Programme Specification[1]

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

Programme Title	*UCAS Code*	*GU Code*
BSc Honours in Electronic & Software Engineering	GH66	GH66-2208

2. Academic Session:

2018-19

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

10

4. Credits:

500

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:

This degree programme aims to:

- provide students with a deep understanding of the theory and practice of computing and electronic engineering;
- give students the opportunity to study a broad range of core computing science and electronic

[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.

engineering 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 and electronics industries;
- expose students to hardware-software engineering in an industrial context via summer work placement;
- give students the opportunity to choose selected topics to study in considerable depth thereby equipping the best graduates to enter research programmes;
- emphasise unchanging principles in computing science and electronic engineering;
- 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 focus of the Electronic and Software Engineering degree is on topics directly relevant to the development of large hardware-software systems, including embedded systems. Graduates will be particularly well equipped to occupy responsible positions within the electronics and information technology sectors.

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

Information Engineering	- 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 computer vision and image processing techniques, database technologies, graphics and sound technologies; - apply human-computer interaction principles, information retrieval techniques; - relate information systems principles; and - define multimedia usage details.
Systems Architecture	- 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	- recall the fundamental concepts underlying imperative, object-oriented, functional and concurrent programming languages.
Software Engineering	- 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	- recall discrete mathematics principles; - analyse algorithms and their complexity; - tell how these underpin other areas of the subject; - recall and explain the fundamental mathematical and scientific concepts on which modern electronics is based, including: complex numbers, matrix algebra, the solution of differential equations, transform theory and logic; - illustrate the fundamental principles which determine how electronic circuits operate—from Ohm's Law and Kirchhoff's Laws to Maxwell's Equations—in realistic engineering situations; - summarise the fundamental principles and engineering practice employed in the construction of complex electronic systems, including: biased active devices, integrated circuits, systems employing mixed hardware and software.
Electronic Engineering	- plan, safely execute, and critically analyse the results of experimental activities; - read, interpret and be aware of the limitations of technical literature and documentation used by professional engineers; - design, construct and test a range of electronic circuits and systems; - use standard electronics measurement and test equipment effectively; - apply software packages in an engineering context—from key office applications (word processing, spreadsheets) to simulation and design tools (SPICE, Matlab, Mathematica, PCB layout, compilers, HDL simulators); - develop computer programs in an engineering context, in both high and low level languages, for the monitoring and control of electronic systems.

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;
- analyse engineering systems—primarily electronic systems—using mathematical and scientific principles;
- design novel electronic systems to meet a given specification and constraints, including a critical analysis of the degree to which the specification can be met;
- evaluate experimental and other numerical data, and use such evaluations to improve design choices.

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;
- critically evaluate a design or product and make improvements, informed by an awareness of the limitations of current technology, user needs and industry standards;
- select appropriate software tools to aid in the analysis and design of circuits and systems;
- explain the constraints within which engineering judgement is exercised in industry and in society, the relationship of electronics to other engineering disciplines and the professional and ethical responsibilities of engineers;
- be creative in the solution of engineering problems and in the development of designs.

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.
- 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.

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 electronic and software engineering in the final year of the programme.

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.

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.

12. Programme Structure and Features:

Structure

The Single Honours degree programme extends over four years of full-time study. A candidate for the Honours degree must obtain a minimum of 500 credits, 240 of which must be awarded for Honours courses.

Level 1

There are two 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; 50 credits of CS out of 120.

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 [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
Students must also take the following courses:					
Electronic Engineering 1X	ENG1021	20	X		1
Electronic Engineering 1Y	ENG1022	20	X		2
Engineering Mathematics 1 (or Maths1R/Maths1S or Maths1X/Maths1Y	ENG1063	40	X		1 & 2

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	CourseCode	Credits	Core	Optional	Semester(s) taught
Java Programming 2	COMPSCI2001	10	X		1
Object Oriented Software Engineering 2	COMPSCI2008	10	X		2
Algorithms & Data Structures 2	COMPSCI2007	10	X		2
Networks & Operating Systems Essentials 2	COMPSCI2024	10	X		1
Web Application Development 2	COMPSCI2021	10	X		2
Digital Electronics 2	ENG2020	10	X		1
Electrical Circuits 2	ENG2023	10	X		2
Analogue Electronics 2	ENG2004	10	X		2
Embedded Processors 2	ENG2029	10	X		2
Electronic Design Project 2	ENG2025	10	X		1 & 2
Engineering Mathematics 2	ENG2086	20	X		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 (15 on University 22 point scale) over the four pre-requisite Level 2 Computing Science courses (Algorithms and Data Structures 2, Computer Systems 2, Java and Object Oriented Software Engineering 2 and Web Application Development 2) at first attempt. At School discretion: minimum average grade of C3 (12 on University 22 point scale) over the same four courses at first attempt. In addition, the student must meet the requirements of the School of Engineering.

Electronic and Software Engineering Students who achieve a minimum average of B3 over the pre-requisite level 2 Computing Science courses, and having achieved a B3 or better in Engineering Mathematics 2 may transfer to CS3H or SE3H or MobSE3H. Those with a C3 average and C3 or better in Engineering Mathematics 2 may transfer to CS3H, SE3H or MobSE3H at the discretion of the School.

Electronic and Software Engineering Students who achieve a C3 average over the pre-requisite level 2 Computing Science courses and Engineering Mathematics 2, but who are not permitted to transfer at the School's discretion to CS3H, SE3H, or MobSE3H, may transfer to CS3.

Course Title	Course Code	Credits	Core	Optional	Semester(s) taught
Systems Programming (H)	COMPSCI4081	10	X		1
Networked Systems (H)	COMPSCI4012	10	X		2
Operating Systems (H)	COMPSCI4011	10	X		2
Professional Software Development (H)	COMPSCI4015	10	X		1 & 2
Communications Systems 3	ENG3014	10	X		1

Control EE3	ENG3015	10	X		2
Electronic Systems Design 3	ENG3026	10	X		1
Real-Time Computer Systems 3	ENG3043	10	X		1
As well as either:					
Digital Circuit Design 3	ENG3020	10		X	2
OR					
Simulation of Eng Systems 3	ENG3036	10		X	1
Students also take:					
ESE Team Project	COMPSCI4044P	30	X		1 & 2
Software Engineering Summer Placement	COMPSCI4046	10	X		Summer between Levels 3 & 4

Level 4

To progress to Level 4, a student must normally achieve a GPA of at least 9 (on University 22 point scale) in the Level 3 Computing Science courses at the first attempt, achieve at least D3 or better in the Team Project and fulfil the requirements of the School of Engineering.

Students failing to achieve the minimal level for progression will be assessed for the early exit qualification of BSc in Electronic and Software Engineering based on their results in level 3, including the industrial summer placement (370 credits in total).

Honours students in Level 4 take three 10-credit CS courses chosen as follows:

Course Title	Course Code	Credits	Core	Optional	Semester(s) taught
At least ONE from:					
IT Architecture (M)	COMPSCI5013	10		X	2
Advanced Software Engineering Practices (H)	COMPSCI4071	10		X	1 & 2
Plus TWO other 10 credit course from any other of Computing Science Level H and M courses to make up the total 30 credits. The list of courses available can vary each year depending on staff availability and resources. The list of courses currently available can be found in the course catalogue: http://www.gla.ac.uk/coursecatalogue/courselist/?code=REG30200000&name=School+of+Computing+Science					
Students should choose 40 credits from the optional courses below:					
Acoustics and Audio Technology 4	ENG4001	20		X	2
Biosensors and Diagnostics 4	ENG4036	10		X	2
Control 4	ENG4042	20		X	1
Digital Communication 4	ENG4052	20		X	1 & 2
Digital Signal Processing 4	ENG4053	20		X	1
Advanced Devices 4	ENG4099	20		X	2
Microwaves and Optical Transmission Systems 4	ENG4100	20		X	2
Power Systems 4	ENG4104	20		X	2
Robotics 4	ENG4118	20		X	2
VLSI Design 4	ENG4138	20		X	1
Biophysics of Cells and Systems 4	ENG4181	10		X	1
Navigation Systems 4	ENG4184	10		X	1
Radar and Electro-Optic Systems	ENG4185	10		X	2

4					
Power Electronics and Drives 4	ENG4187	20		X	1

In addition, ESE students undertake the following compulsory courses:					
ESE Final Year Project 4	ENG4066P	40	X		1 & 2
Professional Skills and Issues (H)	COMPSCI4038	10	X		1

Honours Assessment
Within each year, courses are weighted according to credits. The final Honours assessment is based on 35% of the level 3 aggregated score combined with 65% of the level 4 aggregated score.

For more information on courses see the 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