University of Sheffield

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.

The content of our courses is reviewed annually to make sure it's up-to-date and relevant. Individual modules are occasionally updated or withdrawn. This is in response to discoveries through our world-leading research; funding changes; professional accreditation requirements; student or employer feedback; outcomes of reviews; and variations in staff or student numbers. In the event of any change we will inform students and take reasonable steps to minimise disruption.

Programme Details

1. Programme title	Computer Science (Software Engineering)		
2. Award type	Bachelor of Engineering		
3. Programme details	**FHEQ Level:** 6	**Mode of Study:** Full time Full time	**Duration:** 3 years 4 years (Foundation)
4. Faculty	Faculty of Engineering		
5. School	**Owning:** School of Computer Science		
6. Accrediting Professional or Statutory Body	British Computer Society (BSC)		
7. HECoS code *Select between one and three codes from the HECoS vocabulary.*	**Code:** 100374 **Percentage:** 100	**Code:** **Percentage:**	**Code:** **Percentage:**
Programme code (internal use)	COMU05 (Full time) COMU57 (Foundation)		

9. Programme aims

The programme aims to:	
A1	Provide a thorough academic grounding in the core subject matter of Computer Science, 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 Computer Science and engage students in advanced research areas and methods.
A5	Produce immediately employable graduates with an industrially relevant mix of knowledge, practical skills and self-motivation and with leadership and enterprise skills.

10. 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.
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.
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.

11. Learning and teaching methods *(this should include a summary of methods used throughout the programme, including any unique features and should be written with a student focus as this information will display to current students and applicants i.e. prospectus)*

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 tutorials, and project supervision. Additional support is provided in the first year of study, through optional drop-in sessions, to support the transition to self-directed learning.

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, as well as in-class discussions.

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, focussing on a particular topic, often describing an area of their current research or business. These are typically more interactive sessions than a traditional lecture, encouraging discussions and input from all participants.

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 and **tutorials** are sessions conducted by a staff member or teaching assistant 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. The supervisor's role is to provide guidance to enable the student or group to complete their project. 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.

Projects are undertaken individually and 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.

12. Assessment and feedback methods *(this should include the range of types of methods used and should be written with a student focus as this information will display to current students and applicants i.e. prospectus)*

Assessment types include: examination, assignment, report, classroom testing, presentation, individual project, or group project. It is very common for a module to have multiple assessment points of different types.

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.

Reports are written documents that provide critical analysis of a scenario, results or personal experience.

Classroom testing can take multiple forms. Most common is an online quiz, similar to an examination but usually shorter and carrying less overall weight. In some cases it can involve completion of a task, such as a programming assignment.

Presentations can be recorded or in person. They are typically formally-structured oral reports on a particular topic.

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. These are usually software engineering projects, with one research-oriented group project for students who complete an integrated Masters programme. In software engineering projects they are assessed by a portfolio of evidence including things such as: analysis and design documents, the quality of testing and implementation of the software, 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 or develop the project through lean startup practices. For the research-oriented project, teams tackle a research problem, developing a research output such as a published paper or report, and a similar portfolio of evidence is used to that in a software engineering project. In both cases, 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.

Feedback takes two forms: **formative** feedback, where students get constructive feedback on their work that does not count towards their grades, and **summative** feedback, where students get feedback on assessed work after it has been graded. Formative feedback is available in the less structured classes (such as laboratories and problem solving classes) as well as at timetabled drop-in sessions and through communication with module staff before summative assessment. Summative feedback is provided for submitted work such as assignments and projects when they are graded.

Version Number:	**Purpose / Change:**	**Cohort affected: (academic year and level)**	**Date change approved:**
1	Major Amendment		November 2024
2	Programme Simplification	25/26 - Foundation 26/27 - Year 1 & 2	June 2025

END OF DOCUMENT