Programme Specification

This specification provides a 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 they take full advantage of the learning opportunities that are provided. As the content of the University's degree programmes is constantly being developed the information contained in this document is liable to change.

Programme Details

1. Programme title	Computer Science (Software Engineering) with an Industrial Placement Year		
2. Final award (e.g. BA, MEng or MSc)	Type: BEng	Duration: 4 years	
3. Intermediate/exit awards	Title (if different from main award): Computer Science (Software Engineering)		
	Type: BEng	Duration: 3 years	
	Title (if different from main award):		
	Type:	Duration:	
4. Framework for Higher Education Qualifications level	FHEQ Level 6		
5. Faculty	Engineering		
6. School / Department	Computer Science		
7. Other schools/depts (providing credit bearing modules for the programme)	None		
8. Accrediting Professional or Statutory Body	BCS, the Chartered Institute for IT		
9. Mode(s) of study	Full-time		
10. HECoS code(s) *Select between one and three codes from the HECoS vocabulary.*	100366	100374	100359
11. Relevant Subject Benchmark Statements	Computing		

Programme Code(s) (Internal use)	COMU43

12. Programme aims

The programme aims to:	
A1	provide a thorough academic grounding in the core subject matter of Software Engineering, with advanced study paths informed by the School's wide-ranging research interests.
A2	develop technical, professional and managerial skills through exposure to practical, industrially oriented projects, emphasising teamwork and communication as well as software design and development skills.
A3	provide a route to professional accreditation through the British Computer Society, leading to partial CITP and CEng accreditation.
A4	expose students to leading-edge world-class research in Software Engineering.
A5	provide students with direct experience of working in industry, applying and developing their technical and professional skills.
A6	produce immediately employable graduates with an industrially relevant mix of knowledge, practical skills and self-motivation.
A7	provide an international dimension with opportunities for study in universities in other countries.

13. Programme learning outcomes

Knowledge and understanding (K) On successful completion of the programme, students will be able to demonstrate knowledge and understanding of:	
K1	programming languages and styles, algorithms and data structures.
K2	discrete and continuous mathematical foundations for computing.
K3	software engineering analysis and design methods and process management.
K4	artificial intelligence and biologically inspired models of machine reasoning.
K5	computer hardware design and computer network architectures.
K6	the wider context of professional practice, including the relationship between computer science and society, the environment and the law.
K7	the commercial and industrial dimension to computing, through interaction with clients.
K8	software quality and comparative methodologies (core topic for the degree).
K9	a range of research-led topics taught in the final year of the degree.

K10	how subject-specific knowledge gained during the degree is applied in the workplace.
Skills and other attributes (S) *When considering the skills and attributes developed in this programme, please refer to the Sheffield Graduate attributes (SGAs). SGAs can be found here* On successful completion of the programme, students will be able to:	
S1	function in an Information and Communication Technology (ICT) environment using appropriate technology such as email, the Internet, shared data and code repositories.
S2	conceive, design and write correct working computer programs in several different programming styles, using a variety of compilers and development environments.
S3	construct and manipulate formal and mathematical models.
S4	apply a software engineering process and take a project through the stages of the software lifecycle, using design notations and software engineering tools selectively.
S5	communicate effectively in writing, present a two-sided argument, expose technical information clearly, comprehend and summarise research-level material with proper citation of sources.
S6	communicate effectively in speaking, interview and interact productively with a client, present and defend a substantial piece of work, engage with others and respond effectively to questions.
S7	work effectively in a team, demonstrating personal responsibility and group management ability, interpersonal skills, leadership and delegation, and plan to meet deadlines.
S8	design solutions for complex problems to meet a customer's needs within the context of a wider business practice.
S9	research material from multiple published sources, comprehend and filter such material and from it synthesise theories, principles or designs pertinent to a practical, problem-solving project.
S10	demonstrate personal initiative, self-motivation and problem-solving skills, through the selection and taking through to completion of a practical, problem-solving individual project with a research dimension.
S11	work effectively in an industrial environment.

14. Learning and teaching methods *(this should include a summary of methods used throughout the programme, including any unique features)*

The School fosters an environment with many opportunities for individual and group learning, but the responsibility for learning rests with the student, who must be personally organised and self-motivated to make the most of the programme. Teaching is offered through formal lectures, seminars, computer laboratories, problem-solving classes and project supervision.

Lectures are formal presentations to a large class of students by a lecturer. The purpose of a lecture is to motivate interest in a subject, to convey the core concepts and information content succinctly and to point students towards further sources of information. Lectures are interactive and students are encouraged to ask questions at suitable points. Students are expected to take notes during lectures, adding detail to published course materials (which are generally provided in advance on electronic media).

Seminars are longer semi-formal presentations to a class of students by a lecturer, researcher, industrial partner or student, describing an area of their current research or business. There is typically more opportunity than in a lecture to structure the session internally with questions, problem solving and other kinds of interactive or shared learning experience, in which the students may also participate in the teaching and lead discussions.

Computer laboratories are sessions supervised by teaching assistants (under the direction of the responsible lecturer) in which students work at a computer, to develop a specific practical skill, such as familiarisation, computer programming, or the use of a software engineering or mathematical modelling tool.

Problem-solving classes are sessions conducted by a staff member with a class of students, in which exercises are completed interactively and solutions are provided within the period. The purpose of such a class is to help students engage with, and assimilate the material presented in lectures and start to apply this knowledge.

Project supervision involves regular meetings with a student's individual or group project supervisor, who may also be their personal tutor. During each session, students report on their progress to the supervisor, who highlights further areas of investigation, helps with technical problems, advises about the content and structure of technical reports and generally encourages the students to organise their time effectively.

The transition to self-motivated learning is encouraged through specialist teaching materials such as lecture handouts or copies of lecture slides, supplied via the VLE. Set course texts and background materials are available through the University libraries, at bookshops and also via the Internet. Active learning is fostered and promoted through engagement in practical work, such as exercises, assignments and projects.

Exercises are short tasks, either writing computer programs or working out solutions to other kinds of set problems, which are typically reviewed at the end of the session.

Assignments are typically offered in stages over a number of weeks, involving the design and implementation of a software system to perform a given task, or the researching of a body of information leading to the writing of a discursive essay or report on a given topic.

Projects are undertaken individually or in groups over one or two semesters. Projects typically solve a larger problem, possibly for an industrial client, possibly with a research dimension. Individual projects require personal organisation and presentation skills; group projects also require group organisational and communication skills.

Private study makes up more than half of the time allocated to each module. Students are expected to read around the topics of each module and follow especially any directed reading from recommended course texts. Private study will include further investigations prior to exercises or projects and also consolidation of lecture notes.

Individual industrial placement: The penultimate year of the degree is spent in industry. This provides students with experience of working in a company relevant to the degree, consolidates knowledge gained during their academic studies, and enhances their understanding of how to apply this in practice. It also provides students with opportunities to develop professionally and plan for further development towards a professional qualification.

15. Assessment and feedback methods *(this should include the range of types of methods used)*

Modules may be assessed by formal examination, by assignments, by an individual or group project, or by some combination of these methods.

Examinations can be either paper-based or online, of varying durations but typically 1- or 2-hours long. Question types include (but are not limited to): long-answer written responses, calculated answers, and

multiple choice. Typically, questions will require students both to recall knowledge and apply that knowledge to actively solve or analyse a problem.

Assignments are typically 10-20 hour pieces of continuously assessed coursework, which students complete individually or in groups as directed. An assignment may have multiple stages, each offered over a 2-3 week period, delivered to separate deadlines. Assignments typically assess practical skills.

Individual dissertation projects are completed at Level 3, over two semesters. Students select a topic, research the background literature, prepare a survey/analysis report at the interim assessment stage, and apply this knowledge in a practical, problem-solving project which typically involves the design, implementation and testing of a substantial piece of software. The final assessment stage is by dissertation and presentation session, and the overall assessment is conducted independently by two examiners. A third examiner may be included if the two examiners have divergent opinions of the work, and a *viva voce* examination may be held if there are any questions about the student's work.

Group projects are completed at different levels in the programme, over one or two semesters. Teams undertake software engineering projects and are assessed by a portfolio of evidence including things such as: analysis and design documents, a tested, working software system, and evidence of team practice (such as timesheets, and meeting agendas and minutes). The software engineering challenge is in some cases specified by staff, or in some cases teams must work with an external client to understand a real-world problem. Credit is awarded to the team as a whole on the basis of the quality of the portfolio of work, as evidenced in elements such as the final deliverable, the interim documentation and reported client satisfaction. Credit is weighted towards individual team members based on their participation, as evidenced by peer assessment which is moderated by staff members.

Industrial placement – A variety of methods are used to assess the placement undertaken in the penultimate year. These include two written reports describing and reflecting on personal professional development through experience gained in the year in industry (from an online skills-based placement journal), and an oral presentation to their peers and staff on return from the year in industry.

16. Programme structure, progression and assessment regulations

16a. Standard Programme Information (pre-populated for all programmes)

All programmes are expected to adhere to the University of Sheffield's General Regulations. Details of the University's General Regulations can be found here: http://www.sheffield.ac.uk/calendar/

Details of the programme structure and current modules can be found here:

https://www.sheffield.ac.uk/calendar/regs

Further information about studying at The University of Sheffield can be accessed via our web pages at:

https://www.sheffield.ac.uk/study

16b. Progression and assessment requirements *(this should capture information about e.g. progression hurdles, PSRB requirements, resit of component parts, module capping etc)*

Our degree programmes are designed with a shared core curriculum at Levels 1-2, broadening out into many advanced study paths at Level 3. The common core satisfies the requirements for accreditation by BCS, the Chartered Institute for IT. This common core also permits direct transfers between the related degrees in *Computer Science, Computer Science (Software Engineering)* and *Computer Science* (*Artificial Intelligence)* during the first two years of study.

Accreditation requirements place extra restrictions on the university's standard rules for progression. In particular, BCS accreditation follows the Engineering Council's rules that "A maximum of 30 credits in a

Bachelors or integrated Masters' degree programme can be compensated, and a maximum of 20 credits in a Masters' degree other than the integrated Masters' degree," where *compensation* is defined as "The practice of allowing marginal failure (i.e., not more than 10% below the nominal pass mark) of one or more modules and awarding credit for them, often on the basis of good overall academic performance."

A student who has been awarded credit in compensation of failed units in Years 2 and beyond will be permitted an additional resit attempt in the compensated credits provided the student has met the requirements for progression (or completion) under the General Regulations for First Degrees.

For modules COM1001 Introduction to Software Engineering and COM213 Software Hut, the teamwork requirements of the modules mean that the opportunity to resit will only be available in the following academic year. For other modules a resit assessment will usually be offered in the summer resit period.

A student who does not pass COM3610 Dissertation Project at the first attempt will be ineligible for the award of the Degree of Honours. This is also a requirement for an accredited degree.

To satisfy the requirements for accreditation, COM310 Cybersecurity in Action and MGT388 are must-pass modules.

The industrial placement is assessed on a pass/fail basis and does not contribute to the degree classification: a pass in this placement year is required in order for the degree title to reflect the industrial experience.

Where a student has met the requirements for the award of a degree under the General Regulations for First Degrees but not the requirements for the award of a degree accredited under Engineering Council rules, they may be recommended for the award of a BEng degree in Computer Science and Software Engineering Studies.

17. University scheme on optional Year Abroad or Placement Year

Schools should indicate here if students on this programme cannot apply for the University scheme(s) for an optional year abroad and / or placement year

This programme includes a mandatory industrial placement year. Therefore, an additional placement year would not be appropriate.

Version Number:	Purpose / Change:	Date:
1	Major amendment	November 2024