Programme Specification[1]

1. Programme Title(s) and Code(s):

Programme Title	*UCAS Code*	*GU Code*
BSc Honours in Software Engineering	G430	G430-2208
BSc Honours in Software Engineering (Faster Route)	G601-2208	

2. Academic Session:

2018-19

3. SCQF Level (see Scottish Credit and Qualifications Framework Levels):

10

4. Credits:

490

5. Entrance Requirements:

Please refer to the current undergraduate prospectus at:
http://www.gla.ac.uk/undergraduate/

6. ATAS Certificate Requirement (see Academic Technology Approval Scheme):

ATAS Certificate not required

7. Attendance Type:

Full Time

8. Programme Aims:

The focus of the Software Engineering degree is on topics directly relevant to the development of large and complex software systems. Initially this programme shares the fundamentals with the Computing Science Single Honours Degree. It becomes more specialised from year 3 with a focus on software design and implementation in the Team Project and subsequently in the choice of Level 4 Electives and Individual Project. This programme also includes a formally supervised and assessed industrial summer placement between Level 3 and Level 4.

[1] This 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 full advantage is taken of the learning opportunities that are provided. More detailed information on the learning outcomes, content and teaching, learning and assessment methods of each course can be found in course handbooks and other programme documentation and online at www.gla.ac.uk/

The accuracy of the information in this document is reviewed periodically by the University and may be checked by the Quality Assurance Agency for Higher Education.

This degree programme aims to:

- provide students with a deep understanding of the theory and practice of computing;
- give students the opportunity to study a broad range of core computing science topics;
- encourage students to discover the connections among these topics and to understand their common theoretical foundations;
- produce graduates fit to occupy responsible positions in the software industry;
- expose students to software engineering in an industrial context via summer work placement;
- give students the opportunity to choose selected Software Engineering topics to study in considerable depth thereby equipping the best graduates to enter research programmes;
- emphasise unchanging principles in computing science;
- encourage independent study habits that will stand graduates in good stead throughout their professional careers;
- enable students to enhance their transferable and interpersonal skills, particularly written and oral communication and team working.

9. Intended Learning Outcomes of Programme:

The programme provides opportunities for students to develop and demonstrate knowledge and understanding, skills, qualities and other attributes in the following areas.

At the end of the programme students should be able to demonstrate a deep understanding of advanced topics chosen by the student, such as:

Information Engineering	Students should be able to: - relate how humans interact with computers; - identify design, construction and evaluation techniques for human-computer interfaces; - reproduce details related to the collection, organisation, manipulation, communication, and display of information by computers; - discuss the design and operation of database systems; - use database technologies; - apply human-computer interaction principles, information retrieval techniques; - relate information systems principles; and - define multimedia usage details. -
Systems Architecture	Students should be able to: - explain computer architecture and networking principles, at all levels of abstraction from the system level down to the gate level; - employ concurrent and distributed computation techniques and practice their specification and implementation in hardware and software; - discuss and diagram hardware architecture, computer-based systems, computer communications, computer networks, concurrency and parallelism, distributed computer systems and operating systems.
Programming Languages	Students should be able to: - recall the fundamental concepts underlying imperative, object-oriented, functional and concurrent programming languages; principles of language design and specification; knowledge of implementation techniques; - apply programming fundamentals, analyse comparative programming languages, compilers and use syntax-directed

	tools.
Software Engineering	Students should be able to: - recall the principles underlying the building and maintenance of large software artefacts; - apply modern software engineering methods and to use specific software engineering tools; - relate the strengths and weaknesses of formal and informal software engineering methods; - define middleware; - perform systems analysis and design.
Theoretical Foundations	Students should be able to: - recall discrete mathematics principles; - analyse algorithms and their complexity; - tell how these underpin other areas of the subject; - experiment with data structures and algorithms.

At the end of the programme students should be able to demonstrate advanced skills dependent on topics chosen by the student, such as the ability to:

- program in several imperative, object-oriented, functional and concurrent programming languages; a thorough mastery of a least one of these languages;
- engineer substantial software systems through all stages of their life cycle, namely problem analysis, requirements, design, specification, construction, testing and modification;
- evaluate systems in terms of general quality attributes and possible trade-offs presented within the given problem;
- recognise any risks or safety aspects that may be involved in the operation of computing equipment within a given context;
- specify and implement concurrent and distributed computation.

At the end of the programme students should be able to demonstrate a deep understanding of advanced topics chosen by the student, which will lead to the ability to:

- debate the strengths and weaknesses of different programming languages;
- distinguish social, professional and ethical implications of the use of computers and software development;
- model computer-based systems for the purpose of comprehension, communication, prediction and the understanding of trade-offs;
- critically evaluate and test the extent to which a computer-based system meets the criteria defined for its current use and future development;
- make design decisions based on appropriate correctness and efficiency considerations;
- use formal and semi-formal methods in the analysis and verification of software.

At the end of the programme students should be able to:

- work individually, including managing learning and development, making use of time management and organisational skills;
- work in teams, recognising the different roles team members adopt;
- present succinctly rational and reasoned arguments (orally or in writing);
- effectively perform information-retrieval tasks (e.g. using search engines and catalogues);
- present cases involving a quantitative dimension thus demonstrating basic numeracy; and
- effectively use general IT facilities

10. Typical Learning and Teaching Approaches:

- Knowledge and understanding

  Contact with teaching staff is through lectures, large and small-group tutorials, workshop and laboratory sessions. The majority of staff provide handouts in lectures. Students are expected to augment these with their own notes, using these as a basis for further regular study during the course. Formatively assessed tutorial and laboratory exercises give students the opportunity to exercise their developing knowledge and understanding. Feedback on exercises is given individually or collectively.
- Practical, discipline-specific, skills

Demonstrations are given and case-studies examined in lectures, workshops and tutorials.

Extensive coursework exercises in the early years support the development of programming skills. Major team and individual project work develops particularly the ability to evaluate systems, recognise risks or safety aspects in the operation of computing equipment within a given context, and the ability to engineer substantial software systems through all stages of their life cycle. Subject coursework, in many different styles, often encourages development of all these skills.

The summer placement gives the students the opportunity to exercise core skills learned, on the first three years of the programme, in a commercial environment. It also motivates their further study of software engineering in the final year of the programme.

- Intellectual (thinking) skills

Lectures introduce these skills, which are developed using major project work, coursework, workshops. An awareness of social, professional and ethical implications of the use of computer applications and software development is developed using a series of group exercises, including debates, IT news analysis.

- Transferable skills

Lectures and workshops introduce time management, reflection and communication, and organisation and planning, and these are developed extensively in major project work in particular.

Numeracy in both understanding and presenting cases is developed as required in major and minor project work. Effective information retrieval skills are developed during major project work and coursework exercises. The effective use of IT facilities is developed as a side effect of all practical work.

The industrial placement provides practical experience of time management, working to deadlines and developing interpersonal skills by working with professional software engineers on real projects.

11. Typical Assessment Methods:

- Knowledge and understanding

Unseen examinations, consisting principally of short-answer questions with some essay-style questions. Assessed coursework in the form of tutorial exercises and reports of laboratory activity.

- Practical, discipline-specific, skills

Practical seen examination. Assessed coursework exercises will each assess various subsets of these skills. Major project work is assessed using a final written report, a demonstration and an oral presentation.

- Intellectual (thinking) skills

Major project report assesses all skills. Assessed coursework may assess strengths and weakness of different programming paradigms, modelling computer based systems for the purposes of comprehension, communication, prediction and understanding trade-offs and risks. An awareness of social, professional and ethical implications of the use of computer applications and software development is assessed via individual performance in group work, and by unseen essay on topical issues.

- Transferable skills

Major team and individual projects, as well as some coursework exercises, assess the ability to work individually or in teams, including managing learning and development, time management and organisation skills and the different roles team members adopt. The Professional Skills & Issues course assesses reflection and communication and effective information-retrieval skills.

Reporting on the industrial placement assesses the ability to describe an independent piece of work both in writing, via the placement report, and verbally, through the presentation. It also assesses the student's ability to reflect on how their academic work relates to the practice of software engineering.

12. Programme Structure and Features:

The Single Honours degree programme extends over four years of full-time study.

A candidate for the Honours degree must obtain a minimum of 490 credits, 240 of which must be awarded for Honours courses.

Level 1

There are three sets of courses currently offered at level 1. Either set enables students to continue to Honours level:

Set 1: aimed at students with prior programming experience; 40 credits of CS out of 120.

Set 2: aimed at students with no prior programming experience; 40 credits of CS out of 120. A student who chooses set 2 in Level 1 will need to take Computing Science1F (COMPSCI1006) (10 credits) in Level 2.

Set 3: aimed at students with no prior programming experience; 50 credits of CS out of 120.

Students will be strongly encouraged to include 40 credits of Level 1 Mathematics in year 1 or 2.

Course Title	Course Code	Credits	Core	Optional	Semester(s) taught
SET 1 [40 credits]					
Computing Science 1P	COMPSCI1001	20	X		1 & 2
Computing Science 1F	COMPSCI1006	10	X		1
Computing Science 1S	COMPSCI1018	10	X		2
Other subjects (Level 1, 80 credits)					
SET 2 [40 credits]					
Computing Science 1CT	COMPSCI1016	20	X		1
Computing Science 1PX	COMPSCI1017	10	X		2
Computing Science 1S	COMPSCI1018	10	X		2
Other subjects (Level 1, 80 credits)					
SET 3 [50 credits]					
Computing Science 1CT	COMPSCI1016	20	X		1
Computing Science 1F	COMPSCI1006	10	X		1
Computing Science 1PX	COMPSCI1017	10	X		2
Computing Science 1S	COMPSCI1018	10	X		2
Other subjects (Level 1, 70 credits)					

Level 2

Level 2 entry is guaranteed to students who achieve an average grade of B3 or better in their Level 1 CS courses at first sitting. Entry is not guaranteed to students with an average grade of C3 or better in their Level 1 CS courses at first sitting but may be permitted at the discretion of the School.

In either case, all grades must be at D3 or better – students who have gained a sufficient average grade at first sitting must resit to improve any grade below D3.

Course Title	Course Code	Credits	Core	Optional	Semester(s) taught
Java Programming 2	COMPSCI2001	10	X		1
Object Oriented Software Engineering 2	COMPSCI2008	10	X		2
Algorithmic Foundations 2	COMPSCI2003	10	X		1
Networks & Operating Systems Essentials 2	COMPSCI2024	10	X		1
Algorithms & Data Structures 2	COMPSCI2007	10	X		2
Web Application Development 2	COMPSCI2021	10	X		2
Other subjects (60 credits)					

Computing Science 1F (COMPSCI1006) (Level 1, 10 credits) (semester 1) is required to be taken by any student who has done set 2 in Level 1.

Level 3

Honours students in Science must achieve a grade point average of 12 over 60 credits of Level 2 courses in the subject of their Honours Programme at the first attempt.

Students who do not meet the requirements for entry to our Honours degree programmes may be eligible for entry to the Designated Degree in Computing Science (CS3). Such students must satisfy the progression requirements in Parts 10 and 11 of the Generic Undergraduate Regulations and the requirements of Part 3 of the Supplementary Regulations for the Degree of Bachelor of Science, as set out by the College of Science and Engineering, and must also meet the following additional requirement from the School of Computing Science.

Honours Entry Guaranteed: minimum average grade of B3 over all Level 2 Computing Science courses at first attempt. At School discretion: minimum average grade of C3 over all Level 2 Computing Science courses at first attempt. Entry to the SE3H class is competitive, and only a limited number of places are available for the best students.

Students in the first semester of Level 3 take a fixed curriculum designed to give breadth in the subject:

Course Title	Course Code	Credits	Core	Optional	Semester(s) taught
Systems Programming (H)	COMPSCI4081	10	X		1
Algorithmics I (H)	COMPSCI4009	10	X		1
Interactive Systems (H)	COMPSCI4014	10	X		1
Data Fundamentals (H)	COMPSCI4073	10	X		1
They also take the following compulsory courses:					
Professional Software Development (H)	COMPSCI4015	10	X		1 & 2
Team Project (H)	COMPSCI4047	30	X		1 & 2
Software Engineering Summer Placement	COMPSCI4046	10	X		Summer between Levels 3&4

Students at Level 3 must also take 4 courses from the current Level H and M Electives listed in the course catalogue, subject to meeting any pre-requisites:
http://www.gla.ac.uk/coursecatalogue/courselist/?code=REG30200000&name=School+of+Computing+Science

Level 4

Entry to Level 4 is dependent on the student achieving an aggregate score of 9 (on University 22 point scale) in Level 3 at the first attempt.

Students failing to achieve the minimal level for progression will be assessed for the early exit qualification of BSc in Software Engineering based on their results in Level 3, including completion of the Software Engineering Summer Placement (COMPSCI4046).

Software Engineering Honours students select seven 10-credit subject courses. The courses available can vary each year depending on staff availability and resources. The list of level H and M courses currently available are listed in the course catalogue (see link above).

Of these seven, at least ONE course must be chosen from the list below.

Course Title	Course Code	Credits	Core	Optional	Semester(s) taught
IT Architecture (M)	COMPSCI5013	10		X	2
Advanced Software Engineering Practices (H)	COMPSCI4071	10		X	1 & 2

Students must also take the following compulsory courses:					
Individual Project (H)	COMPSCI4025P	40	X		1 & 2
Professional Skills and Issues (H)	COMPSCI4038	10	X		1
Please note that students must choose ONE security-related course from the following list in either Level 3 or Level 4:					
Cyber Security Fundamentals (H)	COMPSCI4062	10		X	2
Human Centred Security (M)	COMPSCI5060	10		X	2
Safety Critical Systems (H)	COMPSCI4045	10		X	2
Enterprise Cyber Security (M)	COMPSCI5077	10		X	1
Cyber System Forensics (M)	COMPSCI5080	10		X	2

Honours Assessment

Within each year, courses are weighted according to credits. The final Honours assessment is based on 40% of the level 3 aggregated score combined with 60% of the level 4 aggregated score.

Faster Route in Software Engineering

Students taking this route will come straight into level 2. Of their 120 credits, they will take the following courses totalling 80 credits in Computing Science:

Course Title	Course Code	Credits	Core	Optional	Semester(s) taught
Computing Science 1F	COMPSCI1006	10	X		1
Computing Science 1S	COMPSCI1018	10	X		2
Java Programming 2	COMPSCI2001	10	X		1
Object Oriented Software Engineering 2	COMPSCI2008	10	X		2
Algorithmic Foundations 2	COMPSCI2003	10	X		1
Algorithms and Data Structures 2	COMPSCI2007	10	X		2
Networks and Operating Systems Essentials 2	COMPSCI2024	10	X		1
Web Application Development 2	COMPSCI2021	10	X		2
Other subjects from level 1 or 2, totalling 40 credits					

It is strongly recommended that 40 credits of level 1 Mathematics are taken, unless the student has an equivalent mathematics qualification on entry.

Faster Route entry to Honours
Entry requirements same as standard route.

Note that from level 3 onwards, the Faster Route programme is exactly the same as the standard Honours programme.

BSc Designated Degree in Computing Science

Students who do not meet the progression requirements for the BSc Honours in Software Engineering may be eligible to progress to the Designated Degree in Computing Science. Please refer to the BSc Honours in Computing Science programme specification for further information.

For more information on courses see the University course catalogue:
http://www.gla.ac.uk/coursecatalogue/

Regulations
This programme will be governed by the relevant regulations published in the University Calendar. These regulations include the requirements in relation to:

(a) Award of the degree
(b) Progress
(c) Early exit awards
(d) (For undergraduate programmes, where appropriate) Entry to Honours

https://www.gla.ac.uk/myglasgow/senateoffice/policies/calendar/calendar2018-19/

13. Programme Accredited By:

BCS The Chartered Institute for IT

14. Location(s):

Glasgow

15. College:

College of Science and Engineering

16. Lead School/Institute:

Computing Science [REG30200000]

17. Is this programme collaborative with another institution:

No

18. Awarding Institution(s):

University of Glasgow

19. Teaching Institution(s):

20. Language of Instruction:

English

21. Language of Assessment:

English

22. Relevant QAA Subject Benchmark Statements (see Quality Assurance Agency for Higher Education) and Other External or Internal Reference Points:

The following web links introduce the benchmarks that are used to guide and assess our programmes. We monitor our courses against these on a regular basis, further information about this process and about recent developments in these benchmarks can be obtained direct from the school. http://www.qaa.ac.uk/docs/qaa/subject-benchmark-statements/sbs-computing-16.pdf?sfvrsn=26e1f781_12 http://www.theiet.org/careers/profreg/ http://www.bcs.org/server.php?show=nav.7065

23. Additional Relevant Information (if applicable):

Support for students is provided by the Postgraduate/Undergraduate Adviser(s) of Studies supported by University resources such LEADS (www.gla.ac.uk/myglasgow/leads/), Counselling & Psychological Services (www.gla.ac.uk/services/counselling/), the Disability Service (www.gla.ac.uk/services/studentdisability/) and the Careers Service (www.gla.ac.uk/services/careers/).

24. Online Learning:

No

25. Date of approval:	09/08/2018