



Course Handbook
BSc (Hons) Software Engineering
2018-2019
Course Leader: Nick Mitchell
School of Physical Sciences & Computing



Please read this Handbook in conjunction with the University's Student Handbook.

All course materials, including lecture notes and other additional materials related to your course and provided to you, whether electronically or in hard copy, as part of your study, are the property of (or licensed to) UCLan and **MUST** not be distributed, sold, published, made available to others or copied other than for your personal study use unless you have gained written permission to do so from the Head of School. This applies to the materials in their entirety and to any part of the materials.

Contents

- 1 Welcome to the Course**
- 2 Structure of the Course**
- 3 Approaches to teaching and learning**
- 4 Student Support**
- 5 Assessment**
- 6 Classification of Awards**
- 7 Student Feedback**
- 8 Appendices**
 - 8.1 Programme Specification(s)**
 - 8.2 Succeeding at Assessment**
 - 8.3 Coping With Difficulties**

1. Welcome to the course

Software Engineering is a vocational course that studies the development of efficient and effective software to solve complex problems. The course is a mix of theory and practice that will allow you to develop skills that are demanded by the industry.

We hope you are looking forward to the challenges the coming year will bring, whether you are a direct entrant or proceeding with us from last year. If you are new, you may find joining a new course can be confusing at first, whether it is your first experience of Higher Education or you are joining the second or third year after having studied elsewhere. However, we hope you will soon fit in with the other students and become more familiar with the University's facilities, the course regulations and the staff. There is much for you to find out, so don't be afraid to ask any member of staff or your fellow students.

The University offers many opportunities: not just to learn about computing, but also to engage in a wide range of social or sporting activities, to make new friends and to develop your personal skills, and to work in industry. You will have a great deal of independence, but with this comes responsibility. You must balance your use of time to get the most out of University while making sure that you obtain a qualification that reflects your abilities.

Pay particular attention to the way we expect you to work. This can be summarised simply: come to all classes, hand your work in on time, don't copy from other students, make sure that you properly reference material you find in published literature, and if you have a problem, **ask**.

The next few weeks may be a time of great change, but we hope you will settle down quickly and enjoy your time with us in the School.

1.1 Rationale, aims and learning outcomes of the course



On successful completion of the course, you will be awarded BSc (Hons) in Software Engineering. The class of your award is calculated from the marks you achieve on the modules you take. (See Classification of Awards, below).

If you leave early or fail to satisfy the criteria for an Honours award, you may be entitled to a lesser award, e.g. Certificate of Higher Education for completing the equivalent of a full-time first year or Diploma of Higher Education for completing the equivalent of two years full-time.

Why study Software Engineering?

Software Engineers design and build many types of programs that you will be familiar with: word processors, image editors, distributed Internet applications, banking facilities, games, data security, anti-virus applications and data encryption. Less obviously, most electronic devices are microprocessor-based and require software to control them. Systems such as car engine management, racing car telemetry, digital watches, video cameras, iPods, mobile phones, etc., all need reliable, engineered software to operate effectively.

Software engineering is more than just programming. Consider what is involved in building a house. If a builder simply started building the house without a design and detailed plans, it would be a disaster. An architect is an essential player in such a project because every stage of development is considered and sequenced. With detailed plans and build sequence, building the house the customer wants should be relatively straightforward. Many software systems are delivered late, over budget and in an unreliable or unmaintainable state because they have not been properly engineered. A Software Engineer is the architect and the builder. You will learn how to manage, design, test and evaluate many types of software systems. This design knowledge (allowing you to be an Architect) is complemented by programming-related modules to allow you to implement complex software. On

completion of these two themes, you will have the skills to develop effective, efficient systems.

Any system, no matter how cleverly it has been created, must be usable. Using the same house building analogy, if the sink was in one room and the cooker in another, it would be very difficult to use the kitchen. Similarly software systems must be usable, so a Human Computer Interaction (HCI) theme runs through the course. Initially, you learn some useful introductory HCI skills. Later, the key HCI elements including user evaluation are taught in the Software Development module and the final year allows you to apply and evaluate your HCI skills in a development project.

What is special about BSc (Hons) Software Engineering?

When you qualify as a Software Engineer you will have a very wide range of skills which you can apply to almost any software problem so you can seek employment in just about any software-related field. This course helps you acquire a wide range of software development skills from specification and design to complex implementation skills in high level languages such as C++ and Java.

The course prepares you for employment in any software development environment so you are not limited to looking for work with a specific area. For example, Software Engineering graduates have been employed as developers by mobile phone manufacturers, the government's communications centre, GCHQ, British Telecom, and the games industry. One group set up their own games development company. Others have become web developers, data communication managers, lecturers, and a whole host of other software development-related careers.

At the University of Central Lancashire, we are concerned with the development of technical abilities and skills in all our Computing students, but we also want to develop your abilities for logical thought, independent learning, teamwork, ethical practice and communication. These capabilities will be important whatever your career.

To help you adapt to the inevitable changes and developments in technology during your career, the course considers underlying theory and potential new developments as well as current practice and techniques.

Will I find a career in Software Engineering when I graduate?

Jobs are obviously competitive but feedback from Software Engineering graduates suggests that generally they are well prepared for interviews and their first careers. Several students who have undertaken a work placement after their second year have been offered a job when they finish their degree.

The course aims to provide a solid foundation in software development, compatible with large and small organisations. You will not just become a coder; you will be able to design, build, and test high quality robust software solutions across a wide range of IT areas such as embedded real-time systems, interactive PC-based systems, computer games, etc. It will provide an all-round coverage of computing discipline, which will enable you to apply your knowledge quickly to other areas. For example, you will be able to transfer your programming skills to new languages.

Many projects require several engineers to work together so a team spirit is encouraged on this course. Students are usually better at one subject than another; the team ethos encourages the stronger students to help the weaker ones to learn. The advantage to the weaker student is obvious but the stronger student also benefits; explaining concepts to another person reinforces your understanding and raises issues you had not previously considered.

You will gain invaluable transferable skills through a practical approach to computing which is underpinned by academic theory: you don't just know the theory when you finish, you can actually apply it.

What are the aims of the course?

The course aims to develop the skills to enable graduates to develop software for many applications, but also encourages students to become independent learners capable of further research and study when required. Its aims are

- To develop the skills and understanding of theory necessary for the graduates to be employed in a Computing environment
- To develop the confidence to discuss and apply professional, ethical and legal concepts relevant to a software engineer.
- To encourage and enable students to become independent learners.
- To develop critical evaluation, communication, enterprise and self-management skills.
- To produce graduates with the skills and confidence to solve problems independently and as part of a team
- To provide an opportunity for students to develop transferable skills and enhance subject-specific expertise by undertaking a work placement
- Specific Aims
- To foster a systematic approach to the development of a wide range of software solutions using appropriate software development methodologies
- To encourage an analytic approach to the evaluation of tools, techniques, and methodologies for software development
- To ensure the users of software are considered from design through to evaluation
- To develop skills in the implementation of complex systems ranging from interactive user-centred applications to distributed enterprise applications.

What will I be able to do by the end of the course?

The Programme Specification, which is included in Appendix 8.1, lists the Aims of the course and the Learning Outcomes that you will have achieved when you graduate. However, this can all be summed up with:

You will be able to specify, design, implement and evaluate software solutions in a wide range of areas using appropriate software development methodologies and programming languages. You will be able to apply your transferable skills in most Information Technology-based environments.

1.2 Course Team

Key members of your course team are Nick Mitchell (npmitchell@uclan.ac.uk), your Course Leader, Lesley May (lmay@uclan.ac.uk), the First Year Tutor, and Nicky Danino (ndanino@uclan.ac.uk), the Project Tutor.

Each module will be led by a module tutor, who plans the module and sets the assessment. See Section 2 for a list of modules and the current module tutors. Module tutors may change from one year to the next. When you do your project, you will be allocated a project supervisor, who will help you manage your project.

Chris Casey (ccasey@uclan.ac.uk) is the Computing Academic Lead, responsible for the overall quality of all Computing courses and should be contacted if there are problems that can't be resolved by the module tutor or course leader.

1.3 Expertise of staff

You will be taught by staff with a wide variety of industrial and research backgrounds. They have substantial experience of teaching at this level. The School has researchers working in

a range of computing-related areas. Research into Human-computer interaction is important, particularly through the Child-Computer Interaction group (CHICI). There is research into data communications and networks, mobile computing, computer security and software engineering, particularly Agile software development. We have collaborated with Sony, BAE and a variety of UK and overseas Universities.

Staff industrial experience includes working in the games industry, the defence industry and the telecommunications industry, as software or database developers. The School is involved in consultancy and develops software for clients.

1.4 Academic Advisor

You will be assigned an Academic Advisor who will provide additional academic support during the year. They will be the first point of call for many of the questions that you might have during the year. Your Academic Advisor will be able to help you with personal development, including developing skills in self-awareness, reflection and action planning.

In the first year, your Academic Advisor will normally be someone who teaches you so you can speak informally during or after classes, but they will also be available in their office at times they will publish on their office door. You can contact them quickly by email.

In the second and later years, your Academic Advisor will normally be your Course Leader, who has specialist knowledge about your course and the opportunities it offers.

If you have a problem contacting your Academic Advisor or Course Leader, for example, because they are off sick, ask the Administrative Hub for advice.

1.5 Administration details



Campus Admin Services provides academic administration support for students and staff and are located in the following hub which opens from 8.45am until 5.15pm Monday to Thursday and until 4.00pm on Fridays. The hub can provide general assistance and advice regarding specific processes such as extenuating circumstances, extensions and appeals.

Computing and Technology Hub

C&T Building Room 235

Contact Details: candthub@uclan.ac.uk or +44 (0)1772 891994

1.6 Communication



Outside of taught classes, we will normally communicate with you by email and by E-Learn. The University expects you to use your UCLan email address and check regularly for messages from staff. If you send us email messages from other addresses they risk being filtered out as potential spam and discarded unread.

If you want, you can set up rules to redirect emails to your personal email address. If you send us email messages from other addresses, they risk being filtered out as potential spam and discarded unread.

Staff will try to reply to emails within 24 hours during the working week. However, they may take longer during busy periods or if they are away from the University. Time-critical issues should be raised with your course leader or academic lead if you don't get a response in a reasonable time.

1.7 External Examiner

The University has appointed an External Examiner for your course to help to ensure that the standards of your course are comparable to those provided at other higher education institutions in the UK. If you wish to contact your External Examiner, do this through your

Course Leader and not directly. External Examiner reports are available through the Computing Student Noticeboard, which you can access through Elearn Blackboard.

Every module has an External Examiner, who reviews examination papers and coursework briefs set by a module tutor and internally verified by another tutor.

The external examiners moderate a sample of student work after it has been marked and internally moderated by the course tutors. The sample includes work awarded marks from the different classes, that is, Fail (<40), Third (40-49%), Lower Second (50-59%), Upper Second (60-69%), First (70+%). The moderators check that the standards are appropriate and that the work has been marked consistently.

Dr Paul Parry of Sheffield Hallam University is the External Examiner who takes overall responsibility for checking the quality of the course, particularly for assessments and the way they are marked on the key modules. Other examiners have responsibility for other modules

2. Structure of the course

2.1 Overall structure



For a full-time 3 year Honours Degree, you will take 18 modules, six in each of 3 years for a full-time student. Part-time students will study no more than 4 modules per year. If you have previous study at an appropriate level, you may be entitled to exemptions. Each module has a level rating, ranging from 4-6, roughly corresponding to years 1-3 of your Degree. If you are enrolled on a foundation entry year, you will study an additional 6 modules to prepare you for the first year of the 3 year Honours Degree.

To obtain an Honours Degree, you must pass 18 modules at level 4 or above, with at least 12 at level 5 or 6, including at least 5 at level 6. The double module project will provide two of the level 6 modules. Some students will start with a Foundation year. This consists of 6 modules that prepare you for the first year of the Honours programme. Two thirds of these are technical modules. The other modules help you to develop the skills you need to succeed in Higher Education and to provide a broader education, which will be very useful in your future career.

You will normally study in Preston, but you may have the opportunity to study the course at our Cyprus campus or to undertake equivalent modules with a University abroad. Discuss these possibilities with your course leader or the first year tutor.

It is a good idea to take a year out in industry – a placement – between your second and final year. This is optional, but will give you valuable work experience that will make you stand out when you are looking for a career. During your second year, we will provide help and advice on seeking a placement, but we cannot guarantee you a suitable placement. As you will be treated as a normal paid employee, you will have to apply for and undergo the normal company admissions process to obtain a placement. Placements can be anywhere in the UK or even abroad.

2.2 Modules available

Each module is a self-contained block of learning with defined aims, learning outcomes and assessment. A standard module is worth 20 credits. It equates to the learning activity expected from one sixth of a full-time undergraduate year. Some optional modules may not be available every year and timetabling clashes may prevent you taking certain combinations of modules.

Foundation Year

Most students will go directly into the first year. However, if you are taking the foundation year, you will study six modules at level 3. Three of these will develop general academic

skills, which will help you to become an effective University student. Three will develop technical and personal skills more directly related to computing.

[Note: the module tutor may change from one year to the next.]

Code	Module Title	Credits
COC001	Introduction to Software Development	20
COC002	Investigating IT	20
COC003	Problem-solving for Computing	20
COC004	Study Skills 1 – Learning How to Learn	20
COC005	Study Skills 2 – Developing Academic Skills	20
COC006	Introduction to Mathematical Methods	20

At the end of the foundation year and at the end of stage one, you can remain on your current course or choose another. We will discuss the options available in the second semester of the foundation year.

Level 4

The first year is very similar to all the computing degrees. This enables you to try a wide range of computing subjects without committing yourself to a specific course from day 1. Typically, you should use your experience in CO1401 and CO1404 to determine if Software Engineering is the right course for you. If you enjoy these modules then you should enjoy the course. If not, you might want to consider another course at the end of year one. We usually have students transfer both into and out of the course at the end of the 1st year - there is no penalty for doing so.

[Note: the module tutor may change from one year to the next.]

Code	Module Title	Credits
CO1111	Computing Challenge	10
CO1301	Games Concepts	20
CO1401	Programming	10
CO1404	Introduction to Programming	10
CO1507	Introduction to Networking	20
CO1508	Computer Systems and Security	20
CO1605	Systems Analysis and Database Design	20
CO1801	Practitioner Skills	10

Why are the first year modules important?

The first year is a 'common first year' apart from Games Concepts, which is replaced by Interactive Applications on some other courses.

Programming is obviously central to Software Engineering and the Games Concepts module allows you to apply your programming skills to the development of interactive graphics-based programs.

Systems Analysis and Database Design studies the problems of obtaining requirements, designing systems and implementing databases, which are an essential component of many complex software applications.

Such complex applications are often distributed, relying on the data communications technology covered in Introduction to Networking. Software engineers must be able to develop secure systems, which require an understanding of computer technology and human factors explored in Computer Systems and Security.

The Computing Challenge introduces teamwork and presentation skills, which help you to work on projects and to interact with managers and clients.

Level 5

From this point, you will specialise in Software Engineering. Level 5 modules increase in pace, technical coverage, technical depth and theoretical concepts. You will also be encouraged to become an independent learner: we can teach you many things but if we teach you to learn, you can teach yourself for the rest of your life. This year you will find that the more you learn, the more you realise how much more there is to know so it is essential that you develop your independent study skills to cope with this constantly changing subject.

There is a focus on software development this year, particularly in programming; you will use several programming languages as we concentrate on enhancing your programming ability in modules CO2401, CO2402 and CO2412. The Agile Professional module, CO2403, will teach you how to manage a project then lead you into an industrial scenario where you will develop a substantial application as a team using all the Software Engineering skills you have learned so far. For example, in previous years, the team's task has been to develop a fully featured Instant Messenger application, a note-taker application to help the deaf, a network monitoring tool, and a virtual reality game. Computer Security is essential for modern software engineers and also provides you with practical experience with the Linux operating system. Digital Systems considers the development of systems that require hardware and software expertise

Code	Title	Credits
CO2401	Software Development	20
CO2402	Advanced Programming with C++	20
CO2403	The Agile Professional	20
CO2412	Computational Thinking	20
CO2508	Computer Security	20
CO2509	Mobile Computing	20

The Placement Year

You can take this for a Sandwich award. It gives you industrial experience that will help you in your final year and make it easier to find a job when you graduate. Both overseas and UK-based placements have been available.

Code	Title	Credits	Status
CO2802	Industrial Placement Year	120	Optional

The industrial placement is a popular choice for Software Engineers as they can work effectively as a junior software developer even at this stage. Software Engineering placement students are often offered a job with their placement company on completion of their degree.

Level 6 - the Final Year

This could be your third year if you didn't take a placement opportunity or your fourth year if you did. This year will concentrate on advanced topics and preparation for either industry or a higher degree such as an MSc or PhD. This year will be more academically rigorous than in previous years but there is still a focus on practical skills; employers want graduates who can do the job, not just discuss how it could be done.

You will undertake an individual project in an area of interest to you. This is a major piece of work and is worth two of your six final year modules. It is an opportunity to design and develop a solution to any technically complex problem of your choice. The wide range of project topics for Software Engineers mirrors their wide range of skills.

Advanced Software Engineering looks at rigorous and formal methods of developing software. The Object Oriented Methods in Computing module examines advanced OOP concepts in various languages such as C++, Java and C#. The Distributed Enterprise Systems module continues the software development theme by looking at large company-wide systems and how they tackle common problems of security, concurrency and communications. To allow you to specialise, you will choose a single module from a list of options.

Code	Title	Credits
CO3401	Advanced Software Engineering Techniques	20
CO3402	Object Oriented Methods in Computing	20
CO3409	Distributed Enterprise Systems	20
CO3808	Honours Project	20
	One of	
CO2409	Computer Graphics	20
CO3514	Wireless and Mobile Networks	20
CO3519	Artificial Intelligence	20
CO3722	Data Science	20

More on the Project

The project is the opportunity for you to design and develop a solution to any technically complex problem of your choice. The wide range of project topics for Software Engineers mirrors their range of skills. A few examples of past Software Engineering projects are:

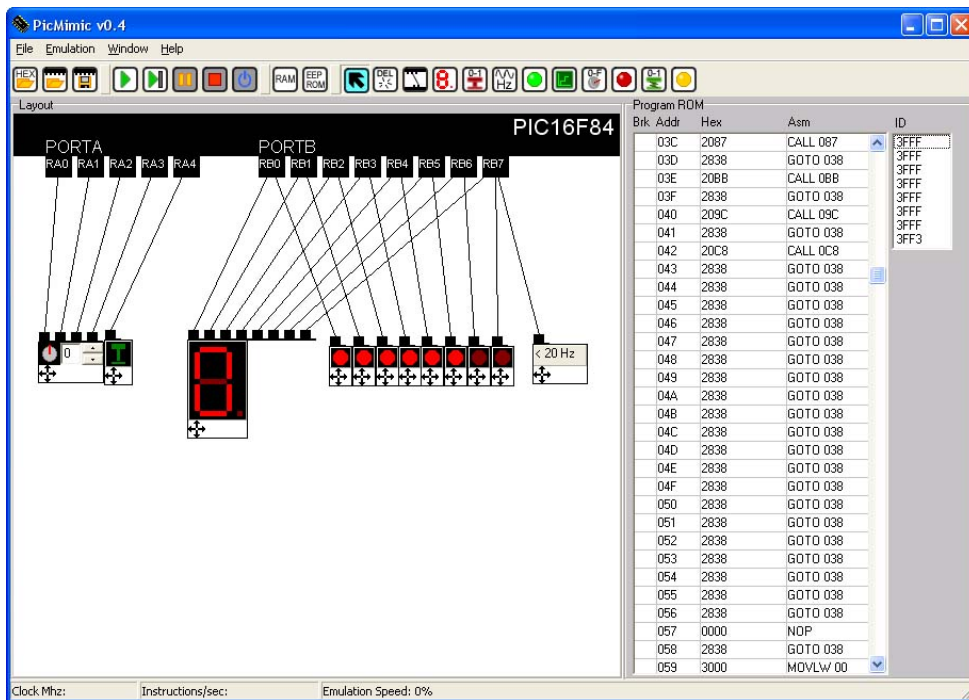
3-D pool game developed in DirectX. This also has an AI engine as the PC acts as a player. One of the screens is illustrated below:



Poker game. Unlike the pool game, which uses game development libraries, this game was written from scratch in C++. The main screen is illustrated below:

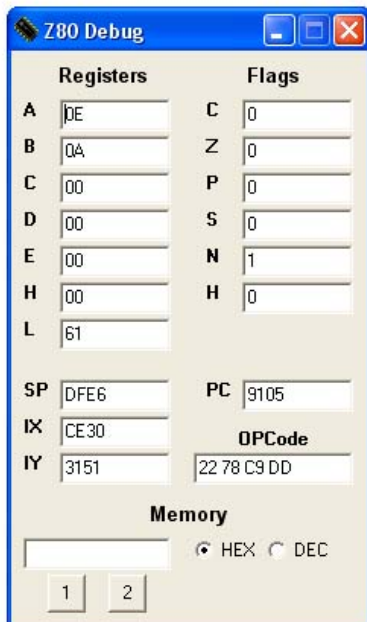


PIC16F84 microcontroller emulator. This emulates hardware used in the lab. It enables programs to be written for the PIC16F84 based target and tested without having access to the hardware. Different hardware configurations can be “wired-up” using this emulator.



Game Emulator

This application emulates old computer hardware to enable games to be played on the PC. This was written in C++. It emulated an old processor, the Z80 CPU, and games platform's video controller chip. An example of the playing area with CPU diagnostics is illustrated below:



2.3 Course Requirements



As a student hoping to become a computing professional, you should uphold the Code of Conduct of the BCS, the Institute for IT, which is the professional body for IT. We encourage you to join the BCS as a Student Member.

2.4 Progression Information

Discussions about your progression on the course normally take place in February each year. It is an opportunity for you to plan your study over the next academic year. The course team will tell you about the various modules / combinations available and you will both agree on the most appropriate (and legal) course of study for you.

After the first year of any of our computing Honours Degree courses, it is possible to transfer to another computing Honours course.

2.5 Study Time

2.5.1 Weekly timetable

Your timetable is available on-line at <https://weeklytimetable.uclan.ac.uk/>. You can also access it through the student portal. When planning the timetable, we will try to ensure that you don't have timetabled classes on one day per week. However, this isn't always possible.

2.5.2 Expected hours of study

We expect you to study for 10 hours per each credit taken (i.e. 200 hours for a 20 credit module) – this includes attendance at timetabled classes, assessment and time spent in private study. It means roughly 36 hours per week, most of which is in your own time. In lectures, the lecturer presents and explains concepts. In practicals, you will usually use worksheets to guide you through computer-based work. Tutorials are often based on worksheets and small or whole group discussion.

Most first year modules in Computing involve 3 hours of class contact, a one-hour lecture and two hours of tutorial or practical. You will normally have about 17 hours per week of class contact in year 1. You should work for at least that long outside of class, giving a working week of 36 hours on average.

Most second and third year modules have a lecture and either one or two hours of tutorials or practicals. You should work for around twice that long outside of class as part of your working week of 36 hours.

In your own time, you will have assignments and directed work from practicals or tutorials as well as reading and adding to your notes from the lectures. However, you are expected to find and read other relevant information for yourself. Computing is a very practical subject and there is always more practical work that you can do to develop your skills.

2.5.3 Attendance Requirements



You are required to attend all timetabled learning activities for each module. Notification of illness must be made to CandTHubAttendance@uclan.ac.uk.

Exceptional requests for leave of absence must be made to Lesley May (lmay@uclan.ac.uk) for first year students or to your Course Leader for other students.

We will monitor your attendance. It is your responsibility to make sure your attendance is recorded. You can check your attendance record through myUCLan. Occasional absences are not a problem, but you should discuss your attendance with the module tutor if your attendance is not recorded for more than one event that you attend.

You must only enter your own details on the attendance system. To enter information for anyone else is dishonest and would result in inaccurate records, which might mean that a student's problems might not be detected until it is too late for us to help. Any student who makes false entries can be disciplined under the student guide to regulations.

International students may have responsibilities under the UK Visas and Immigration (UKVI), Points Based System (PBS) - you MUST attend your course of study regularly; under PBS,

UCLan is obliged to tell UKVI if you withdraw from a course, defer or suspend your studies, or if you fail to attend the course regularly.

If you have not gained authorisation for absence, do not respond to communications from the University and are absent for four weeks or more, you may be deemed to have withdrawn from the course. If so, the date of withdrawal will be recorded as the last day of attendance.

3. Approaches to teaching and learning

3.1 Learning and teaching methods

The course offers you a mixture of lectures, tutorials and practical classes to help you learn. These are supported by material on Elearn Blackboard, our online virtual learning environment. You will need to supplement the classes with reading and practical work in your own time. Each of the methods can be very effective if you make proper use of them:

Lectures

These provide a lot of information to a large group in a short time. In most cases, notes will be provided on Elearn Blackboard. However, they are not a substitute for making your own notes. You don't need to copy down everything the lecturer says. The idea is to understand the main points and to write down what you need to remind you of them later. Don't be afraid to exchange notes with a friend or to experiment with only taking brief notes. Do try to relate the topics covered in a lecture to those covered earlier in this module or in others. Lecturers often provide notes, possibly through the Web. It is useful to have these before the lecture, so that you can add your own notes alongside. If you don't understand something, don't be afraid to ask. Your question may reflect problems that many of your colleagues are having. The lecturer will have limited time to answer questions in detail, but will be happy to clarify points that many students find difficult. Some questions may have to wait until after the lecture. You can always contact your module tutor by email, but there may also be a discussion board on Elearn Blackboard.

Tutorials and Seminars

These involve smaller groups to allow you to participate more actively. Do so. You can also use this opportunity to get help with your own specific problems.

Practical Classes and Laboratories

These give you the chance to practise practical skills under supervision. It is usually possible to get a copy of the practical sheet from Elearn Blackboard before the class. If you know what is required, you can make better use of the member of staff present. If you are stuck, do ask, but make a serious attempt to solve the problem yourself and show your lecturer your work to give them some idea of where you're up to. You can be fairly sure you're not the only one finding the exercise difficult. If the task were straightforward, we'd not be giving it lab time. You may have to spend time outside of the class to complete the exercise. Remember that the purpose of the exercise is not simply to follow the instructions like a recipe: you need to understand and learn from what you have done.

3.2 Study skills

WISER (<http://www.uclan.ac.uk/students/study/wiser/index.php>) provides support on how to take notes, to write essays and to do exams, which can make a big difference to your confidence and ultimately to your final Degree classification. You will be surprised at how few students don't bother to take advantage of the full range of support that is offered.

In your first year, we will help you to develop your study skills during induction, and in other first year modules such as CO1801, Practitioner Skills.

One key skill is time management. University life is very busy during term time. Some people find the difference between college or school and University very difficult indeed. Juggling

your time to attend lectures, seminars and labs, working on assignments and private study, and finding time for part-time work, plus all the other social activities that make university life so much fun; demands excellent time management skills.

Time management is probably one of the most difficult lessons you will have to learn. The workload does not become any less as your course progresses. You should develop skills to manage your workload for yourself. Here are a few tips that may be useful:

- Make a start on each assignment as soon as you receive it. You may have several weeks to complete it, but if you delay starting it, you will discover that deadlines creep up and you have too much to do, or you will concentrate on one piece of work to the exclusion of others. An assignment may look impossibly large, but a little work every day will soon have it done.
- Work in the library or labs when you have no timetabled classes: this way you get your assignments finished and make effective use of your time.
- If an assessment requires group working, co-operation is needed. If you work best late at night and the rest of your group are 'morning people' you will almost certainly have communication problems and possibly time management problems as well. Learning to compromise and being flexible is an essential part of successfully working in groups.

What do the course team expect of me?

We expect you to attend all classes and to do significant additional work outside of classes. Working consistently will help you understand the material and make assignments easier.

You will be expected to adopt a responsible attitude towards the quality of work that you produce and the deadlines you are set. **Keeping to deadlines and completing your assignments on time is an important part of the course. If you fail to keep to deadlines you will be penalised.**

Most day-to-day communication will take place through University e-mail. Read your e-mail regularly, at least once per day – not having seen a notice is no excuse for missing something important. You may wish to set up a rule to forward university e-mails to your home e-mail address automatically to ensure you have all this information.

If you have problems, please discuss them as early as possible with the relevant staff to try to resolve them.

You must inform the Hub

- if you change your address, so that we can contact you when necessary
- if you are absent for more than a couple of days through illness or other reason.

3.3 Learning resources

3.3.1 Learning Information Services (LIS)



During induction, you will learn about the resources provided by LIS (<https://www.uclan.ac.uk/students/study/library/>). Learning how to find and evaluate information is a skill that will benefit you throughout your career.

Books

Although these contain lots of information, it can be difficult to learn from a book unless you approach it properly. A textbook is not a novel - it doesn't have to be read from cover to cover. You need only read and understand the bits that are relevant to you. Therefore, before investing much time in a book, you should know what you want to get from it. This may mean skipping through the book and looking for key sentences and section headings. Use the Contents list and the Index.

There are a number of methods for 'reading with understanding'. You may not want to apply the methods rigorously, but they do contain some good ideas. A typical method is SQ3R:

- | | |
|----------|--|
| Survey | - scan quickly through the book to see what it contains |
| Question | - jot down the questions you hope the book will answer
(i.e. what you want to know) |
| Read | - read the parts of the book which answer your questions |
| Recall | - close the book and see if you can answer the questions |
| Review | - review the relevant sections of the book. |

Journals

These contain articles written by researchers or practitioners. They tend to be more up to date than books, but also more complex and difficult to understand. You will make more use of these during the second and third year, but that shouldn't stop you following up topics that interest you in journals. You can find journals in the library, but most students use the Internet to find published articles.

3.3.2 Electronic Resources

LIS provide access to a huge range of electronic resources – e-journals and databases, e-books, images and texts. See http://www.uclan.ac.uk/students/library/online_resources.php for more information. You should make use of the Discovery search engine (http://www.uclan.ac.uk/students/library/discovery_resource.php).

All modules will be supported by information on Elearn and you should make sure that you make use of this outside as well as in class.

Elearn Blackboard

This is our on-line learning environment, which contains

- Teaching material: outline lecture notes, tutorial and practical exercises and links to further reading
- Assessments: coursework, tests and on-line examinations
- Discussion groups: an opportunity for you to exchange views with other students and teaching staff

The Internet

This is a key source of information, which can give you access to books, journal articles and other material. It is important that you learn how to use Search Engines such as Google (www.google.com) to help you find **relevant** information. Remember that, unlike journal articles, which are reviewed by other experts, anyone can publish on the Internet – don't assume that everything you find is correct. Whichever source you use, you must ensure that you **DO NOT PLAGIARISE** someone else's work. In essence, this means making sure that say where you have got your ideas from: we use the Harvard Convention for References.

3.4 Personal development planning

Personal development planning is about assessing your own skills and abilities and planning how to develop them during (and after) your course. Technical development is part of this, but personal skills such as teamwork and communication skills are also important to your success at University and in your career. Employers put a great emphasis on these aspects.

Personal development is a particular emphasis of two modules in your first two years: CO1801, Practitioner Skills, and CO2403, Agile Professional.

3.5 Preparing for your career



CO1801, Practitioner Skills, and CO2403, Agile Professional are designed in collaboration with Careers to help you stand out from other graduates

The Careers advisory service ([Careers](#)) offers a range of support for you including:-

- career and employability advice and guidance appointments
- support to find work placements, internships, voluntary opportunities, part-time employment and live projects
- workshops, seminars, modules, certificates and events to develop your skills
- a daily drop in service from 09:00-17:00 for CV checks and initial careers information.

For more information, visit the careers team in Foster Building, or access our careers and employability resources via the Student Portal.

4. Student Support

If you have problems relating to a specific module, contact the relevant module tutor

For more general problems, as well as your academic advisor (see below) you can also discuss problems with Nick Mitchell, (npmitchell@uclan.ac.uk), your Course Leader, and in the first year, Lesley May (lmay@uclan.ac.uk), who is the First Year Tutor.

Chris Casey (ccasey@uclan.ac.uk) is the Computing Academic Lead, and should be contacted if there are problems that can't be resolved by the module tutor or course leader.

[The 'i'](#) Student Information Centre offers information and support on a wide range of issues.

4.1 Academic Advisors



Your Academic Advisor is an academic member of staff who will discuss your progress with you and help you to deal with problems. In the first year, your Academic Advisor will teach you so you will have the opportunity to speak to them informally. They will arrange to see you formally several times during the year. You can contact them by email to arrange a private meeting.

4.2 Students with disabilities

If you have a disability that may affect your studies, please either contact the Disability Advisory Service - disability@uclan.ac.uk - or let one of the course team know as soon as possible. With your agreement, information will be passed on to the Disability Advisory Service, who will work with the School to help you study. We will make reasonable adjustments to accommodate your needs and to provide appropriate support for you to complete your study successfully. This can include special arrangements for assessments, such as a personal examination plan. Where necessary, you will be asked for evidence to help identify appropriate adjustments.

Chris Casey (ccasey@uclan.ac.uk) is the acting disability co-ordinator for students with disabilities in CEPS. Please contact him directly for further advice / support, particularly if you have not been allocated a Disability Advisor. He is not a specialist disability advisor but can help to ensure that appropriate arrangements have been put into practice.

4.3 Students' Union One Stop Shop

The Opportunities Centre is the Union's One Stop Shop to find employment or volunteering whilst you study. With thousands of jobs and voluntary positions advertised, agency work through the Bridge and information on over 2000 volunteer positions within the Union.

5. Assessment

5.1 Assessment Strategy



Given the practical and vocational nature of computing courses, there is an emphasis on practical assessment. You will sit examinations, but you will also be assessed on the sort of tasks you might have to perform in industry including communication skills and team work. As a result, your progress will be monitored in a variety of ways.

All modules have some coursework assessment. This may take the form of a report or program to write, a system to analyse or design, or a presentation to give. We usually expect you to document the program, justify design decisions and evaluate the quality of the program. You should read the assessment criteria in the assignment specification carefully. No matter how wonderful your work seems to you, you won't do well unless you make sure you satisfy these criteria. We assess your work considering industry standards and professional norms. If you work to our criteria, you will learn how to become an effective, respected computing professional.

In-course assessment doesn't just assess what you can do – by doing the assessment you will learn and consolidate the skills you have. Your tutors will give you formal feedback on assignments to help you to do better on other assessments, but more importantly for your future career, to show how you can improve your performance on similar tasks in the future. By acting on the feedback from the lecturer, you will develop your competence and understanding.

You will also get a lot of informal feedback on your performance in class, particularly during practical classes.

Many modules have an examination at the end. Some of these examinations may be "open-book" examinations where you are allowed to take notes and/or books into the examination. Others are more traditional examinations, although some of these may be based around a case study that is issued before the examination.

The overall mark for each module is calculated as a weighted average of the coursework and examination marks. The details are given in the module descriptor held on Elearn Blackboard.

5.2 Notification of assignments and examination arrangements

How do I know what assignments I will have?

At the beginning of the year you will be issued with an indicative assignment schedule. Also at the start of each module, the module leader will tell you the latest date by which a piece of coursework will be released and the date by which you must submit it. This is to help you to plan your work. Examinations will be displayed on your on-line timetable.

How do I submit my assignments?

Assignments are usually submitted on-line through Elearn Blackboard, which gives you an electronic receipt. Keep a copy of it safe. *To reduce problems from lost assignments, keep a **complete** copy of the work you hand in.*

As far as possible your work will be marked anonymously, so assignment work submissions must not contain your name.

Aim to complete the coursework before the hand-in date to allow a margin of safety in case of technical problems. The University provides you with the software and hardware relevant to your course. If you choose to use your own equipment you are responsible for backing it up. Therefore please note that **failed/lost computers; failed/lost hard-drives, etc will not be accepted as an excuse for late submission.**

Meeting deadlines and dealing with problems in good time are essential parts of your preparation for industry. If you have a problem that may make it difficult to meet a deadline, discuss it with the relevant lecturer **before** the deadline if possible.

If you fail to submit a piece of work without a good reason, you will be given 0% for that work. This will make passing the module very difficult and may mean that you have extra work to complete over the summer. **It makes sense to hand work in before the deadline, even if it is incomplete.**

Will I be penalised for late work?

Except where an extension of the hand-in deadline date has been approved (using extenuating circumstances forms), lateness penalties will be applied in accordance with University policy as follows:

(Working) Days Late	Penalty
1 - 5	maximum mark that can be achieved is 40%
more than 5	0% awarded

5.3 Referencing

In your assignments, use Harvard convention for referencing whenever you make a reference to someone else's work. You can find lots of information about this on the internet, but you will be given more information about it during your course. If you are in any doubt, ask a lecturer for guidance.

5.4 Confidential material

If you use personal or commercially confidential information in your assignments (e.g. in your project), you have ethical and legal responsibilities to respect confidentiality and maintain the anonymity of individuals and organisations in your work assignments.

Students who do projects for clients must arrange for a client project agreement to be signed by the participants to ensure that they all understand their responsibilities.

5.5 Cheating, plagiarism, collusion or re-presentation

Please refer to the information included in section 6.6 of the University Student Handbook for full definitions. The University uses an online Assessment Tool called Turnitin. A pseudo-Turnitin assignment will be set up using the School space on Elearn Blackboard to allow students to check as many drafts as the system allows before their final submission to the 'official' Turnitin assignment. Students are required to self-submit their own assignment on Turnitin and will be given access to the Originality Reports arising from each submission. In operating Turnitin, Schools must take steps to ensure that the University's requirement for all summative assessment to be marked anonymously is not undermined and therefore Turnitin reports should either be anonymised or considered separately from marking. Turnitin may also be used to assist with plagiarism detection and collusion, where there is suspicion about individual piece(s) of work.

6. Classification of Awards

The University publishes the principles underpinning the way in which awards and results are decided in [Academic Regulations](#). Decisions about the overall classification of awards are made by Assessment Boards through the application of the academic and relevant course regulations.

For computing courses, the only special requirement beyond the University regulations is that you must take a double module project in your final year to be awarded an Honours Degree. This is to satisfy the requirements of the BCS, which is the professional body for IT in the UK. Your project mark guides the Assessment Board if your marks are borderline and they are considering your profile with a view to moving your final award up a class.

7. Student Feedback



You can play an important part in the process of improving the quality of this course through the feedback you give. For example, we made significant changes to the Foundation Entry Year after the first year of operation because of feedback from students indicated that study-skills modules would be better if they were more computing-oriented. A new maths module and two computing-based study skills modules were introduced.

7.1 Student Staff Liaison Committee meetings (SSLCs)

The purpose of a SSLC meeting is to provide the opportunity for course representatives to feedback to staff about the course, the overall student experience and to inform developments which will improve future courses. These meetings are normally scheduled once per semester. Details of the Protocol for the operation of SSLCs is included in section 8.2 of the University Student Handbook. SSLC meetings are scheduled and chaired by the School President and administered by CAS.

Do not simply save up problems to be raised at the meeting. To help resolve them quickly, problems should be raised with relevant staff, your course representative, or support staff as soon as you are aware of them.

8. Appendices

8.1 Programme Specification(s)

UNIVERSITY OF CENTRAL LANCASHIRE

Programme Specification

This Programme Specification provides a concise summary of the main features of the programme and the learning outcomes that a typical student might reasonably be expected to achieve and demonstrate if he/she takes full advantage of the learning opportunities that are provided.

Sources of information on the programme can be found in Section 17

1. Awarding Institution / Body	University of Central Lancashire
2. Teaching Institution and Location of Delivery	University of Central Lancashire Preston
3. University School/Centre	Physical Sciences & Computing
4. External Accreditation	BCS, The Chartered Institute for IT
5. Title of Final Award	BSc (Hons) Software Engineering
6. Modes of Attendance offered	Full-time Part-time Sandwich (Additional Year)
7a) UCAS Code	G600
7b) JACS Code	I300
8. Relevant Subject Benchmarking Group(s)	Computing
9. Other external influences	BCS, The Chartered Institute for IT
10. Date of production/revision of this form	April 2017
11. Aims of the Programme	
Software engineering focuses on the design and development of robust software solutions to complex problems. A software engineer needs to be able to communicate with customers and professionals of other disciplines, produce professionally presented and accurate technical documents, design software solutions to engineering problems, implement solutions from a variety of programming languages, test and evaluate the solution and have a good grasp of any legal and ethical implications of their work. Software engineers need a broad IT background to prepare them for careers	

developing software solutions in a variety of areas such as: computer networking, security, image and signal processing, embedded real-time system development, web, applications, graphics and games. This course not only develops the skills to enable the student to tackle any of these areas, but teaches them to become independent learners capable of further research and study when required. The teaching approach and content is sufficiently practical to enable the student to integrate directly into industry yet sufficiently underpinned academically to enable the student to go directly onto a higher degree if desired.

Common Computing Aims

- To develop the skills and understanding of theory necessary for the graduates to be employed in a Computing environment
- To develop the confidence to discuss and apply professional, ethical and legal concepts relevant to a software engineer.
- To encourage and enable students to become independent learners.
- To develop critical evaluation, communication, enterprise and self-management skills.
- To produce graduates with the skills and confidence to solve problems independently and as part of a team
- To provide an opportunity for students to develop transferable skills and enhance subject-specific expertise by undertaking a work placement

Specific Aims

- To foster a systematic approach to the development of a wide range of software solutions using appropriate software development methodologies
- To encourage an analytic approach to the evaluation of tools, techniques, and methodologies for software development
- To ensure the users of software are considered from design through to evaluation
- To develop skills in the implementation of complex systems ranging from interactive user-centred applications to distributed enterprise applications.

12. Learning Outcomes, Teaching, Learning and Assessment Methods

A. Knowledge and Understanding

The successful student will be able to

- A1. Explain, evaluate and apply techniques and methods to solve a range of computing problems
- A2. Evaluate and apply project management tools and techniques
- A3. Evaluate a range of languages for a given situation
- A4. Evaluate a range of methods for developing effective, efficient and secure systems
- A5. Explain, and evaluate a range of algorithms and data structures to meet a specification
- A6. Evaluate methods for testing and verifying software quality: functionality, usability and security

Teaching and Learning Methods

Acquisition of knowledge is mainly supported through lectures and directed learning. The role of directed learning increases as the course progresses. Understanding is reinforced through practical, tutorial and seminar work. Students develop understanding and retain knowledge best through practice, which may involve a series of small exercises, extended case studies or discussions. Students also learn during practical, development-based assessments, where they can objectively test their work, identify problems, recognise and correct their errors or misunderstandings. Drop-in help sessions are provided to support particular areas.

Assessment methods

Informal and formative feedback is provided in tutorial, seminar and practical classes through class discussion and individual advice. Formal assessment is through practical and written coursework, and time-constrained examinations, which may include on-line multiple-choice exams, traditional examinations, open-book examinations and partially-seen questions.

B. Subject-specific skills

The successful student will be able to

- B1. Solve technical and human problems relating to the development and use of IT-based systems
- B2. Specify and write robust programs using a variety of techniques and programming languages
- B3. Produce and evaluate software solutions to complex problems using the full project lifecycle
- B4. Design a user interface and test its usability
- B5. Develop secure, interactive and distributed systems

Teaching and Learning Methods

Software Engineering is a highly practical subject. Skills are developed in a co-ordinated and progressive manner during the programme. At level 4, the focus is on the acquisition of basic skills through laboratory exercises. At higher levels, more specialist equipment is used. Some practical work demonstrates advanced techniques, while extended practical work enables students to exercise creativity and develop their own solutions. Lectures, sometimes involving on-line demonstration, are supported by tutorials, seminars, practical exercises and directed work.

Assessment methods

A variety of methods are used to assess technical and personal practical skills. These include laboratory exercises, oral presentations, formal reports, and implementation exercises requiring the students to demonstrate a professional approach and develop appropriate supporting documentation and to evaluate methods and products.

C. Thinking Skills

The successful student will be able to

- C1. Investigate complex situations thoroughly and impartially
- C2. Locate, evaluate and integrate information from multiple sources
- C3. Evaluate ideas, methods and systems
- C4. Analyse and solve problems considering technical and relevant commercial issues

Teaching and Learning Methods

Intellectual skills are developed through practical work, tutorial and seminar work and coursework assignments. Discussion among students and with staff during tutorials and supervisory meetings are key methods for the development of thinking skills. Problem-solving is developed in practical classes,

seminars and tutorials. Throughout the course, students practise problem-solving individually and in groups. Students research, apply and evaluate information during the Agile Professional module and during the final year project.

Assessment methods

In class and in supervisory meetings, staff provide informal formative feedback. Intellectual skills are partly assessed through formal examinations but assessment of coursework and practical and theoretical project work is the main vehicle for assessment of the higher order skills. A variety of assessment methods are used, including formal reports, essays, and oral poster presentations.

D. Other skills relevant to employability and personal development

The successful student will be able to

D1. Communicate effectively with clients, users and developers, using informal and formal techniques

D2. Learn and work independently and as part of a team

D3. Operate within an ethical and legal framework appropriate to computing professionals.

D4. Plan, perform, manage and report on team and individual projects

D5. Identify and set personal goals relevant to long-term educational and career planning

Teaching and Learning Methods

The development of essential communication and transferable skills begins at induction and in The Computing Challenge module at the start of the first year. It is continued in the Systems Analysis and Database Design module. Teamwork and communication skills and legal and ethical understanding are further developed in The Agile Professional module at level 5 and in context in other modules through tutorial/seminar work and coursework assignments.

Relevant notations to support technical communication are introduced through tutorial and practical work using appropriate tools.

Teamwork skills are developed through practical experience during induction exercises and are reinforced throughout year 1, in a technical team exercise during induction at the start of level 5 and in Software Development at level 5. It culminates in the course-specific team project in semester 2 of the Agile Professional module, which requires the students to work in a team to solve a technical problem.

Whilst professional and ethical issues are addressed as appropriate in all modules, at each level there is a module designed to tackle professional and ethical issues. Concepts introduced in year 1 Computer Systems and Security are developed in year 2 Agile Professional and applied in the final year Project. These modules highlight issues students will meet in computer-related situations. Such issues are referenced by staff, when appropriate, within all aspects of the teaching. One of the main advantages of having specific modules to focus on these topics is that students become aware of and discuss computing-related issues that they have not formerly contemplated, and are then able to apply the newly found professional approach in the other modules on the course.

A major individual project, supported by supervisory meetings, reinforces and extends the student's abilities: they research topics relevant to their project, summarise and evaluate their findings in a literature review, plan and monitor their progress, solve problems and write an extended report.

Formative assessment during induction week starts the development of the student's ability to identify strengths and weaknesses and to set and work toward personal goals. This is continued during The Computing Challenge and subsequent level 4 modules, where students are encouraged to evaluate themselves and to consider career options. The level 5 Agile Professional module has talks by past placement students and companies to help students assess the benefit of undertaking an industrial placement. At both level 4 and level 5, feedback on assignments is discussed holistically by year tutors to help the students interpret the guidance and translate it into personal action.

Assessment methods

These skills are assessed through written coursework and presentations in many modules, but particularly in The Computing Challenge at the start of the first year, in The Agile Professional team project and the individual project, where students write an academic literature review and a project report, are interviewed, and give a poster presentation. In The Computing Challenge, the students participate in stand-up meetings mid-week to report on the team operation and progress and make a presentation of their achievements at the end of the week to demonstrate their products. During these, they discuss individual contributions and appropriate actions. In The Agile Professional

module, students hold regular meetings to monitor progress, informally assess individual performance and sign off work that has met their quality standards. Progress reports are assessed formally. Students are responsible as a team for an assessed literature review and individually for a critical evaluation of the project.

13. Programme Structures*				14. Awards and Credits*
Level	Module Code	Module Title	Credit rating	
Level 6	CO3401	Advanced Software Engineering Techniques	20	Bachelor Honours Degree Requires 360 credits excluding CO2802 but including a minimum of 220 at Level 5 or above and 100 at Level 6 Bachelor Degree Requires 320 credits excluding CO2802 but including a minimum of 200 at Level 5 or above and 60 at Level 6
	CO3402	Object Oriented Methods in Computing	20	
	CO3409	Distributed Enterprise Systems	20	
	CO3808	Honours Degree Project	40	
	CO2409	Computer Graphics	20	
	CO3514	Wireless and Mobile Networks	20	
	CO3519	Artificial Intelligence	20	
CO3722	Data Science	20		
Level 5	CO2802	Only for a Sandwich award Industrial Placement Year	120	Students who successfully complete CO2802, Industrial Placement Year, in addition to meeting the requirements of the award, will have the award "in sandwich mode"
Level 5	CO2401	Software Development	20	HE Diploma Requires 240 credits including a minimum of 100 at Level 5 or above
	CO2402	Advanced Programming	20	
	CO2403	The Agile Professional	20	
	CO2412	Computational Thinking	20	
	CO2508	Computer Security	20	
	CO2509	Mobile Computing	20	
Level 4	CO1111	The Computing Challenge	20	HE Certificate in Computing Requires 120 credits at Level 4 or above
	CO1404	Introduction to Programming	10	
	CO1401	Programming	10	
	CO1507	Introduction to Networking	20	
	CO1508	Computer Systems and Security	20	
	CO1605	Systems Analysis & Database Design	20	
	CO1301	Games Concepts	20	
CO1706	Interactive Applications	20		
		* Students enrolling on the first year of Software Engineering take CO1301 rather than CO1706		
Level 3*	COC001	Introduction to Software Development	20	Students who exit after successful completion of 120 credits at Level 3 will receive a transcript of the modules and grades
	COC002	Investigating IT	20	
	COC003	Problem-solving for Computing	20	
	COC004	Study Skills 1 – Learning How to Learn	20	
	COC005	Study Skills 2 – Developing Academic Skills	20	
	COC006	Introduction to Mathematical Methods	20	
* Only taken by Foundation Year Entry students				
15. Personal Development Planning				

Students are introduced to Personal Development Planning (PDP) during induction at the start of the first year. Following an introductory lecture, students conduct PDP activities with their Academic Advisors. Students' assessments of their own skills are used to guide team selection for the team challenge provided by The Computing Challenge module. Further work is done in during the following 4 weeks of this module through meetings with the first year tutorial team. Students are encouraged to audit their skills; set goals and produce a Progress Plan. In a progression meeting students consider matching their skills to their target Degree course.

At the start of the second year, students are told about the benefits of undertaking a placement and the work needed to find one. There are presentations by returning placement students. Other PDP activities involve meetings with their course leader. These sessions help students to identify their skills, use the feedback they have received on assessment performance, consider their long-term goals and identify the personal development necessary to succeed on the course and find relevant employment. Students also develop a CV and are involved in other employability activities during The Agile Professional module.

At the start of the third year, students are reminded of the support provided by the careers service and undertake activities to ensure they have started thinking about their future career or future studies.

Academic advisors are a key point of contact for students and ensure they take advantage of the available opportunities. They help students review the experiences and skills they gain while at university. They guide students to sources of help and advice where required. Problems identified by academic staff are followed up by academic advisors to help students to identify issues and decide appropriate actions.

16. Admissions criteria

(including agreed tariffs for entry with advanced standing)

**Correct as at date of approval. For latest information, please consult the University's website.*

112 UCAS tariff points at A2 or BTEC National Diploma (Distinction Merit Merit)
AND

5 GCSEs at grade C or above including Maths and English.

Students whose first language is not English must achieve an IELTS 6.0 (with no component score less than 5.)

Foundation Entry:

80 UCAS tariff points at A2 or BTEC National Diploma (Merit Merit Pass)
5 GCSEs at grade C or above including Maths and English.

Students whose first language is not English must achieve an IELTS 6.0 (with no component score less than 5.)

Qualifications equivalent to the above are accepted.

17. Key sources of information about the programme

- **University Web Site (<http://www.uclan.ac.uk/courses/index.php?q=software>)**

CO1706	Interactive Applications	O	✓		✓	✓	✓	✓		✓			✓	✓	✓					
--------	--------------------------	---	---	--	---	---	---	---	--	---	--	--	---	---	---	--	--	--	--	--

* Students enrolling on the first year of Software Engineering take CO1301.

Note: Mapping to other external frameworks, e.g. professional/statutory bodies, will be included within Student Course Handbooks

19. LEARNING OUTCOMES FOR EXIT AWARDS:

For **each exit award available**, list learning outcomes relating to the knowledge and understanding, subject specific skills, thinking, other skills relevant to employability and personal development that a typical student might be expected to gain as a result of successfully completing each level of a course of study.

Learning outcomes for the award of: BSc Software Engineering

- A1. Explain, evaluate and apply techniques and methods to develop high quality software
- A2. Evaluate and apply project management tools and techniques
- B1. Solve technical and human problems relating to the development of IT systems
- B2. Specify and write robust, secure and usable programs
- B3. Produce and evaluate software solutions to complex problems
- C1. Locate, evaluate and integrate information from multiple sources
- C2. Analyse and solve problems considering technical and relevant commercial issues
- D1. Communicate effectively with clients, users and developers, using informal and formal techniques
- D2. Learn and work independently and as part of a team

Learning outcomes for the award of: DipHE Software Engineering

- A1. Explain and apply techniques and methods to develop software-based systems
- A2. Apply project management tools and techniques
- B1. Solve technical and human problems relating to the development of IT systems
- B2. Specify and write programs
- C1. Locate and use information from multiple sources
- D1. Communicate with clients, users and developers, using informal and semi-formal techniques
- D2. Learn and work independently and as part of a team

Learning outcomes for the award of: CertHE Computing

- A1. Explain and apply techniques and methods to solve a range of computing problems
- A2. Describe key features of operating systems and networked IT systems.
- B1. Design and implement simple software with an appropriate user interface
- B2. Analyse an IT system and propose appropriate security considering legal and ethical issues.
- C1. Analyse and solve problems
- C2. Locate and use relevant information
- D1. Communicate with clients, users and developers, using simple techniques
- D2. Work independently and as part of a team

8.2 Succeeding at Assessment

There are guidelines for succeeding at assessments. Those who are unaware of these or who decide to ignore them will be at a disadvantage.

a) Do what you are asked to

When an assignment or examination question is set, the lecturer will have a good idea of what is necessary to answer it properly. You will normally be asked to do several tasks and be given guidance on the relative worth of each. Read carefully what is required and attempt every part but do not spend too much time on components of little worth.

b) Think of the person who will evaluate your work

Ensure that your work is well-organised and easy to mark. Don't use fancy folders that take five minutes to undo. Be concise and stick to the point. Try to demonstrate that even if you haven't time to complete the assignment tasks you do understand what is required.

c) Pace yourself

Equipment and staff are always harder to find as deadlines approach. You will save yourself time and effort if you start assignments early and don't let them pile up. If you leave work until the last minute, it will be hurried and will contain silly errors. Use an assessment timetable to plan and monitor your work - and complain to a lecturer who is late in giving out assignments!

d) Obey the rules

Read the assessment regulations carefully. Ensure that something is handed in on time. Even if you haven't managed to spend sufficient time on a piece of work or if you feel that you have misunderstood what is required, your attempt may show the lecturer where you need help.

e) Be sensible

Make sure you have done the preparatory work before you tackle relevant parts of the assignment. There are likely to be practicals on relevant material: do them. If you find them difficult or don't understand them, discuss them with the relevant tutor.

If an assignment seems very difficult or to require a very long time, discuss it with the lecturer to make sure you understand what is required.

f) Prepare properly for examinations

Everyone worries before exams, it's natural – a combination of a fear of the unknown and concern over the consequence of failure. However, you can increase your confidence by preparing properly.

1. **Use past papers** – they are available in each module's page on the University's Elearn Blackboard learning environment. Sometimes very similar questions will appear on your examination. At least they will give you a good idea of what the lecturer expects. The best use of past examination questions is to provide a focus for your revision. It may be useful to refer to them during the year as topics are covered.
2. **Keep up during the year** – revision does not mean "learning from scratch". Try to review your lecture notes at the end of each week and highlight any areas you don't understand. Find out about these immediately.

3. **Make a revision timetable** – allocate each subject roughly a fair share of time and try to stick to it. Don't make it too ambitious - you will need time to go out and relax. Joint revision with a few friends can be more pleasant, but beware of believing they understand the material any better than you do.
4. **Tackle the examination sensibly** – don't panic: if you are finding it difficult, so are other people. Make sure you have all the relevant equipment. Read the paper carefully, especially the "rubric" that specifies the number of questions you must answer and any restrictions on them. Plan to use the available time appropriately: allow equal time if the questions are allocated equal marks. Make sure you attempt and hand-in the full number of questions required, even if you think your answer to one question is poor. It's much easier to convert 0/20 to 5/20 than it is to convert 15/20 to 20/20.
5. **Choose your questions carefully** and ensure that you answer the question that is set, not the one you wish had been set.
6. **Present your answers thoughtfully** – make them easy to mark. Write clearly, but don't waste time on excessive neatness, for example, cross out rather than using liquid paper. Aim at quality not quantity. Don't try to save paper.
7. If you are running out of time, **use notes to show how you would have answered the question.**
8. **If your mind goes completely blank, move to another question,** or try to think of related topics or try to picture the relevant lecture notes.
9. **Don't waste time on post-mortems after the examination.**
10. **Be aware of hints given by lecturers** - the lecturer may indicate whether a particular subject is important (i.e. likely to appear on the examination) or what you are expected to know about a subject.
11. **Find out the consequences of failure** - normally if the worst comes to the worst and you fail an examination, you will be given a chance to resit another examination in that module at a later date

8.3 Coping With Difficulties

Will I be able to cope with the course?

We have a lot of experience of teaching computing to people from all sorts of backgrounds. Support is designed into the course for those who need it. For example, there are support sessions for Maths and Programming and you are encouraged to take advantage of these if necessary. Lecturers publish times when you can speak to them about your progress or discuss problems that you are having.

We have accepted you on this course because we believe you have the potential to succeed. Of course, to realise that potential, you must remain well motivated and work steadily throughout the year. Remember that you should do about 200 hours of work per module.

What if I have a disability

If you have a disability that may affect your studies, please let one of the course team know as soon as possible. We will make reasonable adjustments to accommodate your needs and to provide appropriate support for you to complete your study successfully. Where necessary, you will be asked for evidence to help identify appropriate adjustments.

What if my part-time work is interfering with my course?

Everyone is aware that many students have to work to support themselves, but you must decide whether the extra money is worth any risk to your studies. Work of 8-10 hours a week is unlikely to cause significant problems, providing that you are well organised. Much longer hours on a regular basis may affect your chance of success. If you have major financial problems, you should seek expert advice, possibly starting with the Students' Union or the "i". In the long term, it may be better to switch to a part-time route, to give you more time to work.

I am a part-time student, what if work interferes with my studies?

Make sure that you don't take too many modules. You can take up to four modules in a year, but four modules is a heavy workload on top of a full-time job. A smaller number of modules per year over a longer period may lead to a better degree. If you provide evidence that work is particularly intense, we may be able to arrange extensions to coursework.

What if I have problems?

If you have a problem with a particular piece of equipment or with installing software, ask a technician or LIS Customer support (01772 895355, or internal extension x5355, email LISCustomerSupport@uclan.ac.uk or in person in the Library) for help.

If you encounter problems with an assignment, seek help from the relevant lecturer. Discussing the problem with your friends may help, but make sure that any work handed in for an assignment is really your own. **Copying someone else's work is cheating and such cases are treated very seriously.** In the unlikely event that you cannot do the coursework even with advice from the lecturer, your notes and the library, hand in whatever you have done and then learn from the lecturer's comments.

There are many sources of help and support for general problems (e.g. your Academic Advisor or The "i" in Foster Building). Talk to someone: the relevant lecturer if it is a problem with a particular module, or your Course Leader or Academic Advisor if it is of a more general nature. Student Services have specially trained counsellors who can give advice on a wide range of personal problems. The sooner we are aware of the problem, the sooner we can help or advise you on the options available to you.

If there are circumstances beyond your control that affect your performance, tell the Module Leader as soon as possible. If a short extension or other action is appropriate, you should get documentary evidence and submit it in a special "Extenuating Circumstances"

envelope obtainable from the School Office. Where appropriate, we will take action during the year to alleviate genuine and significant extenuating circumstances. At the end of the year the Assessment Board will take these circumstances into account where appropriate. Deadlines for submitting extenuating circumstances are listed on the back of the envelope.

What if it all goes wrong?

To pass a year, full-time students must pass or be condoned in 6 modules. If, after resits, you don't manage to do this, you may be able to take the failed modules as a part-time student, or to repeat the year as a full-time student. Obviously, these options have financial implications, so you should discuss them with your family and other relevant people e.g. the Student Union Advisory Service or the University Academic Advisor.

Getting Help and Advice

Your lecturers will be able to help you with problems in their subject. Each lecturer is available for consultation. Their availability may be displayed outside the lecturer's office, or as part of their email signature). Do not be afraid to use this time.

Teaching staff will help you with software relating to their modules in practical classes. Staff in the library can provide additional information about the available software on the main computers. They can help you with problems you may have in using the computers. If you have technical problems with machines in the School, please email liscustomersupport@uclan.ac.uk with the following information

Room Number

Equipment Number e.g. 4738 or other identifier if no equipment number

Brief description of the fault

If the fault has affected your coursework, include a copy of the email in the coursework submission.

Other students will often help you with minor problems and can be a great help if used sensibly. They may be able to explain concepts or help with non-assessed practical work. However, don't assume that a student who sounds knowledgeable really does know better than you and do not copy assessed work. The work you hand in for assignments must be your own unless the assignment asks for group work. If other students are experiencing similar problems, you or your **Student Representative** should gather information and discuss it with the relevant lecturer. If problems persist, the first year tutor or, if the problems are serious, the Head of School may be able to sort them out.

Your academic advisor, course leader, or year tutor (year 1) are a good source of advice. The <i> and the Student Union Advice Centre have a lot of experience of helping students tackle a wide range of problems.