Model-View-Controller (MVC) design pattern, the role of Enterprise JavaBeans (EJB) in web application development. Java Persistence API in web application development. Setting up an Application Server, establishing a connection to a database.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Enterprise Application Development**
		5	CMPU4023	

Module Assessment

Written examination: 70%

Continuous assessment: 30%

Essential Reading

Hunter and Crawford, 2001, Java Servlet Programming, 2nd Edition, O'Reilly

Larry Brown, Marty Hall, 2006, Core Servlets and JavaServer Pages, 2nd Edition/Volume II, Prentice Hall

Hans Bergsten, 2006, JavaServer Faces, 3rd Edition, O'Reilly

Ian F. Darwin, Jason Brittain, 2003, Tomcat: The Definitive Guide, 1st Edition, O'Reilly and Associates

Vlada Matena and Beth Stearns, 2000, Applying Enterprise JavaBeans, 1st Edition, Prentice Hall.

Supplemental Reading

Bruce Perry, 2004, Java Servlet and JSP Cookbook, 1st Edition, O'Reilly and Associates

Web references, journals and other

World-Wide-Web Consortium http://www.w3.org/

Oracle – Java http://www.oracle.com/technetwork/java/index.html

Apache Group http://jakarta.apache.org

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Enterprise Systems and Architecture**
		5	CMPU4025	

8.4.15 Enterprise Systems and Architecture

Module Author

Ronan Bradley

Module Description

This module covers the range of enterprise architectures, enterprise application and service delivery approaches that are used within organisations. The module also delivers an understanding of the context within which enterprise computing architectures are used in large organisations.

Module Aims

The aim of this module is to provide students with knowledge of the purpose, design and context of enterprise architectures and systems within an organisation.

Learning Outcomes

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

1. Explain the concepts of enterprise wide IT systems.
2. Explain the role of IT within the organisation and the organisational factors that guide IT decisions
3. Classify enterprise systems on the basis of their purpose, architecture and role within the organisation
4. Describe the various processes within the organisation supported by enterprise systems.
5. Describe the primary applications found in an organization.
6. Analyse the role of enterprise architecture
7. Describe common enterprise architectures
8. Analyse and evaluate enterprise integration concepts and architectures
9. Analyse and compare major enterprise integration technologies
10. Compare different service delivery approaches and the business and technology issues associated with these

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Enterprise Systems and Architecture**
		5	CMPU4025	

architectures

Learning and Teaching Methods

This module will employ teaching methods and learning situations such as lectures, seminars and tutorials, as well as case studies and guest lectures where appropriate and available).

Module content

Enterprise Systems in the Organisation

- Role of IT within the enterprise, organisation and responsibilities of IT departments, challenges facing enterprise IT departments, costing and evaluation of enterprise IT projects including use of alternative service models)
- Organisational enterprise processes and IT infrastructure, such as supply chain related processes, finance and human resources.
- Classification of enterprise systems by functionality and technical architecture
- Common types and application areas: Database systems, ERP systems, customer relationship management, supply chain management, decision support systems, Internet/intranet/extranet systems, e-commerce systems. Specific examples of industry enterprise systems.

Enterprise Systems Architectures

- Concepts of an enterprise system architecture
- Legacy systems, multi-tier client-server architectures, distributed systems architecture
- Role of enterprise services such as naming, security, and transaction management

Enterprise Integration Technology

- Integration brokers
- Messaging, middleware and XML
- Application Servers
- Business Process modelling and workflow

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Enterprise Systems and Architecture**
		5	CMPU4025	

Enterprise Integration Architectures

- Enterprise Application Integration
- Service Oriented Architecture

Service delivery models

- Role of Open Source within the enterprise
- Current service delivery models such as cloud computing, out-sourcing and software as a service
- Technical and related business issues associated with service delivery models

Module Assessment

The module will be assessed with 40% continuous assessment and 60% written examination.

Essential Reading

Linthicum, David S. (2003), Next Generation Application Integration: From Simple Information to Web Services. Addison-Wesley Information Technology Series

Lam, Wand (2007), Enterprise Architecture and Integration: Methods, Implementation and Technologies, Information Science Reference

Supplemental Reading

Relevant supplemental references will be indicated during the teaching of the module.

Web references, journals and other

Relevant web references, journals and other will be indicated during the teaching of the module.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Final Year Project**
		20	CMPU4027	

8.4.16 Final Year Project

Module Author

Paul Doyle

Module Description

This module requires students to propose, design, implement, test, document and present a complex software project to demonstrate the level of knowledge gained over the course of their studies. The project must be an independent piece of work which is both coherent and well structured. The student is expected to be self-motivated so as to drive this work through to completion. They are expected to identify the key areas of the project and to make real decisions that will ultimately affect the end deliverable. The project deliverable will be an assessable, independent body of work that will demonstrate the student's ability to work on their own and their ability to communicate key aspects of the project.

Module Aims

The aim of this module is to ensure that the student either successfully complete a large-scale software system or that they can critically research an area of Computing. Completing a large-scale software system includes all aspects of software development including, but not limited to; Analysis, Design, Development, Implementation Integration, Documentation and Maintenance. Critically researching an area of Computing includes the correct use of the Scientific Method; using a well designed, defined and identified research method (qualitative, quantitative or a mixture of both); well-designed data collection and analysis (if applicable) and critical analysis of their research process. In either case the student undertakes a project which follows a prescribed method and requires them to evaluate the outcomes critically.

Learning Outcomes

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

1. Define a problem area and write a project proposal
2. Evaluate similar systems to their proposal identifying a set of clear user requirements

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Final Year Project**
		20	CMPU4027	

3. Undertake research of the problem area to determining the boundaries and scope of the project.
4. Undertake research of the proposed solution to identify appropriate technologies to use
5. Select and implement a formal design methodology.
6. Write a project plan and project manage the project to completion.
7. Develop a fully operational software/hardware system.
8. Critically evaluate the delivered system against the proposed objectives and requirements.
9. Produce a report (in English) of academic quality with appropriate referencing.
10. Present and defend their findings.

Learning and Teaching Methods

The project module involves a combination of seminars, self-paced work, research, and weekly meetings with their assigned supervisor. This is primarily a self-learning module with the supervisor providing guidance and feedback to the student throughout the project. The student is expected to state at the outset the objectives for the project and to meet these objectives throughout the year. The student re also given explicit verbal feedback at the Interim report stage which is a presentation by the student on the work completed so far and the work planned in the future.

Module content

Students are provided with detailed guidelines on the project process. These guidelines detail the requirements of the project in terms of the project content and deliverables and they also provide guidance on the content and format of the Interim Report and Final Project Manual. The guidelines also explain the roles of the key stakeholders including the student, the supervisor, second reader and project co-ordinator. A number of key seminars are also provided on the following topics

- Writing a Proposal
- Preparing for the Interim and final reports

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Final Year Project**
		20	CMPU4027	

- How to write and deliver a project presentation
- Critical Analysis and Evaluation
- Software Research and Design Methodologies

Module Assessment

The module assessment will comprise 100% Continuous Assessment.

There are two key milestones within the project lifecycle at which the student is assessed. At the Interim Report stage the student is assessed on their research findings, system design, testing and evaluation criteria, report writing and project management. The assessment is based on a key set of criteria set-out in the guidelines document.

At the Final Report stage the student is assessed on any additional research undertaken post-design, the system implementation phase, testing and evaluation phase, report writing, communication skills, project management, and the quality, complexity and usability of the delivered system. The assessment is based on a key set of criteria set-out in the guidelines document.

Essential Reading

Paul Doyle, 2010. The Project Guidelines

Kathy Schwalbe, 2005, A Guide to the Project Management Body of Knowledge: Course Technology

Strunk, W. and White, E. B., 2000, The Elements of Style (4th ed.) Longman

Booth, W., Colomb, G.C., Williams, J.M., 2003, The Craft of Research, University of Chicago Press

Dawson, C., 2005, "A Practical Guide to Research Methods: A User-friendly Manual for Mastering Research Techniques and Projects", How To Books

Supplemental Reading

Paul Clements (Author), Rick Kazman (Author), Mark Klein, 2003, Evaluating Software Architectures Methods Studies, Addison-Wesley Professional

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Final Year Project**
		20	CMPU4027	

Web references, journals and other

http://www.comp.dit.ie/fyp

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Forensics**
		5	CMPU4028	

8.4.17 Forensics

Module Author

Fredrick Mtenzi

Module Description

The use of modern technologies has substantially increased in the last few decades. This has led computers, mobile devices and communications systems being able to create and store unprecedented amount of digital information. However, as we use modern technologies, far more information is retained on these devices than most people would realize. The information retained is usually called electronics evidence (e-evidence). To make matters worse rarely are users aware that their activities have left multiple trails of evidence. In most cases users make insufficient or no attempt to delete those trails regardless of how incriminating they may be. Even techno-savvy users who want to go undetected may not be able to completely delete or disguise all trails of their activities or artefacts. And in some cases, deleting evidence may not be possible. Therefore, Forensics provides a new set of technological investigative skills and tools. The number of different cases and crimes found in today's digital world indicate with certainty that Forensics will be needed in most types of investigations. The e-evidence can be used not only to prove straightforward charges such as illegal possession of pirated software, but also to imply motive or intent by forming a "digital profile or dossier" of an individual or the circumstances surrounding a lawsuit or case.

The module aims to develop an understanding of the range of approaches used in computer forensics. This requires an understanding of three phases for recovering evidence from a computer system or storage medium. The three phases are acquiring data; analyzing data and reporting on the analysis. It is necessary that students have a clear understanding of how data is stored on a range of computer systems as well as being able to discuss the relevant legal issues involved in the collection and the documentation of evidence in a computer forensics investigation.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Forensics**
		5	CMPU4028	

Module Aims

The aims of this module are to:

- introduce the students to the principles Forensics,
- give the students a thorough understanding of the techniques involved in collecting, storing and maintaining admissible electronic evidence,
- provide them with an in-depth practical Forensics knowledge in real-life,
- expand the student's ability to analyse computer systems and storage media to enable them to complete a comprehensive investigation of information stored and moved electronically.

Learning Outcomes

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

1. Define and explain the components of Forensics,
2. Discuss, relate and organise the fundamental concepts of Forensics,
3. Critically analyze different aspects of Forensics, Legal and Ethics
4. Experiment and demonstrate ability to use various tools (Commercial and Open Source) in a computer forensic lab
5. Collect admissible e-Evidence from a Computer System or storage media which can be presented in the court.
6. Correctly and completely document a forensic investigation

Learning and Teaching Methods

In this module a number of teaching methods may be employed including lectures, practical sessions, tutorials and case studies. At least one industrial seminar may be arranged. Focus should be placed on empowering the students to develop their skills independently of the presence of a tutor or lecturer.

Module content

Introduction

- Introduction to forensics

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Forensics**
		5	CMPU4028	

Single System Forensic Evidence

- Identifying and Recording Memory and BIOS Information, Forensically imaging drives, recording cryptographic hashes, obscurity methods, general solutions to encryption, wiping

Windows System Analysis

- Windows File Systems (Master Boot Record, FAT File System, NTFS), Recovering Deleted Files, Windows Artefacts, Tracking User Activity (Office Forensics and Web Usage)

Linux System Analysis

- Linux File Systems (ext2, ext3, ext4, ReiserFS, Linux Swap), Linux Analysis

Macintosh System Analysis

- Evolution of MacOS, Mac Disk, Apple Partition Map, Trees and Nodes, Recovering Deleted Files, System Log and Other System Files

Legal and Ethical Issues

- Evidence Handling, Legal Compliance Privacy Issues, Criminal and Civil Courts, Expert Credentials, Rules of Evidence, Documenting the Investigation

Admissibility of Electronic Evidence

- Preparing for e-Evidence collection and preservation

Forensics Tools

- Use of Forensic tools (commercial and open source)

Module Assessment

This module should have a 50% weighting for the examination and a 50% weighting for the continuous assessment. While it is important that the student can demonstrate their technical ability with coursework, it is equally important that they demonstrate an understanding of the theoretical aspects of Forensics.

Essential Reading

Linda Volonino, Reynaldo Anzaldua and Jana Godwin, 2007, Computer Forensics: Principles and Practices, Prentice Hall

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Forensics**
		5	CMPU4028	

Brian Carrier, 2005, File System Forensic Analysis, Addison Wesley

Supplemental Reading

Keith J. Jones, Richard Bejtlich, Curtis W. Rose, 2005, Real Digital Forensics: Computer Security and Incident Response, Addison-Wesley Professional

Warren G. Kruse II, Jay G. Heiser, 2001, Computer Forensics: Incident Response Essentials, Addison-Wesley Professional

E. Casey, 2001, Handbook of Computer Crime Investigation: Forensic Tools & Technology, (Ed.), Academic Press

Web references, journals and other

http://www.forensics.nl/links

http://www.sans.org/rr/

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Engines 1**
		5	CMPU4030	

8.4.18 Games Engines 1

Module Author

Bryan Duggan

Module Description

The purpose of this module is to introduce students to the core concepts required to program 3D game engines. This course combines a foundation in maths and physics programming for 3D games, 3D graphics and artificial intelligence for games. This course has a technical focus and gives students the opportunity to learn practical 3D games development from the ground up. This course explores analytical geometry, linear algebra, matrices, Newtonian physics and quaternions and applies these techniques to problems in game engine programming using standard API's such as DirectX and OpenGL.

Module Aims

The aim of this module is for students to learn the fundamentals of 3D game engine programming.

Learning Outcomes

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

1. Demonstrate an understanding of trends and advances in computer game technology.
2. Solve common problems in computer games such as entity position, orientation and movement by using a variety of mathematical tools.
3. Program a scene in 3D using standard API’s.
4. Represent and program a flexible camera suitable for a variety of game types (FPS and RTS).
5. Make use of a physics engine to perform physics integration
6. Make use of an audio engine to perform functionality such as 3D positional audio and occlusion
7. Apply software design patterns to computer games engines
8. Use UML and object oriented design to model computer games

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Engines 1**
		5	CMPU4030	

engines.

9. Critically analyse how the techniques learned on the course are used in commercial games.

Learning and Teaching Methods

Class time is split into a series of interactive “studio classroom” based lectures and practical problem solving in labs. In lectures, students have access to a PC, with appropriate software and development kits, so that material and examples can be examined in a live environment. In addition, students will be expected to proactively and independently seek out resources on the internet and from the library to supplement their own learning.

A Virtual Learning Environment (VLE) will be employed to distribute all teaching materials and to support student interaction with both other students and academic staff. Where new material is not presented in lectures, such material will be made available through the VLE and students are expected to proactively use this resource.

Module content

Mathematics for games

- Vectors, trigonometry, matrices, translations, rotations, scaling, linear algebra, equations of lines, rays and planes, intersections of lines rays and planes

3D graphics

- Introduction to the game loop, mathematics programming, the rendering pipeline, vertex buffers, texturing, meshes, particle systems, camera models.

Physics Simulation

- Newtonian physics (force, acceleration, velocity, position, mass). Friction. Hamiltonian mechanics (quaternions, torque, angular velocity and acceleration, equations of motion and rotation.

Game Engines

- Asset management, design patterns, programming a flexible camera, UML, object oriented game design.

Middleware

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Engines 1**
		5	CMPU4030	

- Physics - rigid bodies, joints, integration, collision detection. Audio – 3D positional audio, occlusion

Module Assessment

This module has a 50% weighting for the examination and a 50% weighting for the continuous assessment. Continuous assessment will consist of a single significant assignment, which may be individual or team based. Students will be expected to document and present their assignment work in the form of an ePortfolio or a blog. Students will be expected to enter their assignment into national and international competitions such as the XNA Ireland Challenge and the Imagine Cup. While it is important that students can demonstrate their technical ability through continuous assessment, it is equally important that they demonstrate both an understanding of the mathematical concepts and an appreciation for how the techniques learned on the course are used in commercial games. This will be assessed in the examination. Students will not be required to write code in the examination, but will be required to solve problems using the mathematics and physics concepts learned and to critically analyse examples of the techniques learned on the course in commercial games.

Essential Reading

Introduction to 3D Game Programming with DirectX 10, Frank D Luna, Jones & Bartlett Publishers; 1 edition (October 25, 2008)

Game Physics Engine Development, Second Edition: How to Build a Robust Commercial-Grade Physics Engine for your Game, Ian Millington, Morgan Kaufmann; 2 edition (August 6, 2010)

Supplemental Reading

Advanced Animation with DirectX (Focus on Game Development), Jim Adams , Course Technology PTR; 1 edition (May 22, 2003)

Web references, journals and other

http://gamedevelopers.ie

http://gamesfleadh.ie

http://imaginecup.com

http://dreamspark.com

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Engines 1**
		5	CMPU4030	

http://seriousgames.ie

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Engines 2**
		5	CMPU4031	

8.4.19 Games Engines 2

Module Author

Bryan Duggan

Module Description

The purpose of this module is to build on the skills learned in Game Engines 1 so that students can learn how higher order behaviours are implemented in computer games. This course begins by exploring perception and path finding and continues with a comparison of techniques used to implement higher order NPC (Non Player Character) decision making. This course has a strong practical and technical focus and gives students the opportunity to learn practical games development at a low level.

Module Aims

The aim of this module is to build on the knowledge learned in Game Engines 1 so that students can learn how higher order behaviours are implemented in computer games.

Learning Outcomes

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

1. Compare approaches to implementing perception and propose optimised solutions to perception problems.
2. Critically analyse how perception is implemented in commercial computer games.
3. Program simulations using steering behaviours
4. Implement a variety of path following and path finding algorithms.
5. Demonstrate an understanding of the issues involved in practical, optimised path finding.
6. Use UML and object orientated design to model both technical architectures and behaviours in computer games.
7. Propose architectures for NPC higher order decision making.
8. Critically analyse how the techniques learned on the course are

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Engines 2**
		5	CMPU4031	

used in commercial games.

Learning and Teaching Methods

Class time is split into a series of interactive “studio classroom” based lectures and practical problem solving in labs. In lectures, students have access to a PC, with appropriate software and development kits, so that material and examples can be examined in a live environment. In addition, students will be expected to proactively and independently seek out resources on the internet and from the library to supplement their own learning.

A Virtual Learning Environment (VLE) will be employed to distribute all teaching materials and to support student interaction with both other students and academic staff. Where new material is not presented in lectures, such material will be made available through the VLE and students are expected to proactively use this resource.

Module content

Perception

- Audio perception models, visual perception models, algorithms for visual perception, algorithms for intersection of lines, rays and planes.

Steering behaviours

- Unit steering behaviours (seek, pursuit, arrive path following, obstacle avoidance, interpose, hide). Group behaviours (separation, alignment, cohesion, flocking). Combining steering behaviours (weighted sum, prioritised dithering, spatial partitioning).

Path following and path finding

- Graphs, breath first search, depth first search, Djikstra’s shortest path, the A* algorithm, data structures, optimisations, path smoothing.

Higher order NPC decision making

- Finite state machines, concepts and examples, state transition diagrams, the state machine design pattern, goal driven agent architectures, composite goals, sub goals, goal arbitration.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Games Engines 2**
		5	CMPU4031	

Module Assessment

This module has a 50% weighting for the examination and a 50% weighting for the continuous assessment. Continuous assessment will consist of a single significant assignment, which may be individual or team based. Students will be expected to document and present their assignment work in the form of an ePortfolio or a blog. Students will be expected to enter their assignments into national and international competitions such as the XNA Ireland Challenge and the Imagine Cup. While it is important that students can demonstrate their technical ability through continuous assessment, it is equally important that they demonstrate both an understanding of the mathematical concepts and an appreciation for how the techniques learned on the course are used in commercial games. This will be assessed in the examination. Students will not be required to write code in the examination, but will be required to solve problems using the mathematics and physics concepts learned and to critically analyse examples of the techniques learned on the course in commercial games.

Essential Reading

Mat Buckland (2005), Programming Game AI by Example, Worldware Publishing Inc.

Supplemental Reading

AI Game Development: Synthetic Creatures with Learning and Reactive Behaviors, Alex J. Champandard, New Riders Games (December 6, 2003)

Pierre Rautenbach (2008) 3D Games Programming: Using DirectX 10 and Open GL, Cengage Learning Business Press

Web references, journals and other

http://gamedevelopers.ie

http://gamesfleadh.ie

http://imaginecup.com

http://dreamspark.com

http://seriousgames.ie

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Geographic Information Systems**
		5	CMPU4032	

8.4.20 Geographic Information Systems

Module Author

Mark Foley

Module Description

A Geographical Information System (GIS) is a computer system designed to facilitate the collection, management, and analysis of large volumes of geographical knowledge. This module will focus on the fundamental principles of GIS, the practical techniques of implementing a GIS and the creation and use of a GIS for spatial analysis.

Module Aims

The aim of this module is to take students from no prior knowledge of GIS to a position where they (i) understand the role and current state of the art in Geographical Information Systems, (ii) can analyse a problem in GIS and build an appropriate solution and (iii) are familiar with the basic techniques of spatial analysis and modelling.

Learning Outcomes

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

1. Understand the role of GIS and its application in solving practical problems.
2. Understand how geographic data is represented in a computer.
3. Understand the unique character of geographic data and how this is mapped to the real world.
4. Be able to confidently use the market-leading commercial GIS package.
5. Understand the rudiments of spatial databases and when and where to use database technology.
6. Understand how GI data is created and acquired and be aware of possible sources of data.
7. Understand the art and science of cartography and map design.
8. Solve problems in spatial analysis especially in the areas of

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Geographic Information Systems**
		5	CMPU4032	

visualization, query/measurement and design/modelling.

Learning and Teaching Methods

Lectures, self-study, labs, tutorials, and any combination of discussion, case study, problem-solving exercises, readings, seminars, and computer-based learning.

Module content

The module content will include the following topics. Material may be added to or deleted from this list over the lifetime of the module to reflect the changing nature of the relevant technologies.

Principles

- Introduction to GIS and its applications.
- Representing geography and the nature of geographic data.
- Georeferencing.
- The notion of uncertainty in GI data.

Techniques

- GIS software.
- GIS data collection.
- Creating and managing GI data.
- Introduction to distributed and web GIS.

Analysis

- Cartography and map production.
- Visualization of GI data.
- Query and measurement
- Data summary and inference
- Introduction to spatial modelling

There will be also be hands-on work with the leading GIS software packages and students will apply the course material to practical problems and case studies.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Geographic Information Systems**
		5	CMPU4032	

Module Assessment

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one or more of assignment, essay, problem-solving exercise, oral presentation, and class or lab tests.

Continuous Assessment 50%

Examination 50%

Essential Reading

The material listed here represents a sample of possible reading material for the module. The actual reading material will be determined by the lecturer, taking account of availability of new material and changing technologies which could, in turn, determine module content.

1. Main Text

Geographic Information Systems and Science, 3rd ed.; Paul A. Longley, Michael F. Goodchild, David J. Maguire, David W. Rhind; Wiley, 2010; ISBN: 978-0-470-72144-5.

2. Supplementary Reading

Introduction to Geographic Information Systems, 5th ed.; Kang-tsung Chang; McGraw Hill, 2009; ISBN: 978-007-126758-8.

Geospatial Analysis - a comprehensive guide, 2nd ed.; Michael de Smith, Michael F. Goodchild and Paul A. Longley; Winchelsea, 2007; ISBN: 978-1906221-980.

3. Software

Practical hands-on experience using GIS will be provided through the use of a current market-leading GIS software package.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Image Processing**
		5	CMPU4034	

8.4.21 Image Processing

Module Author

Brian MacNamee and Paul Kelly

Module Description

This is a first course in Image Processing. The course begins with low level image processing and progresses its way up to the beginnings of image interpretation. The learner will be required to apply their understanding of the concepts involved through building applications that manipulate images through the use of suitable packages (e.g. Matlab, Scilab or OpenCV).

Module Aims

The aim of this module is to teach the fundamentals of image processing. This will include image representations, fundamental image processing operations, segmentation and the beginnings of image analysis. Its theoretical content is to be supported with a strong practical element that encompasses both programming and the use of suitable packages.

Learning Outcomes

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

1. Demonstrate an understanding of the fundamentals of image processing
2. Implement image processing techniques to solve image processing problems
3. Compare different approaches to solving the same image processing tasks
4. Select appropriate image processing techniques to solve image processing problems.
5. Critically analyse how image processing feeds into the processes of image analysis and understanding
6. Describe the state of the art of image processing in particular application areas

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Image Processing**
		5	CMPU4034	

Learning and Teaching Methods

Lectures. tutorials, lab demonstrations and discussions. Students will be expected to put into practice some of the module concepts by developing sample applications. A Virtual Learning Environment is extensively used in this module.

Module content

The module content will include, but not be limited to (image processing is a fast moving discipline and so new topics will be introduced to the module over time, the following:

- Introduction to image processing
- Image processing hardware considerations
- Image acquisition and representation
- Fundamentals of image processing
    - Point operations
    - Neighbourhood operations
    - Geometric operations
    - Mathematical morphology
    - Frequency domain processing
- Segmentation
    - Thresholding
    - Edge detection
    - Boundary detection
- Image analysis and understanding
    - Template matching
    - Hough transform
    - Shape descriptions

Module Assessment

Written examination - 70%

Continuous assessment - 30%

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Image Processing**
		5	CMPU4034	

Essential Reading

Rafael C. Gonzalez & Richard E. Woods, 2002, Digital Image Processing 2/e, Prentice Hall

Supplemental Reading

David Vernon, 1991, Machine Vision: Automated Visual Inspection and Robot Vision, Prentice Hall. (Available online)

Web references, journals and other

OpenCV wiki pages: http://opencv.willowgarage.com/wiki/

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Language Technology**
		5	CMPU4041	

8.4.22 Language Technology

Module Author

Svetlana Hensman, John Kelleher and Brian Mac Namee

Module Description

This module introduces students to the basic concepts in natural language processing and current trends and techniques for processing language. There is also a practical element of the module with students implementing different aspects of a natural language based system.

Module Aims

The aim of this module is to present students with comprehensive foundation and practical skills in the area of natural language processing. The module introduces the main components of English phonetics, morphology, and part-of-speech parsing, and discusses and evaluates the current best practice approaches. The module also aims to familiarise the students with basics of representing meaning and inference, as well as current techniques for semantic disambiguation and machine translation.

Learning Outcomes

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

1. Demonstrate sufficient understanding of the area of natural language processing
2. Explain the functioning of finite-state automata and their application for morphological parsing
3. Discuss n-grams and their applications
4. Describe the principles behind a part-of-speech parser, and compare and contrast different part-of-speech tagging and parsing approaches
5. Discuss and evaluate variety of approaches to semantic parsing
6. Differentiate among existing word sense disambiguation algorithms

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Language Technology**
		5	CMPU4041	

7. Explain the fundamentals of machine translation applications
8. Implement natural language processing based solutions to solve application problems

Learning and Teaching Methods

This module will be delivered over one semester with two lectures and one practical lab per week.

Lectures will deliver theory and techniques. During practical sessions the students will implement the techniques and test existing language processing software.

Module content

Regular Expressions, Finate-State Automata and Morphology

- Regular expressions; Deterministic and Non-deterministic Finate-State Automata

N-grams

- N-grams; Smoothing; Spelling correction

Part-of-Speech Tagging

- Word classes; Tagsets for English; Rule-based, Stochastic and Transformation-based Part-of-Speech Tagging; Resolving ambiguity

Parsing

- Context-free grammars.Top-down and Bottom-Up Parsing. Parsing with Unification Contraints. Probabilistic Context-Free Grammars. Dependency Grammars.

Semantics and semantic analysis

- Semantics representation; Syntax-Driven Semantic Analysis; Robust Semantic Analysis

Word Sense Disambiguation

- Selectional Restriction-Based Word Sense Disambiguation; Robust Word Sense Disambiguation; Machine-Learning Approaches; Dictionary-Based Approaches

Machine Translation

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Language Technology**
		5	CMPU4041	

- Machine Translation; The Interlingua Idea for Machine Translation; Direct Translation; Using Statistical Techniques for Machine Translation

Module Assessment

This module is assessed by continuous assessment and examination. Continuous assessment will comprise 50% of the marks, and the end of the module examination will comprise the remaining 50%.

Essential Reading

Danial Jurafsky, James H. Martin, 2008, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, 2nd ed.

Supplemental Reading

Chris Manning and Hinrich Schütze, 1999, Foundations of Statistical Natural Language Processing

Web references, journals and other

http://www.aclweb.org

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Music Technology**
		5	CMPU4042	

8.4.23 Music Technology

Module Author

Bryan Duggan and Dan Barry

Module Description

Computers have been used for decades to generate process and analyse musical sounds. Their ability to perform these tasks has grown so great that a single computer with appropriate hardware and software can provide the functionality of a traditional recording studio. Digital audio and computer music are now ubiquitous, whether part of a multi-media application, a computer game, the latest singles in the charts or streaming audio over the Internet. The aim of this module is to introduce students to techniques and applications of digital audio in music technology. The course explores applications such as sequencing and spatial audio using popular toolkits, hardware and API's. This course has a strong practical and technical focus.

Module Aims

The aims of this module are to:

The aim of this module is to introduce students to applications and techniques of digital audio in music technology.

Learning Outcomes

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

1. Demonstrate an understanding of trends and advances in music technology.
2. Produce multi-track music using popular music and digital audio software packages.
3. Demonstrate an understanding of the underlying principles and techniques used by these packages.
4. Apply effects such as panning and reverberation and demonstrate an understanding of how these effects are produced.
5. Develop music and sound applications included MIDI based applications.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Music Technology**
		5	CMPU4042	

6. Demonstrate an understating of the techniques and algorithms employed by 3D positional audio toolkits.
7. Program 3D positional audio using standard API's.
8. Demonstrate a basic understanding of music theory.

Learning and Teaching Methods

Class time is split into a series of interactive "studio classroom" based lectures and practical problem solving in labs. In lectures, students have access to a PC, with appropriate software and development kits, so that material and examples can be examined in a live environment. In addition, students will be expected to proactively and independently seek out resources on the internet and from the library to supplement their own learning. Students will have access to cutting edge audio hardware and will gain hands on experience using the hardware in practical class sessions.

A Virtual Learning Environment (VLE) will be employed to distribute all teaching materials and to support student interaction with both other students and academic staff. Where new material is not presented in lectures, such material will be made available through the VLE and students are expected to proactively use this resource.

Module content

Sequencers Sampled audio, MIDI, quantisation, mixing, plug-in architecture, latency.

Programming Sound Applications and MIDI Programming music, representing rhythm and pitch, wavetables vs FM synthesis. Introduction to MIDI, MIDI Format: tracks, events, timing, channels, instruments, wire protocol, Programming MIDI, MIDI standards.

Audio Effects Theory and Design Understanding Equalisation, Reverberation, Dynamic Range Processors and various other audio FX. The DSP required to implement such effects.

3D Positional Audio Speaker configurations, vectors, virtual 3D space, the listener, the Doppler effect, sound cones, DirectXSound, OpenAL, hardware acceleration, EAX.

Digital Audio and Music Technology Application Areas Impact of technology on music production. Digital music in multi-media, the Internet and streaming audio, computer games.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Music Technology**
		5	CMPU4042	

Module Assessment

This module has a 70% weighting for the examination and a 30% weighting for the continuous assessment. Continuous assessment will consist of assignments and in class examinations. While it is important that students can demonstrate their technical ability through continuous assessment, it is equally important that they demonstrate an understanding of the concepts. This will be assessed in the examination.

Essential Reading

Curtis Roads, (2000), The Computer Music Tutorial, MIT Press,

Supplemental Reading

Eduardo Reck Miranda, (2001), Composing Music with Computers, Focal Press.

Richard Boulanger (2000) The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing,and Programming (Paperback)

Web references, journals and other

http://csounds.com

Journal of Computer Music

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Rich Web Application Technology**
		5	CMPU4043	

8.4.24 Rich Web Application Technology

Module Author

Michael Collins

Module Description

This module teaches how to develop rich web applications with a particular emphasis on client-side development. The module shows how key design characteristics will effect the decision on selecting specific technologies to develop a web application. Advances in underlying Internet frameworks will also be examined and how these can aid in developing sophisticated applications.

Module Aims

The aims of this module are to:

- Teach the student how to develop rich web applications.
- Teach the methods and techniques in selecting appropriate technologies according to specific design characteristics in developing sophisticated web applications.
- Provide the student with sufficient knowledge in updating existing web applications using more appropriate technologies that are specifically designed for certain environments.

Learning Outcomes

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

1. Develop a sophisticated web application using predominantly client-side technologies.
2. Make appropriate decisions in selecting suitable technologies to implement a complex web application.
3. Distinguish the importance of correct design patterns for web applications that involve different components.
4. Analyse existing web applications and be able to identify areas that may be improved using more suitable mechanisms.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Rich Web Application Technology**
		5	CMPU4043	

Learning and Teaching Methods

This module will teach the student how to implement a rich web application correctly. It will incorporate how to correctly design a web application and identify its core entities. Exercises will be given to the student to design and architect different web applications and these will supplement their theoretical knowledge. The content of this module contains a strong practical element that will involve initial theoretical material to be delivered in lectures and subsequent exercises to be completed in the laboratory. These exercises form the primary source of learning for the student so it will be expected that students complete these each week. If an exercise is not completed during scheduled laboratory time, it will be assumed that the student will complete it during their self-study time.

The student will be given many references to books and online material that supplement content delivered during their lectures. This will form part of their self-study to be done outside of normal contact hours.

Module content

Ruby – Using Ruby to develop rich web applications

AJAX – Introduction to AJAX, AJAX and JavaScript, developing applications with AJAX, advantages of AJAX against using other client-side technologies.

Python – Introduction to Python, using Python in rich web application development

Adobe Flex – Developing rich and sophisticated interfaces to web applications, comparison to other similar GUI technologies, XML

Module Assessment

Written examination: 70%

Continuous assessment: 30%

Essential Reading

Dave Thomas, Chad Fowler, Andy Hunt, 2004, Programming Ruby, 2nd Edition, O'Reilly

Paul Barry, 2010, Head First Python, 1st Edition, O'Reilly

Anthony T. Holdener, 2008, AJAX: The Definitive Guide, 1st Edition,

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Rich Web Application Technology**
		5	CMPU4043	

O’Reilly

Yakov Fain, Victor Rasputnis, Anatole Tartakovsky, 2010, Enterprise Development with Flex, 1st Edition, O'Reilly

Supplemental Reading

Ian F. Darwin, Jason Brittain, 2003, Tomcat: The Definitive Guide, 1st Edition, O’Reilly and Associates

Alaric Cole, Elijah Robison, 2010, Learning Flex 4, 1st Edition, O'Reilly

Simson Garfinkel & Gene Spafford, 2001, Web Security, Privacy & Commerce, 2nd Edition, O’Reilly and Associates

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Quality Management**
		5	CMPU4045	

8.4.25 Software Quality Management

Module Author

Ronan Fitzpatrick

Module Description

Modern business models are increasingly being shaped by Information Systems which have at their core quality software products. These quality products are critical to competitive advantage. At the same time software owners are now accountable in law for the impact that these products can have on their employees, clients and customers. Organisations need to know that the products that they are developing or acquiring are technically sufficient, legislation compliant, acceptable to the user community and are developed by domain specialists who understand their business model.

Module Aims

The aim of this module is to equip professionals with the core knowledge necessary for specifying, selecting and acquiring quality software products.

Learning Outcomes

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

1. Discuss the strategic significance of quality software products
2. Explain the principles of quality engineering and its strategic importance
3. Specify software quality requirements, manage the quality engineering and quality assurance processes, evaluate quality software products
4. Clarify and critique the relevance of Quality accreditation systems, certification systems and International standards

Learning and Teaching Methods

Lectures, self-study, tutorials, and any combination of discussion, case study, problem-solving exercises, readings, seminars, and computer-based learning.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Quality Management**
		5	CMPU4045	

Module content

The business case for software quality

Software quality models

External/Internal quality factors and their attributes

Quality in software process models

Physical and cognitive quality

Software quality engineering

Managing for software quality

Software measurement and quality metrics

Quality assurance activities and practices

Quality accreditation and certification systems

International standards

Module Assessment

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one or more of assignment, essay, problem-solving exercise, oral presentation, and class or lab tests.

Continuous Assessment20%

Examination 80%

Essential Reading

McCall, J., Richards, P. and Walters, G (1977) Factors in software quality, Vols I-III, Rome Aid Defence Centre, Italy.

Boehm, B. (1978) Characteristics of Software Quality, North Holland, with J.R. Brown, H. Kaspar, M. Lipow, G. McLeod, and M. Merritt.

Pressman. R, (2000) Adapted by Darrel Ince, Software Engineering: A Practitioner’s Approach (European edition) McGraw-Hill Internationsl, Berkshire, UK

Supplemental Reading

Kan, S. (2002) Metrics and models in Software Quality Engineering, Addison-Wesley

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Software Quality Management**
		5	CMPU4045	

Boehm, B.; See extensive list of readings

Fitzpatrick, R. (1996) Software Quality: Definitions and Strategic Issues, Staffordshire University, School of Computing Report

ISO 9000 international standards

Web references, journals and other

Scholarly articles for Nielsen, J. http://www.google.com/scholar?q=nielsen,+J&hl=en&lr=&rls=GGLC,GGLC:1970-01,GGLC:en&sa=N&tab=ss&oi=scholart

http://www.comp.dit.ie/rfitzpatrick/papers/quality01.pdf

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Spatial Databases**
		5	CMPU4046	

8.4.26 Spatial Databases

Module Author

Patrick Browne

Module Description

This module focuses on the use of database management systems (DBMS) to manage spatial information. A Spatial Database is a specialised DBMS that is designed to facilitate the collection, querying, management, and analysis of large volumes of geographical knowledge. A spatially enabled DBMS is a central component of a Geographical Information System (GIS). Spatial databases have a major role to play in managing the national physical and informational infrastructure. The understanding of advanced spatially enabled DBMS is vital in implementing any information system where geographic data is required. This module focuses on the role of the DBMS in geographical applications.

Module Aims

The aim of this module is to familiarise students with the features of a spatially enabled database systems and to develop applications that use the spatial database extensions.

Learning Outcomes

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

1. use a database to store and query spatial data
2. develop applications that use a spatially enabled DBMS, e.g. spatial data mining, navigation.
3. distinguish and use appropriate database models
4. apply the various query languages appropriate to spatial querying to given situations
5. understand the extensions required by application programs to handle spatial data

Learning and Teaching Methods

Lectures, self-study, labs, tutorials, and any combination of discussion,

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Spatial Databases**
		5	CMPU4046	

case study, problem-solving exercises, readings, seminars, and computer-based learning.

Module content

Foundations Fundamental geographic concepts for GIS The world in spatial terms, how natural and man made features can be stored in a DBMS. Qualitative and quantitative location e.g. coordinate systems and geo-referencing. Maps as representation of the world and of information. Geometric and thematic information.

Algorithms and Data Structures for GIS: Abstract Data Type (ADT), spatial ADTs and their operations e.g. intersection, overlay, network traversal, auto-correlation, statistical operations, searching.

Spatial representations: ADTs for representation, raster, vector, TIN, quadtrees, R-trees, scan orders, polygon coverage, discrete objects, networks, time, connections and topology, networks, distance and direction, flow and diffusion, spatial hierarchies, boundaries, spatial patterns, attributes, and representions of relationships.

Applications of spatial databases: Spatial data mining, spatial data on the World Wide Web, transportation networks, natural resources, soil data, oceanography, land cover, geology, climate, terrain, land records, administrative boundary data, demographic studies, and health data.

Spatial databases Spatial data: definitions, formats, models, queries the relational model, advanced SQL, data modelling techniques, implementing a simple database, post relational database models, object-relational and object-oriented models, spatial data structures, spatial indexing e.g. R-Tree, networking, database issues in GIS.

The course will involve practical work on a range of appropriate software e.g. spatially extended DBMS ,map servers, statisital packages, markup languages, client and server side programming..

Module Assessment

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one or more of assignment, essay, problem-solving exercise, oral presentation, and class or lab tests.

Continuous Assessment 50%

Examination 50%

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Spatial Databases**
		5	CMPU4046	

Essential Reading

Shekhar and Chawla; Spatial databases – A tour, 2003, Prentice Hall, 2003

Supplemental Reading

Obe, Hsu; PostGIS in Action, 2010 Manning.

Rigaux, Scholl, Voisard; Spatial databases, 2002, Morgan Kaufmann

Web references, journals and other

The Open Geospatial Consortium, Inc. (OGC) - http://www.opengeospatial.org/

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Spatial Statistics and Spatial Knowledge Discovery**
		5	CMPU4047	

8.4.27 Spatial Statistics and Spatial Knowledge Discovery

Module Author

Patrick Browne

Module Description

There are vast amounts of spatially related data available from government departments (e.g. CSO, agriculture, environment), local authorities, health boards, and private industry. This module studies techniques to analyse large and diverse data sets with a view to gleaning new and useful information. We use two closely related approaches. Firstly, we study how basic statistical techniques such a correlation and regression can be adapted to handle spatial data. Secondly, we study how basic knowledge discovery techniques, such as association rules, can be used in location based analysis.

Module Aims

The module aims to equip the student with the necessary skills to extract decision support information from large datasets using statistical and knowledge discovery techniques. In order to analyse large spatia data sets the student will study these techniques and the appropriate software.

Learning Outcomes

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

1. use basic descriptive statistics to describe spatial data.
2. use inferential statistics and probability to help make inferences, judgments, and decisions.
3. use statistical packages to analyse spatial data
4. use data mining software to assist in knowledge discovery.
5. use data mining and statistical software for decision support.

Learning and Teaching Methods

Lectures, self-study, labs, tutorials, and any combination of discussion, case study, problem-solving exercises, readings, seminars, and computer-based learning.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Spatial Statistics and Spatial Knowledge Discovery**
		5	CMPU4047	

Module content

Basic statistics and probability (no previous knowledge expected).

Correlation and regression

Spatial autocorrelation and spatial regression.

Association rules, and other techniques for data mining.

Techniques for analysing spatial data sets.

The use of statistical packages for spatial analysis.

The use of data mining software in a spatial context.

Module Assessment

Continuous Assessment 50%

Examination 50%

Essential Reading

Spatial Data Analysis: Lloyd: Oxford University Press, 2009

Spatial Databases – A tour, Shekhar, Chawla, 2003, Prentice Hall.

Supplemental Reading

Applied Spatial Data Analysis with R: Roger S. Bivand, Edzer J. Pebesma and V. Gómez-Rubio

Geographic Data Mining and Knowledge Discovery, Second Edition Harvey J. Miller

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Strategic Management**
		5	CMPU4048	

8.4.28 Strategic Management

Module Author

Mark Foley

Module Description

Strategic management is the process of specifying an organisation's objectives, developing policies and plans to achieve these objectives, and allocating resources so as to implement the plans. It is the highest level of managerial activity, usually performed by the company's Chief Executive Officer (CEO) and executive team. It provides overall direction to the whole enterprise. An organisation's strategy must be appropriate for its resources, circumstances, and objectives. The process involves matching the companies' strategic advantages to the business environment the organisation faces. One objective of an overall corporate strategy is to put the organisation into a position to carry out its mission effectively and efficiently. A good corporate strategy should integrate an organisation's goals, policies, and action sequences (tactics) into a cohesive whole.

Module Aims

Strategic management can be seen as a combination of strategy formulation and strategy implementation. The aim of this module is to provide the student with the knowledge and skills to understand the strategy process both the formulation of the strategy and the implementation of it. The student will develop the skills necessary to understand how strategy is applied in an organisation.

Learning Outcomes

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

1. Perform a situation analysis.
2. Set objectives. This involves crafting vision statements (long term view of a possible future), mission statements (the role that the organisation gives itself in society), overall corporate objectives (both financial and strategic), strategic business unit objectives (both financial and strategic), and tactical objectives.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Strategic Management**
		5	CMPU4048	

3. Create a strategic plan. The plan provides the details of how to achieve these objectives.
4. Understand allocation of sufficient resources (financial, personnel, time, computer system support)
5. Understand how to establish a chain of command or some alternative structure (such as cross functional teams)
6. Understand how to assign responsibility of specific tasks or processes to specific individuals or groups
7. Understand how to manage the process. This includes monitoring results, comparing to benchmarks and best practices, evaluating the efficacy and efficiency of the process, controlling for variances, and making adjustments to the process as necessary.
8. Understand how to implement specific programs, this involves acquiring the requisite resources, developing the process, training, process testing, documentation, and integration with (and/or conversion from) legacy processes.

Learning and Teaching Methods

Lectures, self-study, labs, tutorials, and any combination of discussion, case study, problem-solving exercises, readings, seminars, and computer-based learning.

Module content

Introduction to strategy and the Strategy process

The Strategic Position

Strategic Change

Strategy into Action

How Strategy Develops

Module Assessment

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one or more of assignment, essay, problem-solving exercise, oral presentation, and class or lab tests.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Strategic Management**
		5	CMPU4048	

Continuous Assessment 20%

Examination 80%

Essential Reading

The material listed here represents a sample of possible reading material for the module. The actual reading material will be determined by the lecturer, taking account of availability of new material and changing technologies which could, in turn, determine module content.

Main Text

Strategic Management Concepts, 10e Fred R. David, Prentice Hall, ISBN 0-13-150346-4

Supplemental Reading

Exploring Corporate Strategy, 7e, Johnson, Scholes and Whittington, Prentice Hall, ISBN 0-273-68739-5

The Strategy Process, Mintzberg, Lampel, Quinn & Ghoshal, Prentice Hall, 2003, ISBN 0-13-122790-4

Management Information Systems, 7e, Laudon & Laudon, Prentice Hall, 2002, ISBN 0- 13-061960-4

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software**
		5	CMPU4051	

8.4.29 Systems Software

Module Author

Paul Kelly

Module Description

This module introduces the students' to the various components of the UNIX/Linux operating systems from a system programmer’s perspective including both the shell and programming interfaces.

In this course, students will learn to develop software using the C programming language in the UNIX programming environment. Topics covered will include the user/kernel interface, fundamental concepts of UNIX, advanced I/O, filesystems, device drivers and processes. Fundamental concepts of software development and maintenance on UNIX systems will also be covered.

Students are expected to have a good working knowledge of the C programming language and to be able to competently use a UNIX system with a command-line shell interface.

Module Aims

The aims of this module are to provide the student with skills in the advanced concepts, structures, mechanisms and techniques of UNIX systems programming.

Learning Outcomes

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

1. describe and employ the fundamental concepts, structures, mechanisms of systems libraries and calls of UNIX-based systems programming,
2. use the UNIX tools in developing software in C, including gcc, gdb, ddd, gprof, cvs, make
3. use signals at the command level and as part of a program
4. write software using inter-process communication (IPC) and appropriate system calls
5. program terminal I/O and relevant system calls

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software**
		5	CMPU4051	

6. write concurrent programs using processes and threads

Learning and Teaching Methods

Lectures with demonstrations

Laboratory practicals based on lectures

Extensive use of a Virtual Learning Environment (VLE)

Module content

Software development tools in UNIX/Linux

Advanced scripting techniques

File Systems: File and directory structures, Permissions, Sequential and random file access, Accessing directories, I/O redirections

Processes: Process model, Process environment, Process creation and termination, Process control, Process times

Race conditions and deadlocks

Daemons: Characteristics, Coding, Error logging, Client-server model

Design and implementation of a UNIX-oriented Shell

Signals: concepts, Catching and handling signals, Signal system calls

Interprocess communication: Process synchronisation and communication concepts

Pipes: Programming Concepts, Limitations, Named pipes (FIFOs), Semaphores, Shared memory

Socket programming: APIs and their implementation.

Terminal I/O: Getting and setting terminal attributes, Canonical and non-canonical modes, Nonblocking I/O, Pseudo terminals

Advanced I/O: Record locking, Streams, I/O Multiplexing, Asynchronous I/O, Memory Mapped I/O

POSIX Threads: Concepts, Thread environment, Thread invocation and synchronisation

Module Assessment

Written Examination
- 70%

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software**
		5	CMPU4051	

Programming assignments, written tests, laboratory assessments, on-line tests - 30%

Essential Reading

K. Haviland, D. Gray, B. Salama (1999), Unix System Programming, Addison-Wesley.

Supplemental Reading

R. Stevens, S.A. Rago (2005), Advanced Programming in the UNIX Environment, Second Edition, Addison-Wesley.Maurice

Bach (1986), The Design of the UNIX operating System, Prentice Hall.

S. Sarwar, R. Koretsky, S.Sarwar (2002), Linux: The Textbook , Addison-Wesley.

K. Robbins, S. Robbins (2003), UNIX Systems Programming: Concurrency, Communication, and Threads, Prentice Hall.

Web references, journals and other

The Open Group Base Specifications Issue 6 IEEE Std 1003.1-2001: http://www.opengroup.org/onlinepubs/009695399/mindex.html

GNU Online Documentation available online at: http://www.gnu.org/manual/manual.html

Further current web links as listed on VLE

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software for Embedded Mobile Devices**
		5	CMPU4052	

8.4.30 Systems Software for Embedded Mobile Devices

Module Author

Frank Duignan, Richard Hayes

Module Description

This module is aimed at those who wish to work in the area of developing consumer electronic devices, toys, mobile devices and embedded systems in general. Students will learn about the components typically found in mobile devices such as CPU's, displays, audio codecs and storage. They will study the system software which is required to manage this hardware and also provide a high level, abstract interface for application software.

Module Aims

The aim of this module is to prepare students for employment in the field of mobile and embedded device development. Specifically, the module will deal with mobile devices based on the ARM family of micropocessors such as phones, media consumption devices and embedded systems generally.

Learning Outcomes

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

1. Design and implement programs to control common peripherals such as displays, serial and analogue interfaces.
2. Devise and use testing and debugging methods for embedded systems
3. Develop solutions that will work within the limited power and processing budgets of mobile devices.
4. Develop a simple application that will work with a mobile operating system.
5. Draw up system requirement specifications for embedded applications.

Learning and Teaching Methods

Lectures, laboratory work, self-learning.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software for Embedded Mobile Devices**
		5	CMPU4052	

Module content

Lecture material

1. Outline of components typically found in mobile media devices.
2. The ARM processor family
3. The microprocessor system bus (address/data/control bus timing and interfacing)
4. Interfacing to peripherals using high speed serial protocols
5. Digital audio technology
6. Display technology
7. Structured software design for mobile devices
8. The role of the operating system.
9. Resource and power management for embedded systems.

Laboratory programme

- The ARM development toolchain.
- Digital input/output
- High speed serial communications (SPI)
- Audio input and output.
- Interfacing to an LCD display.
- Interfacing to storage devices.
- Software development for a mobile operating system.

Module Assessment

There are two main sections to the module mark:

Section 1:

- Two online tests. Each of these accounts for 12.5% of the total module mark.
- Laboratory reports account for 10% of the module mark.
- Interview relating to laboratory work accounts for 15% of the module mark.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software for Embedded Mobile Devices**
		5	CMPU4052	

Section 2:

- The remaining 50% of the module marks are assigned to an end of module written exam. Students are expected to answer 4 questions out of a total of 6.

Section 1 accounts for 50% of the total module mark.

Similarly, Section 2 accounts for 50% of the total module mark.

Essential Reading

Sloss, Symes and Wright, 2004, ARM System Developer's Guide – designing and optimizing system software, Elsevier

Supplemental Reading

Yiu, Joseph, 2009, Microprocessor Systems The Definitive Guide to the ARM Cortex-M3, Newnes

Web references, journals and other

Manufacturer data sheets and user guides will be provided with course material

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Universal Design and Web Accessibility**
		5	CMPU4054	

8.4.31 Universal Design and Web Accessibility

Module Author

John Gilligan, Dave Carroll, Damian Gordon, Ciarán O'Leary

Module Description

This modules centres on the principles and practice web accessibility - whose aim is to ensure that as wide range a range of users as possible can access the web-based resources, including users with disabilities. More specifically, web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. The module looks at web accessibility from the general perspective of User Interfaces and its relationship with HCI.

Module Aims

The aims of this module are to:

The aim of this module is to develop a on the foundation in the key tools and technologies used in incorporating accessible design in developing web-based applications and to provide an opportunity for students to gain practical experience of using these in order to develop some of the key skills required to develop such applications.

Learning Outcomes

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

1. Explain Universal Design and the general context of User Interfaces and HCI
2. Demonstrate their understanding of principles of User-Centred Design as an enabler of Universal Design and Accessibility
3. Discuss the difficulty and complexity of the task of Accessible Web Design
4. Select and employ the appropriate WCAG Guidelines for Accessible Web Design
5. mplement software interfaces incorporating principles of Accessible Web Design
6. Implement software interfaces incorporating accessible

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Universal Design and Web Accessibility**
		5	CMPU4054	

frames

7. Illustrate the importance of Accessible Web Design
8. Select and employ the appropriate tools for testing accessibility
9. Compare and contrast different Web Development tools for accessibility

Learning and Teaching Methods

This module will be delivered using a combination of lectures, practicals, case studies, role playing, discussions, and thinking skills. The main theoretical material will be delivered in lectures. The practical work to assess the accessibility of software, and design and implement accessible software interfaces evaluate accessibility, and inclusivity of Web sites will be carried out in laboratory sessions.

Module content

User Interfaces and HCI

- Introduction to User interfaces, Human Computer Interaction (HCI) and Universal Design
- Interactive systems: user control and input / system reaction and output
- Enhanced models of HCI, e.g. adaptive, context-aware systems, multi-modality

Role of assistive technology,

- Alternative formats and interaction modes:
- Role of adaptable and adaptive user interfaces and content:
- Examples of exclusionary UI design vs. inclusive UI design:

Methods how to achieve Universal Design,

- “User-Centred Design” and “Evaluation”.

W3C/WAI Standards

- Web Content Accessibility Guidelines 1.0 and 2.0
- WAI-ARIA, the Accessible Rich Internet Applications Guidelines

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Universal Design and Web Accessibility**
		5	CMPU4054	

Creating Accessible Content and Accessible Navigation

- Navigation Challenges
- Creating Accessible Frames
- Accessible Image Maps

Testing Accessibility

- Online testing
- Downloadable programs
- Web Development Tools and Accessibility

Examples of web tools

- Accessibility features
- Emerging Technologies

Module Assessment

Continuous assessment will comprise 50% of the marks for this module. An end of module examination will comprise the remaining 50%.

Continuous assessment will take the form of exercises and assignments of varying difficulty. Those within the teaching weeks will be designed to consolidate the material delivered during the teaching programme supplemented by self-study. This work may involve writing a program, formalizing problems, using reasoning tools for solving such problems, a case study on some research in one of the areas, or a mixture of these. Where possible and appropriate, contact practical sessions will be used to provide the Student with time and assistance in completing continuous assessment work.

Essential Reading

Cook and Hussey (2007), Assistive Technology Principles and Practice, (3rd Ed), Mosby.

Supplemental Reading

Scherer, Marcia J. (2003) Connecting To Learn, Educational and Assistive Technology for People With Disabilities, American Psychology Association

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Universal Design and Web Accessibility**
		5	CMPU4054	

Nielsen, J., 1999, Designing Web Usability : The Practice of Simplicity, New Riders Press

Web references, journals and other

National Disability Authority www.nda.ie

Centre for Excellence in UD www.ceud.ie

Web Access Initiative www.w3.org/wai

Higher Education Disability Site www.Ahead.ie

Central Remedial Clinic www.crc.ie

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Mapping**
		5	CMPU4055	

8.4.32 Web Mapping

Module Author

Mark Foley

Module Description

Web mapping is the process of designing, implementing, generating and delivering maps on the World Wide Web. This module essentially covers two areas – (i) basic web development and (ii) the aspects of web development specific to GIS. No prior knowledge of web development is assumed.

The first objective of the module is to teach students the skills required to develop a simple three tier web application incorporating client side technologies such as XHTML and CSS, server side scripts written using scripting languages such as PHP and possibly a back end database.

The second objective is build on the basic web development skills to add mapping and GIS elements to web applications. This is achieved using client-side technologies (Javascript frameworks such as OpenLayers and the Google Maps API) and server-side technologies (map servers such as Geoserver and UNM Mapserver).

Other open-source GIS technologies are also introduced.

Module Aims

The aims of this module are to:

- Enable the student to become familiar with XHTML, CSS, JavaScript, PHP and databases in web applications.
- Present the student with opportunities to develop applications using a variety of technologies.
- Enable the student to understand the nature of geographic data and how this may be stored and accessed via the web.
- Teach the student to use various technologies to develop web mapping applications.
- Teach the students to understand the role of and use of standards in web mapping and GIS generally.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Mapping**
		5	CMPU4055	

Learning Outcomes

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

1. Develop web applications incorporating a variety of technologies.
2. Incorporate mapping in web applications.
3. Choose and use the appropriate set of technologies for any given project.
4. Understand and incorporate various storage technologies and formats in a web mapping project.
5. Understand and use appropriate standards in a web mapping project.

Learning and Teaching Methods

Lectures, self-study, labs, tutorials, and any combination of discussion, case study, problem-solving exercises, readings, seminars, and computer-based learning.

Module content

The module content will include the following topics. Material may be added to or deleted from this list over the lifetime of the module to reflect the changing nature of the relevant technologies.

- Internet/Web architecture.
- XHTML & CSS.
- File formats and storage techniques for GI data.
- Client-side scripting
- Javascript frameworks for web mapping.
- 3-Tier Web Applications.
- Server-side technologies for web mapping.
- Web services (especially OGC standards such as WMS, WFS etc.)
- Introduction to scripting languages.
- Connecting to back-end databases.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Web Mapping**
		5	CMPU4055	

Module Assessment

The methods of assessment to be used to measure the learning objectives stated above are written examination and continuous assessment including one or more of assignment, essay, problem-solving exercise, oral presentation, and class or lab tests.

Continuous Assessment 70%

Examination 30%

Essential Reading

The material listed here represents a sample of possible reading material for the module. The actual reading material will be determined by the lecturer, taking account of availability of new material and changing technologies which could, in turn, determine module content.

Main Texts

HTML, XHTML, and CSS, Sixth Edition: Visual QuickStart Guide, 6th ed.; Elizabeth Castro; Peachpit Press, 2007; ISBN: 0321430840.

JavaScript and Ajax for the Web: Visual QuickStart Guide, 7th ed.; Tom Negrino, Dori Smith; Peachpit Press, 2009; ISBN: 0321564081.

GIS for Web Developers: Adding ’Where’ to Your Web Applications; Scott Davis; ISBN: 978-0974514093

Supplemental Reading

The Web Wizard’s Guide to PHP; David Lash; Prentice Hall, 2003; ISBN: 0321121740.

Mapping Hacks: Tips & Tools for Electronic Cartography; Erle et al.; ISBN: 978-0596007034.

Desktop GIS: Mapping the Planet with Open Source; Sherman; ISBN: 978-1934356067.

Beginning MapServer: Open Source GIS Development; Kropla; ISBN: 978-1590594902.

Web Mapping Illustrated: Using Open Source GIS Toolkits; Mitchell; ISBN: 978-0596008659.