Student Handbook

BSc (Honours) Computer Science

Year 1

DT228-1

Academic Year 2016-2017

Table of Contents

Welcome to DT228-1

On behalf of the teaching team, we would like to warmly welcome you to year 1 of DT228 – the Bachelor of Science in Computer Science at Dublin Institute of Technology. This is a four year programme in Computer Science which produces graduates who work on the information technology sectors in Ireland and internationally. Over the past decade this programme has produced highly successful graduates, many of who have now established their own businesses.

DIT is one of the largest Higher Education Institutions in Ireland, and has a history dating back to the nineteenth century. DIT has a proud tradition of providing technical education in the centre of Dublin – which it continues to do across its campuses at Kevin St, Cathal Brugha St, Bolton St, Aungier St and Rathmines. – and in its new campus at Grangegorman.

Throughout this handbook you will get detailed information on your modules, your assessment, your programme and your life in DIT. You can also get a great deal of additional information through the DIT website and from your year mentor.

Year 1 of this programme is comprised of 11 modules, the majority of which are one semester long. One semester modules last 13 weeks and are followed by examinations. You are required to pass all modules in order to progress to year 2. Throughout year 1 you will master the fundamentals of the discipline, by taking modules such as Programming and IT Fundamentals. You will get an opportunity to work with your classmates on group assignments and will develop your communications skills through presentations – on modules such as Communications for Computer Technologists.

In year 2, you will deepen your knowledge and increase your skills, leading to year 3 where you will have an opportunity to undertake work placement, and year 4 where you will complete a final year project, finish your degree, and begin a rewarding career.

We wish you the best of luck with your studies, helping you to reach your true potential at third level

Dr. Susan McKeever (Programme Chairperson) and Dr. Michael Collins (Year Mentor)

1. My Year

1.1 Registration

You are required to register before you commence our studies. Once you register, you will be provided with a student card and a student number, as well as login details for your e-mail account and other DIT systems – including webcourses (the online system you can use to access many of your notes and submit your assignments). Information on how to register, obtain a student card and pay fees is available online at http://www.dit.ie/registration/.

1.2 Contact Details

Your programme chairperson is Dr. Susan McKeever. She is responsible for managing the overall programme and co-ordinating the programme committee (which consists of lecturing staff and students). You should contact her if you need advice about the overall programme, the structure of the programme or your progression on your modules.

Your class or year mentor is Dr. Michael Collins. He acts as a contact point if you have any queries relating to your modules, any medical circumstances or any other particular queries relating to the course.

Dr Susan McKeever
susan.mckeever@dit.ie
Room: KE 201

Dr Michael Collins
michael.collins@dit.ie
Room: KE 201

Your lecturers and their contact details are listed further on in this document.

The School Office for the School of Computing is located on the third floor of the Annexe in Kevin St, *beside the entrance to the bridge*.

1.3 The Building

Rooms are identified by a combination of letters and numbers e.g. KE1-008 (this room is in the Kevin street Main Building [KE], on the first floor [1]). The Kevin St campus has two buildings – the Annex (room numbers start with KA) and the Main Building (room numbers start with KE).

All of your lectures and most of your lab sessions will be held in Kevin St. Some of you will also use the lab rooms in Aungier st, where the School of Computing labs are located on the first floor: 1-005 and 1-006.

1.4 Timetable and Calendar

You can access your personal timetable online at http://www.dit.ie/timetables/. A guide to using the timetabling system is available at http://www.dit.ie/registration/studentclasstimetables/.

The DIT academic calendar is available online at http://www.dit.ie/academicaffairsandregistrar/calendar/ . The year is divided into two semesters. Each semester is comprised of 13 weeks of classes followed by examinations. There is one review week in each semester, which is a week specially reserved for assessment and revision. Semester 1 runs from September to January. Semester 2 runs from January to May.

Note that the online timetabling system uses a different week numbering system to the academic calendar.

1.5 Attendance Requirements

You will have both lecture and lab classes. In each lab classes, you will typically be doing practical work to reinforce and learn the hands-on aspects of your modules.

There is a strong correlation between attendance at classes and marks achieved. It is very difficult to catch up on material and continuous assessments if you regularly miss classes. Continuous assessments are assessed through the semester and cannot be repeated until the next run of the module – so it If you are unable to attend classes you should inform your mentor and the relevant lecturer.

Attendance will typically be recorded in laboratories, and may account for a portion of your continuous assessment mark, depending upon the marking scheme used by the individual lecturer. Individual lecturers may also track attendance at their lectures.

1.6 Modules

This year you will be completing 11 modules. Each module has the following:

- **ECTS Credits**: These are credits which you gain for successfully completing the module. Over this year, you will complete a total of 60 ECTS credits. Each module has a multiple of 5 ECTS credits, as you will see on the next page.
  .

- **Semester**: This is the semester in which the module is offered, which will be Semester 1 (September to January), Semester 2 (January to May) or Year-long (September to May).

- **Module Title**: This is the name of your module.

- **Module Code**: This is a code which is sometimes used to identify the module.

- **Continuous Assessment (CA) Weighting**: Modules have two forms of assessment (or assessment components). Continuous assessment takes place on an ongoing basis throughout the year, through lab work, assignments and reports. The continuous assessment mark you achieve is combined with the examination mark to get the overall mark for the module. Marks are combined using the *weightings* outlined for each component. For example if a module mark consists of 60% exam and 40% CA, then the module mark will be calculated as (Exam Mark %*0.6) + CA Mark % *0.4).

- **Examination Weighting**: This is the weighting for the examination, which takes place at the end of the semester.

- **Pass requirements**: This is the rule sets out the requirement to pass the module.

- **Lecturer**: This is the lecturer for the module, along with their contact details.

Module Code	Module Title	ECTS	Weekly Contact Hours			Assessment		
			Lecture	Labora-tory	Tutorial	Exam Weight	Non-Exam Weight	Pass requirements
CMPU1001	Algorithm Design and Problem Solving	5	2	1	1	60%	40%	Combined CA and EXAM mark must be >= 40.
CMPU1005	Communications for Computer Technologists	5	3				100%	CA mark must be >=40%
CMPU1031	Web Development 1	5	2	2		50%	50%	Combined CA and EXAM mark must be >= 40.
CMPU1017	IT Fundamentals	5	3			70%	30%	Combined CA and EXAM mark must be >= 40.
CMPU1018	Mathematics 1	5	2		1	70%	30%	Combined CA and EXAM mark must be >= 40.
CMPU1019	Microprocessor Systems	5	2	2		50%	50%	Combined CA and EXAM mark must be >= 40.
CMPU1022	Operating Systems 1	5	2	1		60%	40%	Combined CA and EXAM mark must be >= 40.
CMPU1028	Programming with Persistent Data	5	2	1		50%	50%	Combined CA and EXAM mark must be >= 40.
CMPU1024	Program Design	5	2	1	1	60%	40%	Combined CA and EXAM mark must be >= 40.
CMPU1025	Programming	10	2	2	1	50%	50%	Combined CA and EXAM mark must be >= 40. AND **Exam mark must be >=40%**
CMPU1006	Computer Architecture and Technology	5	2	1	1	60%	40%	Combined CA and EXAM mark must be >= 40.

The module pass mark is 40%. Programming also has a written exam threshold mark of 40% in order to pass the module. Each module has a detailed module descriptor which is included in the Appendix of this document. The definitive source programme and module information is our programme document, which is available at the *DT228 Course Document and Information* module in webcourses at: http://www.dit.ie/lttc/webcourseslinks/.

1.7 Selection of Modules

In year 1, all modules are core, so you must take and pass all modules in order to progress to year 2.

1.8 Supervision Arrangements

This is not applicable to first year DT228 students.

1.9 Assessment

Students on this programme are assessed through:

- Continuous Assessment
- Examination

Examinations take place in two sessions: (1) December /January and (2) May in local locations, some of which may be off campus, and which will be notified to you, as detailed in the academic calendar - http://www.dit.ie/academicaffairsandregistrar/calendar/.

Continuous assessment takes place throughout the semester. If is very important that you keep up to date with your continuous assessments for your modules – as the standard repeat mechanism is to wait for the next running of the module the following academic year.

You will get an assessment schedule from your year mentor at the start of each semester. The assessment schedule will be available online in the DT228 Course Document and Information module on Webcourses.

The weighting for the continuous assessment and examination components for your modules are included in section 1.6 above.

1.10 Submission Guidelines

The individual lecturers will instruct you on how and where to submit your assignment. Most will use Webcourses, but others may use either the School Office or email. You are required to submit assignments by the deadline set by your lecturer. The penalty for late submission is set by the individual lecturer. If you are unable to submit an assignment on time for a valid reason (valid reasons are set out in the Personal Circumstances form) you may able to submit a Personal Circumstances form. The Personal Circumstances form is available at:
http://www.dit.ie/qualityassuranceandacademicprogrammerecords/student-assessment-regulations/general/

All assessments are intended to determine the skills, abilities, understanding and knowledge of each of the individual students undertaking the assessment. Cheating is defined as obtaining an unfair academic advantage and any student found using any form of cheating, attempting to cheat or assisting someone else to cheat may be subject to disciplinary action in accordance with the Institute's Disciplinary

Procedure. You should consult the Information Leaflet In Relation To Cheating which is included as an appendix to this document.

1.11 Getting my results

You will complete examinations in January and May, at the end of semester 1 and 2 respectively. Following these examinations, an examination board is held to confirm your marks, after which marks are released online. You will receive an e-mail providing you with details of how to access these results when they are released. Following the completion of semester 2, you will also receive written notification of your results.

If you are completing examinations in the supplemental session in August, you will receive notification of your results online in September. You will receive an e-mail providing you with details of how to access these results when they are released.

1.12 Obtaining Feedback

Individual lecturers will provide you with feedback on your assignment(s).

Upon completion of the examination process, you can view your examination script. To do so you must submit a request to the School Office.

The rules for rechecks, remarks and appeals are outlined in the General Assessment Regulations as described in section 1.13.

1.13 Assessment Regulations

The General Assessment Regulations govern all assessment in the Institute, including all rules regarding:

- Absence
- Appeal
- Assessment
- Carrying modules
- Compensation
- Conduct during examinations
- Coursework
- Deferral
- Examinations
- Notification of Results
- Personal Circumstances
- Progression
- Reassessment
- Recheck and Remark

The General Assessment Regulations are available online at http://www.dit.ie/qualityassuranceandacademicprogrammerecords/student-assessment-regulations/general/

The programme document contains a detailed description of this programme. It is available through the library and online in the DT228 Course Document and Information module on Webcourses.

1.14 Progression

In order to progress from this year to year 2 of the programme, you are required to pass all modules. Compensation rules may apply, as set out in the General Assessment Regulations (see section 1.13).

If you do not pass a module, you will be informed of the requirements to retake that module or a component of the module. Examinations can typically be retaken in the supplemental examination session in late August. The standard repeat mechanism for the Continuous Assessment component of a module is to retake it when the module is next running - which is typically the following academic year.

1.15 Applying for Exemptions

If you think that you have already completed sufficient previous study to quality for an exemption on a module, you may apply for an exemption. You will be required to complete an exemption form, and provide documentation (e.g. certificates) to support your case that previous work equates to the content of the module you wish to be exempt from. You should contact your year mentor in order to submit an exemptions application.

1.16 Health and Safety

All students must familiarise themselves with the Health and Safety rules of the Institute, which are available online at http://www.dit.ie/healthsafety/.

1.17 IT

You are provided with your login for the Institute's IT Systems when you register. You use the same login for:

- Webcourses: http://www.dit.ie/webcourses
- Email: http://mydit.ie/
- Computer laboratories
- Wifi

You will find a detailed explanation of how to use the DIT IT systems, including those listed above, printing services, password facilities and others at http://www.dit.ie/is/student/.

Prior to using DIT computer services, all students should familiarize themselves with DIT's Student Regulations Governing the use of Computer Resources - http://www.dit.ie/is/governance/regulations/studentregulations/. A copy of these regulations in provided as an Appendix to this document.

You can get support for your IT queries by contacting support@dit.ie or by ringing 01 402 3123.

School of Computing Lab Rules

- Labs are a key school resource and should be treated with respect at all times.
- Laptops can be used in the labs but at present students must use the wireless network only.
- Students must not unplug PCs to power their own laptops.
- Eating and drinking in the labs is strictly forbidden

1.18 Your Own Laptop

As a student of Computer Science, it is important that you have the tools you need to help you to study. While our computer laboratories are fully equipped with computers that are sufficient for your current basic needs, it is advisable to get a proper laptop of your own. There is wifi available throughout the Kevin Street and Aungier Street campuses, so you should be able to use your laptop in the college if it is wifi enabled. It is also likely that you will use cloud storage, but your laptop should have a working USB port and a hard drive with at least 250GB capacity.

You should choose a machine that has a 64-bit processor with an Intel® Core I5 or higher spec chip, with 8GB of RAM. This processor has VT-x support that you will need for installation of virtual machines. You will also need Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality, all of which is available on the Intel® Core I5. If you are going for a different processor, please check that it has these features.

You will use different Operating Systems and different utilities for the different modules that you take. Although you will study the Linux Operating System, you can access that on a laptop that runs either under Windows or Mac OSX. The Operating System you choose can be either Mac OSX or Windows (7 minimum), but be aware that if you go for Mac OSX, you may need to use a virtual machine with Windows on it to make use of the same software that is in common use on the DIT ICT domain.

Please be aware that technology is moving on all the time, so this specification is intended to see you through the next few years:

· Processor: Dual or quad core 64-bit processor (Intel i5 2GHz or higher (or equivalent)) with a minimum of 8GB RAM with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality.

· Hard Drive should have at least 250GB

· Operating System can be either Windows (7 minimum) or Mac OSX.

1.19 Webcourses

Webcourses is DIT's online virtual learning environment. Most, but not all, lecturers use webcourses to support the teaching of their module. This system is used for delivery of lecture notes, online discussion, assignment submission and assessment feedback. You can access webcourses online at http://www.dit.ie/lttc/webcourseslogin/ using your IT login. If you cannot access a particular module, you should contact the module lecturer.

You will only be able to access webcourses if you have registered.

1.20 Rules and Regulations

You should review the rules and regulations of the Institute, which cover the following areas:

- Registration Regulations
- Regulations Governing the use of Computer Resources
- General Assessment Regulations
- Disciplinary procedures
- Identity and Access Management
- Social Welfare Fraud Controls
- Student Alcohol Policy
- Student Dignity and Respect Policy

These are available online at http://www.dit.ie/registration/hererules/

1.21 Student Charter

The mission of the Institute emphasises partnerships between staff and students and working together to improve the quality of service and the response to diversity of needs. The focus in our Institute community is thus on personal responsibility to each other. Our Student Charter is intended to underpin this joint personal accountability, and was drawn up by a group of staff and students after consultation with staff and students across the Institute. It sets out the level of service and standards of excellence we intend to provide for our students from the point of making an application to come to the Institute, to student life while studying here including the academic, social, cultural & athletic environments of the Dublin Institute of Technology and describes the Institute's expectations of students in such matters.

The student charter is available online at
http://www.dit.ie/media/campuslife/olddocuments/DITStudentCharter.pdf

2. My Programme

2.1 Programme Structure

The programme is a four stage programme which can be completed in four years of full-time study. Each stage is split into two semesters. Stages (years) 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 three paths:

1. Work Placement: Students undertake a full time placement in industry for six to seven months.
2. Take a Service Learning and Civic engagement project whereby students are immersed in an environment different to their typical environment. For example, previous students have undertaken projects in Malawi, working with the Wells for Zoe charity.
3. Specialist Modules: Students undertake a team project and four modules in specialist subject areas.

At stage 4 students select eight modules, four per semester, in addition to a Final Year Project.

Full details of the programme structure are available in the programme document.

2.2 Modules per semester

The following are the modules you will be studying in year 1, by semester. The full descriptors for the modules are shown in the appendices of this document. One of your modules, Programming, is a year long module, the remainder are a single semester long.

The list of modules and lecturers is shown in the table. Occasionally, the lecturer assigned to a module may change - particularly for modules scheduled ahead in Semester 2. As at Sept 2016, the scheduled names are:

Semester	Module Name	Lecturer	Lecturer's contact details
1	Communications for computer technologists	Leslie Shoemaker	leslie.Shoemaker@dit.ie
	Web Development 1	Cindy Liu	Cindy.liu@dit.ie
	IT Fundamentals	Art Sloan	art.sloan@dit.ie
	Program Design	Jonathan McCarty	jonathan.mccarthy@dit.ie
	Mathematics 1	Dr. Blathnaid Sheridan	Blathnaid.sheridan@dit.ie
1 and 2	Programming	Dr. Michael Collins	michael.collins@dit.ie
2	Microprocessor Systems	Frank Duignan	frank.duignan@dit.ie
	Operating Systems 1	Damian Gordon	Damian.gordon@dit.ie

	Programming with Persistent Data	Jane Ferris	jane.ferris@dit.ie
	Computer Architecture and Technology	Art Sloan	art.sloan@dit.ie
	Algorithm Design and Problem Solving	Andrea Curley	Andrea.curley@dit.ie

2.3 Study Abroad Opportunities

In third year of this programme, you have the opportunity to study for one semester with our international partners in Toulouse (France), Darmstadt (Germany), Brussels (Belgium), Varna (Bulgaria), Brno (Czech Republic), Barcelona (Spain) or Oulu (Finland). These will be explained in more detail during the second year of the programme.

2.4 Work Placement Opportunities

In third year of this programme, you have the opportunity to undertake placement for six months with an industry partner. The work placement module is very popular, providing students with real experience of working and learning in industry. Preparation for work placement begins with CV preparation and interview techniques training. Those who opt for a work placement and who have sufficiently good academic grades previously will then be put forward for interview by DIT with our work placement companies.

2.5 Outcome and Career Opportunities

Graduates of this programme are equipped with the skills to work in a variety of role in the science and technology industries. They are very successful on gaining employment in Ireland, but with some opting to go abroad to work with companies further afield.

In addition, some of our students opt to take further study at either MSc or PhD level.

Given the nature of Computer Science and its relevance to new products and services, we also have a cohort of graduates who choose to set up their own businesses.

2.6 Transfer and Progression

Students gain a BSc (Hons) in Computer Science on successful completion of year 4 of the programme. However, you have the option of exiting the programme at the end of year 2, with the award of Higher Certificate in Computer Science. You can exit after year 3 with BSc (Ordinary) Computer Science. To exit after Year 3, you must complete the Level 7 individual project module.

If you wish to apply for a transfer to another programme in the School such as the part-time BSc programme, you should contact the programme chairperson.

2.7 Professional Body Accreditation

This programme is accredited by the Chartered Body for Science and Technology. Students entering the programme will be eligible for membership of BCS, The Chartered Institute for IT upon graduation. For membership details please see http://www.bcs.org/.

Your final degree award will be classified as follows, with the final average calculated based on…

Average Mark	Classification
>=70%	First Class Honours (usually called a *first*)
60%- 69%	Second Class Honours, Upper Division (usually called a *two-one*)
50%-59%	Second Class Honours Lower Division (usually called a *two-two*)
40% - 49%	Pass

2.8 External Examiners

In the final year of your programme, you may be invited to meet with the external examiners at the time of the final examination board in June.

The external examiners for the programme play a vital role in assuring the quality of the programme. The external examiners for your programme are:

- Paul Collins
- Kevin Foley

3. My Input

3.1 Introduction

Students play a vital role in programme development and monitoring. As a student, you can provide feedback both formally and informally. You are also represented, through your class representative, on the programme committee, and by the Student Union on College Board and Academic Council.

3.2 Providing Feedback

Upon completion of each module, you are asked to submit a Student Feedback Form (Q6A) to your lecturer. Your lecturer will use this feedback in the ongoing development of their module.

Once per year you are asked to provide general feedback on your programme using the Programme Survey Questionnaire (Q6C). This feedback is provided to school management and is used for the ongoing development of the programme.

Copies of these forms are available online at
http://www.dit.ie/qualityassuranceandacademicprogrammerecords/forms/

Staff-student meetings are held throughout the year, during which you get an opportunity to provide feedback to the academic staff on the programme.

You can, at any point, ask to meet with your lecturer or your class mentor to provide informal feedback.

3.3 Student Representation

The Programme Committee is responsible for designing, monitoring and managing your programme. The Programme Committee meets at least once per semester. Your class representative is a member of this committee, and can bring issues of concern to the committee. At the end of year academic year the programme committee produces an Annual Monitoring report which provides a review of the year, incorporating feedback from students, staff and external examiners, leading to actions which will help enhance the programme in the following year.

College Board has responsibility for developing and monitoring the implementation of academic policy matters and in particular academic quality assurance procedures. All modifications to your programme need to be approved first by the Programme Committee and then by College Board. General academic issues of relevance to programmes in the College are discussed at College Board. You are represented at College Board by your Student Union.

Academic Council is a statutory body, provision for which is made in the DIT Act. It is appointed by the Governing Body of the Institute to assist it in the planning, co-ordinating, developing and overseeing the academic work of the Institute and in protecting, maintaining and developing the academic standards of the programmes and other academic activities of the Institute. You are represented on Academic Council by your Student Union.

3.4 Programme Review

Every five years the Programme Committee is required to review the programme, and present the reviewed programme to a panel comprised of academic staff from DIT, academic staff from elsewhere and industry representatives. This review is informed by the annual monitoring process and your feedback.

4. My Life in DIT

4.1 Planning my Studies

You can access DIT's online library system at http://www.dit.ie/library/. Certain facilities require the use of a login name and password. DIT library staff can advise you on how to access these systems.

Opening hours for the library vary according to the time of year. You can find opening hours online at http://www.dit.ie/library/openinghours/.

The Institute's Maths Learning Centre is provided to support you if you are having difficulty with Mathematics. You can find more information about this at http://www.maths.dit.ie/mlc/.

4.2 Clubs and Societies

There is a vast array of societies on offer in DIT. Currently there are in excess of 75 societies which range from volunteering, performing arts, course-related, campaigning & political, religious, sign language, student media, plus so much more. DIT Societies are run by the students for the students and supported by the staff of the society's office through advice, administration and finance.

For more details see http://www.dit.ie/societies/.

4.3 Supports

Check out DIT's Campus Life (http://www.dit.ie/campuslife/) for information about:

- Accommodation
- Careers
- Chaplaincy
- Counselling
- Disability Service
- Health Centre
- Mature Student Support
- Sports
- Student Financial Aid

Supports and advice for first year students are available online at http://www.dit.ie/fyi/.

Find out about your Student Union at http://www.ditsu.ie/.

4.4 Become a student ambassador

Every year, the College of Sciences and Health selects a number of Student Ambassadors to represent the College at events such as our Open Days. If you are interested in becoming an ambassador, send an e-mail to science@dit.ie.

5. Frequently Asked Questions

5.1 I can't find the information I need on the website

The DIT website has a search functionality which you can use to locate the information that you need. If you are still unable to find it, please talk to your Year Mentor.

5.2 What if I think I'm on the wrong course?

Talk to your year mentor.

5.3 What is a first, two-one and two-two?

This is the classification system we use in DIT for final degree awards. A first is a First Class Honours – the highest grade possible, achieved when your mark is at least 70%. A *two-one* is a Second Class Honours – Upper Division, for marks of at least 60% but below the level of first class honours. A two-two is a Second Class Honours Lower Division, for marks of at least 50% but below the level of Second Class Honours, Upper Division.

5.4 What if I have personal circumstances that mean I can't continue with my programme?

In a situation like this, you should always talk to your year mentor. You may also wish to talk with some of the other DIT services, including those listed at http://www.dit.ie/campuslife/.

5.5 What if I am sick and unable to do an exam?

You should contact the examinations office immediately, and submit to the examinations office a Personal Circumstances form with supporting documentation within the required time period as set out in the form - http://www.dit.ie/qualityassuranceandacademicprogrammerecords/student-assessment-regulations/general/.

What if I am sick during the semester?

You should contact your year mentor immediately, and submit to the year mentor a Personal Circumstances form with supporting documentation within the required time period as set out in the form - http://www.dit.ie/qualityassuranceandacademicprogrammerecords/student-assessment-regulations/general/

5.6 What are ECTS credits?

This is the credit system we use in DIT and elsewhere in Europe. It is the European Credit Transfer System, through which all modules and programmes have a number of credits which represents notionally the amount of learning hours you're usually expected to complete in order to complete the module. Each credit equates to 20 learning hours, and each year of your programme typically requires the completion of 60 ECTS credits. Typically, a one semester module represents 5 ECTS credits, and a year-long module (such as Programming) represents 10 ECTS credits – although there are exceptions within the programme to this.

5.7 I can't find my class

You need to check your timetable frequently, especially early in the semester as there may be changes made at short notice. Your timetable is on the web timetables system at http://www.dit.ie/timetables/.

5.8 What if I fail to submit an assessment?

If you fail to submit, you risk forfeiting the marks for that assessment – with late penalties potentially applied, depending up on the conditions set by your lecturer. Be aware that continuous assessments do not as standard have a repeat opportunity within the same academic year, unlike written exams which have a supplemental session in August/ September. It is very important that you keep up with your coursework.

5.9 What if I didn't meet a deadline for submitting an assessment?

Each lecturer sets the conditions for late submissions, typically applying a late penalty. For example, a lecturer may specify that submissions that are late, but within a week of the deadline lose 50% of their mark.

If you have extenuating circumstances for missing deadlines, such as medical circumstances, please make sure to inform your year mentor.

5.10 I've lost my student card

A replacement is available (for a fee) from Registrations, http://www.dit.ie/registration/studentidcard/replacements/. Remember you will not be admitted to your exams without this card.

5.11 What public transport goes to the various DIT locations?

See http://www.dit.ie/campuslife/transport/

6. Useful Links

- School of Computing website - http://www.dit.ie/computing/
- Dublin Institute of Technology – http://www.dit.ie
- College of Sciences and Health – http://www.dit.ie/science
- Academic Calendar - http://www.dit.ie/academicaffairsandregistrar/calendar/
- Accommodation Service - http://www.dit.ie/campuslife/studentsupport/accommodation/
- Campus Life - http://www.dit.ie/campuslife/
- Careers Service - http://www.dit.ie/careers/
- Chaplaincy - http://chaplaincy.dit.ie/
- Counselling - http://www.dit.ie/campuslife/counselling/
- Disability Support Service - http://www.dit.ie/campuslife/disability/
- EMail - http://mydit.ie/
- Feedback Forms - http://www.dit.ie/qualityassuranceandacademicprogrammerecords/forms/
- First Year Experience - http://www.dit.ie/fyi/
- General Assessment Regulations - http://www.dit.ie/qualityassuranceandacademicprogrammerecords/student-assessment-regulations/general/

- Health and Safety - http://www.dit.ie/healthsafety/
- Health Centre - http://www.dit.ie/campuslife/studenthealthservice/
- Information Systems - http://www.dit.ie/is/student/
- Library - http://www.dit.ie/library/
- Library Opening Hours - http://www.dit.ie/library/openinghours/
- Mature Student Support - http://www.dit.ie/study/mature/support/
- Quality Enhancement Handbook - http://www.dit.ie/qualityassuranceandacademicprogrammerecords/quality/handbook/
- Registrations - http://www.dit.ie/registration/
- Societies - http://www.dit.ie/societies/
- Sports - http://www.ditsports.ie/
- Student Financial Assistance Scheme - http://www.dit.ie/campuslife/studentsupport/studentfinancialsupport/
- Student Regulations for Information Systems - http://www.dit.ie/is/governance/regulations/studentregulations/
- Student Union - http://www.ditsu.ie/
- Study Skills - http://studyskills.dit.ie/Survival%20Guide/Survival%20Guide/Home/Homepage.html
- Timetables - http://www.dit.ie/registration/studentclasstimetables/
- Webcourses - http://www.dit.ie/webcourses

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

Appendix A: DIT Academic Calendar – Session 2016/2017

(The Academic Year comprises 2 Semesters of 15 weeks during which student learning and assessment takes place.) *. *Review week dates for DT228 will be communicated at start of term.*

A copy of the Academic Calendar can be viewed and downloaded from the DIT website at:

http://www.dit.ie/academicaffairsandregistrar/calendar/

Appendix B: Information Leaflet In Relation To Cheating

Introduction

All assessments are intended to determine the skills, abilities, understanding and knowledge of each of the individual students undertaking the assessment. Cheating is defined as obtaining an unfair academic advantage and any student found using any form of cheating, attempting to cheat or assisting someone else to cheat may be subject to disciplinary action in accordance with the Institute's Disciplinary Procedure.

The Dublin Institute of Technology takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessments. Plagiarism, and other forms of cheating, are breaches of academic values, academic conventions and codes of practice. It is widely accepted within academia that in the pursuit of knowledge, innovation and creativity academics and students alike will build upon the works of others. Fundamental to this process of human inquiry and discovery is the prerequisite that all sources of information utilised should be appropriately acknowledged. This elementary precondition enables the cultivation of scholarly activities and research to progress in an open and free environment.

If you are having difficulty with your work it is important to seek help from your mentor rather than be tempted to use unfair means to gain marks. Do not risk losing your degree and all the work you have done.

Definitions

The Institute’s regulations define a number of different forms of cheating, although any form of cheating is strictly forbidden. These are:

- submitting other people's work as your own - either with or without their knowledge. This includes copying in examinations; using notes or unauthorised materials in examinations;

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

- impersonation - taking an assessment on behalf of or pretending to be another student, or allowing another person to take an assessment on your behalf or pretend to be you;
- plagiarism - taking or using another person's thoughts, writings or inventions as your own. To avoid plagiarism you must make sure that quotations from whatever source must be clearly identified and attributed at the point where they occur in the text of your work by using one of the standard conventions for referencing.

The Institute provides clear guidance on how to reference your work correctly and your mentor can also help you. It is not enough just to list sources in a bibliography at the end of your essay or dissertation if you do not acknowledge the actual quotations in the text. Neither is it acceptable to change some of the words or the order of sentences if, by failing to acknowledge the source properly, you give the impression that it is your own work.

- collusion - except where written instructions specify that work for assessment may be produced jointly and submitted as the work of more than one student, you must not collude with others to produce a piece of work jointly, copy or share another student's work or lend your work to another student in the reasonable knowledge that some or all of it will be copied;
- duplication - submitting work for assessment that is the same as, or broadly similar to, work submitted earlier for academic credit, without acknowledgement of the previous submission;
- falsification - the invention of data, its alteration, its copying from any other source, or otherwise obtaining it by unfair means, or inventing quotations and/or references.

How to avoid plagiarism

Plagiarism can be either an intentional act whereby work is deliberately utilised and claimed as one's own, or it can occur unintentionally either through bad academic practice by the student or failure to inform yourself about the Institute's regulations. Plagiarism is not confined to written assignments, projects or theses; it incorporates all academic work, including practical workshops, demonstrations, three dimensional work and artistic practice.

The best way to avoid plagiarism is to become informed. You should request information from your lecturer, examine programme writing style guides and conventions, access programme documents and consult the Institute's General Assessment Regulations. Be clear about the particular referencing system of the discipline concerned, while noting that with modularisation students may study modules in different disciplines. Therefore, be sure you are using the correct referencing procedure appropriate to the discipline you are studying. Above all, clearly

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

acknowledge all sources of information you have accessed during your work. The Institute's Library Services have several useful texts on plagiarism which are both accessible and informative, and they are a good guide to referencing. Library staff provides Information Literacy sessions that include guidance on referencing and plagiarism.

Students may be asked to sign a declaration on all written assignments/theses submitted to verify that the work is not plagiarised. If such a declaration is not signed, however, students will still be subject to the regulations governing plagiarism.

Procedure for suspected case of plagiarism

While the lecturer may utilise their own professional judgement firstly to resolve the matter, some suspected cases of plagiarism may have to be dealt with by following due process as outlined below. Any judgement made will be confined to academic assessment principles, the Institute's regulations and based on an academic decision.

Initial stage

If a lecturer suspects that plagiarism has occurred, they shall notify the Head of School/Department/Assistant Head of School, as appropriate, who will inform the student of the concern and arrange a meeting (within 10 working days). In attendance should be the relevant Head of Department/Assistant Head of School, the lecturer, the student and a student representative (optional student's choice). During this meeting the student will be clearly informed of the precise nature of the concern. The student will be asked to provide clarification relating to the concern, and may also provide additional details in relation to the matter. At the end of this meeting the Head of Department/Assistant Head of School and the lecturer will consider the case (taking into account the academic experience of the student) and make an academic decision, choosing one of the following options:

- The matter has been resolved
- To resolve the matter the student may need to resubmit the work in question
- The matter is not resolved.

Enquiry Stage

If the matter is not resolved (option 3) the Head of Department/Assistant Head of School should promptly report the case to the Head of School who in turn will request the College Administrator to move the case to the Panel of Enquiry stage as provided for in Section 11.6 of the General Assessment Regulations. The student has a right to be informed of the membership of the Panel of Enquiry.

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

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

Appendix C: Student regulations governing the use of computer resources

1. Introduction

A. Email/Internet services are Institute facilities intended for use for teaching, learning, research and administration in support of the Institutes objectives.
Email addresses and Internet access, (where available to Staff), are provided for this purpose.

B. Computing resources are provided to support the academic, research, institutional, and administrative objectives of the Institute. These resources are intended for the sole use of the Institute's staff and other authorised users ("users") to accomplish tasks related to the user's status and duties as a member of the Institute consistent with the Institute's objectives. These resources, including software and data provided by the Institute, must not be used for commercial use or significant personal use.

C. Computers are powerful communication tools and must be used wisely. Use of these resources in a manner which contravenes these Regulations may result in disciplinary action which may include suspension or expulsion from the Institute.

D. The basic principle is that all users are expected to use common sense and to conduct themselves in a professional and appropriate manner in their emails and use of the Institute's computer facilities and the Internet. Staff are reminded that network postings or messages may be archived for years in various Internet search databases. In addition, these messages may be produced to others or to a Court in connection with litigation or disclosed under the Freedom of Information Act.

E. Users are individually accountable for all actions associated with their use of the Institute's information and technology systems.

F. Use of the Institute's computer facilities is a privilege granted to Institute Staff and the Institute reserves the right to withdraw or limit access to such facilities.

G. These Regulations apply to all staff of the Institute and to all users of Institute computer resources.

2. Confidentiality

A. The Institute does not provide users a guarantee or right to privacy or confidentiality in connection with the use of email and internet systems, and users should have no expectation of privacy in this regard.

B. The Institute reserves the right to retrieve information from its computers for the purpose of finding lost information or retrieving information lost due to system failure. The Institute reserves the right to monitor computer usage if there is a suspicion on

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

reasonable grounds of criminal activity or any breach of these Regulations, and in this event there should be no expectation of prior warning or notice.

C. Such monitoring includes the generation of logs which may be posted to detect and establish breaches of these Regulations. The Institute's computers and networks are Institute property and subject to standard maintenance and auditing activities as well as reasonable cause searches without notice to staff.

D. Users consent to such monitoring and accept that it is essential to properly safeguard the business of the Institute and to protect the rights of all staff.

E. Data is backed up as a regular feature of network administration. Deletion of email messages and other data does not necessarily prevent the retrieval of items. The Institute will not accept liability for lost or deleted data.

3. Use of Computer Resources

A. General

Users are required to abide by the law, by these Regulations, by the Data Protection Legislation, by the HEAnet Acceptable Usage Policy (attached), and by any additional regulations as may be laid down from time to time, in relation to the proper usage of computer equipment and materials.

It is the user's responsibility to be informed of the correct operating procedures for the computer resources or products used. A user who is uncertain as to the correct procedure in any situation should obtain clarification before proceeding.

Users must not engage in conduct which interferes with others' use of shared computing resources and/or the activities of other users, including studying, teaching, research and administration in or for the Institute.

B. Reporting/Queries

Users must immediately advise the relevant members of staff of any suspected acts of violation, breach in the security system or virus.

If users have any queries about the Institute's information and technology systems they should seek advice from the relevant member of staff.

C. User Access

Users must not utilise any other person's access rights.

Users must not attempt to gain access to resources or data for which they have not been specifically authorised nor should they attempt to bypass or probe any security mechanisms governing access to the computer systems.

A user must not misrepresent himself or herself as another individual in electronic communications.

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

Users must not divulge their Institute email address to any website that is not required by virtue of their designated duties, studies or research.

D. Content

The Institute's computer facilities and services should not be used to create, send, post, download, forward, view, store or display offensive, abusive, slanderous, vulgar, threatening or defamatory messages, text, graphics, or images or material from whatever source which may put the Institute at risk of prosecution, civil action, embarrassment or loss of reputation. This includes harassment, discrimination and intimidation of individuals on the basis of race, sex, religion, ethnicity, sexual orientation, disability, age marital status, family status or membership of the traveller community, etc. Specific examples include, but are not limited to, material that:

- is sexually explicit (whether visually or in written form) including descriptions or images of nudity or sexual acts;
- is discriminatory;
- advocates or supports violent or criminal acts;
- involves gambling;
- is by way of chain letter;
- violates copyright laws;
- breaches the Institute's harassment guidelines;
- involves use or transfer of unauthorised or unlicensed software;
- involves knowingly transferring viruses or virus based files;
- involves participation in responses to SCAMS, SPAMS or illegal activities.

Provided that the user first obtains the consent in writing from his or her Head of School or other designated officer, access to material described above for the purposes of bona fide study or research undertaken as part of an Institute programme of studies is not prohibited.

Email messages should be accurate, courteous and necessary.

Users should note that all messages on the Internet or access to Internet sites coming from the Institute's information and technology systems are identified as Institute activities. Electronic mail will be treated as a record of the Institute. It may be required to be retrieved as legal evidence. Therefore electronic mail should be used in the same way as other forms of written communication on Institute notepaper and all electronic mail messages should be regarded as permanent documents which will or may become public whether under the Freedom of Information Act or otherwise.

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

Users must not use Institute computer facilities to comment or communicate unofficially on any legal disputes or actions involving the Institute, its employees or students.

E. Computer Security

Users must not remove the Institute's information and technology equipment from Institute premises without prior authorisation of Head of Department or more senior officer. Users are responsible for the safekeeping of such information in computers whilst they are off-site and for reporting any associated loss of such information or computers to Head of Department or more senior officer.

Users must ensure that they do not deliberately or otherwise corrupt or destroy any software or data facilities or equipment accessible to them or introduce viruses to these resources. Users must not physically damage or deface any computer, ancillary equipment, documentation or related materials.

Messages should not be sent to a larger audience than is reasonably justifiable, in particular when they contain attachments.

Care should be taken when addressing email messages, to avoid mis-delivery.

Large and non-essential Internet downloads should be avoided, especially during peak hours, when they may be likely to impact on network performance.

Users are advised that software products are covered by licensing agreements. Such products and related materials shall not be copied.

Eating, drinking and smoking are strictly forbidden in any computer service area.

Users may not remove, disconnect, power off or otherwise interfere with any item of computer equipment without authorisation.

F. Compliance

Users must comply with the instructions and advice of Institute staff having responsibility for provision and support of computer services and for regulation of their use.

4. Discipline

Any user in breach of these regulations is liable for the legal and / or disciplinary consequences of that action which may take the form of withdrawal of facilities, suspension, expulsion or prosecution. These Regulations were adopted by the Dublin Institute of Technology Governing Body on 11th July 2001.

Any issues arising should be referred to local support staff in the first instance or escalated to the Information Services Support Desk (email: support@dit.ie, Telephone: 01 402 3123)

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

Appendix D: Module Descriptors

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.
- To explore the ideas of algorithm performance and efficiency.

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

- 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.

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.

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

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	

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
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

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

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

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

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

- 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	

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 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

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

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.

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

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

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	

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.

Module content
User centeredness and advocacy

Information assurance and security

IT systems model

Management of complexity

Information and communication technologies: HCI, Networking

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

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	

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.
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.

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

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 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.

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

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	

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.

Module content
Analogue systems vs. Digital systems

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

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

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:

- 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.

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

- 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	

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:

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

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

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 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.

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

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.

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	

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.

File Design: Efficient design of file structure and content.

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

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

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 (50%):

- Individual assignment
- Lab test

Written examination (50%)

- 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	

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
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.

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

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.

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	

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 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.

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

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.

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.

Reinforce good programming practices - both in theory and practical work - including version control, efficient coding practices, and maintainable code production"

Module Assessment

Assessment of the module is a combination of the following:

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

Continuous Assessment (50%)

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

Written examination (50%)

- One three hour, end of module examination.

  The pass mark for this module is 40%. Students must also achieve a minimum of 40% in the written exam in order to pass the overall module.

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	

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.

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

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

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.

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

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

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/