QAA MEMBERSHIP

QAA

Subject Benchmark Statement

Computing

October 2019

Contents

How can I use this document?

This is the Subject Benchmark Statement for Computing. It defines the academic standards that can be expected of a graduate, in terms of what they might know, do and understand at the end of their studies, and describes the nature of the subject.

The UK Quality Code for Higher Education (Quality Code) sets out the Expectations and Core practices that all providers of UK higher education are required to meet. Providers in Scotland, Wales and Northern Ireland must also meet the Common practices in the Quality Code.

The Quality Assurance Agency for Higher Education (QAA) has also published a set of Advice and Guidance, divided into 12 themes, and a number of other resources that support the mandatory part of the Quality Code. Subject Benchmark Statements sit alongside these resources to help providers develop courses and refine curricula but are not part of the regulated requirements for higher education providers in the UK.

This Statement is intended to support you if you are:

- involved in the design, delivery and review of courses of study in computing or related subjects
- a prospective student thinking about studying this subject, or a current student of the subject, to find out what may be involved
- an employer, to find out about the knowledge and skills generally expected of a graduate in this subject.

Subject Benchmark Statements provide general guidance for articulating the learning outcomes associated with the course but are not intended to represent a national curriculum in a subject or to prescribe set approaches to teaching, learning or assessment. Instead, they allow for flexibility and innovation in course design within a framework agreed by the subject community.

It may be helpful to refer to relevant Advice and Guidance when using this statement.

Explanations of unfamiliar terms used in this Subject Benchmark Statement can be found in QAA's Glossary.

About the Statement

This Subject Benchmark Statement refers to bachelor's degrees with honours[1] in Computing.

It has been produced by a group of subject specialists drawn from, and acting on behalf of, the subject community. The process is facilitated by QAA, as is the full consultation with the wider academic community and stakeholder groups each Statement goes through.

In order to ensure the continuing currency of Subject Benchmark Statements, QAA initiates regular reviews of their content, five years after first publication, and every seven years subsequently, or in response to significant changes in the discipline.

Relationship to legislation

Higher education providers are responsible for meeting the requirements of legislation and any other regulatory requirements placed upon them, for example by funding bodies. This Statement does not interpret legislation, nor does it incorporate statutory or regulatory requirements. The responsibility for academic standards remains with the higher education provider who awards the degree.

Higher education providers may need to consider other reference points in addition to this Statement in designing, delivering and reviewing courses. These may include requirements set out by professional, statutory and regulatory bodies (PSRBs), and industry or employer expectations.

Sources of information about other requirements and examples of guidance and good practice are signposted within the Subject Benchmark Statement where appropriate. Individual higher education providers will decide how they use this information.

Summary of changes from the previous Subject Benchmark Statement (2016)

This version of the Statement forms its fourth edition, following initial publication of the Subject Benchmark Statement in 2000 and review and revision in 2007 and 2016.

This latest version of the Statement is the consequence of the revision to the UK Quality Code for Higher Education which was published in 2018. It has been revised to update references to the Quality Code and other minor changes within the sector. Changes have been made by QAA and confirmed by the past co-chair of the most recent review group.

There have been minor revisions to the subject specific content namely:

1. the addition of Data Science to the diverse set of degree courses (paragraph 2.7)
2. the inclusion of a reference to probabilistic machine learning and big data sets (paragraph 2.13, second bullet point)
3. for integrated Master’s courses such as MComp, M’Eng and MSci, the inclusion of an explicit reference to the Subject Benchmark Statement for master’s degrees in computing (paragraph 6.10).

[1] Bachelor's degrees are at level 6 in *The Framework for Higher Education Qualifications in England, Wales and Northern Ireland* and level 10 in *The Framework for Qualifications of Higher Education Institutions in Scotland*, as published in The Frameworks for Higher Education Qualifications of UK Degree-Awarding Bodies

4 the ACM Computer Engineering report is now published (2016) and the reference has been updated (Appendix 1)

5 an explicit reference is now made to the ACM (2017) Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity (Appendix 1).

6 the 2008 ACM Computing Curricula Information Technology was updated in 2017 and referenced as Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology (Appendix 1)

1 Introduction

1.1 In as much as human ingenuity and creativity has fostered the rapid development of the discipline of computing in the past, courses in computing should not limit those who will lead the development of the discipline in the future.

1.2 Computing is concerned with the understanding, design and exploitation of computation and computer technology - one of the most significant advances of the twentieth and twenty-first centuries. It is a discipline that:

- blends elegant theories (including those derived from a range of other disciplines such as mathematics, engineering, psychology, graphical design or well-founded experimental insight) with the solution of immediate practical problems
- underpins the development of both small and large scale, secure, reliable and usable systems that support organisational goals
- helps individuals in their everyday lives
- is pervasive, ubiquitous and diversely applied to a range of applications, and important components are often invisible to the naked eye.

1.3 The reasons for studying computing are as diverse as its domains of application. Some students are attracted by the depth and intellectual richness of the theory, others by the possibility of engineering large and complex systems. Many study computing for vocational reasons, or because it gives them the opportunity to explore creative and dynamic technologies. Whatever the perspective, computing can claim characteristics that, while present in other disciplines, are rarely present in such quantities and combinations.

1.4 Computing promotes innovation and creativity. It requires a disciplined approach to problem solving. It approaches design and development through selection from alternative possibilities justified by carefully crafted arguments. It controls complexity first through abstraction and simplification, and then by the integration of components. Above all, it is a product of human ingenuity and provides major intellectual challenges, yet this limits neither the scope of computing nor the complexity of the application domains addressed.

1.5 Computing as a discipline is attractive to innovation, and this can equally arise from the foundational intellectual areas (for example algorithmics and cryptography) as from technology-driven opportunities.

1.6 It is hardly surprising that the diversity of computing is reflected in the varied titles and curricula that higher education providers have given to their computing-related degree courses. While this Statement aims to capture the nature of computing as a discipline, individual higher education providers may need to draw on a wider range of materials and resources, including other Subject Benchmark Statements, to capture fully the specific character of their particular degree courses.

1.7 Computing degrees will continue to evolve in response to developments in the subject area and to reflect changes in the school curriculum. This Statement therefore concentrates on general graduate outcomes and does not specify a core computing curriculum. The curriculum documents from the Association of Computing Machinery (ACM) are widely used as a source of guidance on possible curriculum content.

1.8 Computing degrees often integrate a period of time working within a company (or similar organisation) as an intern or placement student. Placements offer the opportunity for students to apply and validate their learning and skills in the context of the real world and provide early exposure to professional competences.

2 Nature and extent of computing

2.1 The computing discipline is evolving at a rapid rate, touching upon all aspects of life. Computing delivers innovative solutions to problems, and drives technological, economic and social progress.

2.2 Computing as a discipline consists of central elements: mathematics; fundamentals of computation; and realisation of computer systems in both hardware and software.

2.3 Computing graduates apply their understanding, skills, knowledge and experience to create social and economic value by building secure, reliable and usable systems.

2.4 Computing includes aspects that overlap with areas of interest to a number of adjacent subjects. Examples of such areas are:

- engineering, especially parts of electrical and electronic engineering
- physics, with concern for multimedia and device-level development of computing components
- mathematics (logic and theoretical models of computation, stochastic modelling, numerical methods, analysis and optimisation)
- business (information services)
- philosophy, physiology, biology, and psychology (aspects of artificial intelligence)
- linguistics.

2.5 The application of computational techniques across science and engineering has fundamentally affected practices within those disciplines. Therefore, computing is both a rigorous academic discipline in its own right and also facilitates and supports a wide range of other disciplines, from computational physics to computational biology and computational social science.

2.6 The concept of computational thinking is central to the discipline.

> 'Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modelling the relevant aspects of a problem to make it tractable. It is using invariants to describe a system's behaviour succinctly and declaratively'
>
> Wing, J M (2008) Computational thinking and thinking about computing, *Philosophical Transactions of the Royal Society A, Mathematical, Physical and Engineering Sciences*, vol 366, issue 1881

2.7 The term computing applies to an increasingly diverse set of degree courses all based on the foundations of computer science. This Statement identifies computer science, computer engineering, software engineering, information technology, data science and information systems as discipline areas and outlines the content covered by these. A UK computing degree may include subject matter from more than one discipline area.

2.8 Computer science provides the necessary knowledge to understand and build computational systems. Its main characteristics include:

- fundamental computational concepts and algorithmic thinking, including recursive, distributed and parallel possibilities and attention to the benefits and the limitations of these; the role of these in devising approaches to areas of system design, problem solving, artificial intelligence, simulation and computational modelling

- recognition of the relationships between the concepts of requirements, specification, design, programme and data (in all its forms) validation and maintenance, as well as the power of transformation and proof, and the place of these in computing
- understanding the power behind abstraction, the potential of multiple levels of abstraction and the role this plays in computing
- understanding the opportunities for and the potential of automation, but also the proper balance between automation and how humans effectively interact with computers, recognising the role of redundancy, diversity and separation of concerns in achieving reliable, usable and secure systems, often in the presence of uncertainty
- recognising simplicity and elegance as useful concepts and principles.

Generally, these are expressed in the ability to specify, design and write computer programmes.

2.9 The discipline areas of computer engineering, software engineering, information systems and information technology draw upon the fundamentals of computer science and each other.

2.10 Computer engineering is concerned with the realisation of computer science fundamentals in computer hardware. It includes:

- scientific and engineering principles that underpin the design and operation of modern computer hardware and electro-mechanical interfaces
- the understanding of the trade-offs between hardware and software in overall system design
- memory, processors, peripherals, communication and networking
- real-time and embedded systems, mobile devices.

Generally, these are expressed in the ability to understand the construction of, and make best use of, computational devices, interfaces and protocols.

2.11 Software engineering is concerned with the building of software systems. It includes:

- problem definition, specification (including formal specification), design, implementation (including debugging) and maintenance, software testing, change management and documentation
- cybersecurity, including information security, and safety-critical systems
- understanding risk, reliability and scalability of the range of possible options and an appreciation of design trade-offs.

Generally, these are expressed in the ability to create fit for purpose software in a variety of application domains.

2.12 Information technology is concerned with the application of computing technologies to other domains. It includes:

- the selection and application of software and hardware
- integration of components to provide solutions in a variety of application domains
- risk, cybersecurity and service management aspects of IT systems.

Generally, these are expressed in the ability to deliver a computer-based system as a solution to desired needs.

2.13 Data science provides a balance between science, statistics and technology. It includes:

- the selection and application of analytical software tools
- the application of probabilistic machine learning techniques
- applying tools and knowledge to address the challenges of small and large data sets.

2.14 Information systems is concerned with the modelling, codification and storage of data and information for later retrieval and analysis. It includes:

- data management, databases, information modelling, indexing and searching
- systems analysis, system lifecycle and interactions between information systems and other socio-technical systems, including societal and environmental issues.

Generally, these are expressed in the ability to construct systems that acquire, codify, store, transform and transmit information.

2.15 Many higher education providers deliver degrees focused on specific aspects of computing in society, for example computer networking, games, multimedia, and health informatics. These courses count as computing if their content is informed by one or more of the discipline areas listed in paragraph 2.9. The mere fact that computers are deployed to solve problems in a certain area does not itself make that area fall within the field of Computing.

2.16 Additionally, computing is widely taught in joint and interdisciplinary courses for which it may be appropriate to draw on a number of Subject Benchmark Statements. This Statement is the reference point for the computing component of such courses.

2.17 This Statement does not specify a syllabus or include a body of knowledge. The ACM, in conjunction with the Institute of Electrical and Electronics Engineers (IEEE) and other professional societies, maintain (and regularly update) curricula in several areas: computer science, computer engineering, information systems, information technology and software engineering. These documents should be used to inform course design and curriculum content (see Appendix 1).

2.18 The title of a course cannot describe the whole of its content. However, course titles are not divorced from graduate knowledge, skills and abilities. There are natural overlaps between the different identified discipline areas and course specifications indicate careers the course's graduates would be expected to proceed into. So a degree in software engineering could include aspects of computer science (for example, formal methods) and information technology (for example, user advocacy). However, the title and course specification of the degree references the curriculum from the dominant discipline area.

2.19 Course designers, students and employers will need to be aware of this spectrum of course identities, but behind such variation there are key ideas that can be expected to characterise any honours degree course in computing:

- the concept of computational thinking, the recognition of its main elements and the relevance of these to everyday life
- the computing system (including an information system), and the process of
- developing or analysing it is important; understanding of the system and its operation will go deeper than a mere external appreciation of what the system does or the way(s) in which it is used
- the balance of practice and theory such that practical activity is supported by an

understanding of underlying principles.

3 Subject-specific and generic skills

3.1 Students of computing are expected to develop a wide range of skills. These may be divided into three broad categories:

- computing-related cognitive skills
- computing-related practical skills
- generic skills for employability.

3.2 Cognitive, practical and generic skills need to be placed in the context of the course of study as designed by the higher education provider and/or the possible pathways selected by the individual student. There is an implicit interplay between these identified skills within and across these three categories. The extent to which students acquire these skills depends on the emphasis of individual degree courses.

3.3 Computing-related cognitive skills:

i computational thinking, including its relevance to everyday life
ii an understanding of the scientific method and its applications to problem-solving in this area
iii knowledge and understanding: demonstrate knowledge and understanding of essential facts, concepts, principles and theories relating to computing and computer applications as appropriate to the course of study
iv modelling: use such knowledge and understanding in the modelling and design of computer-based systems for the purposes of comprehension, communication, prediction and the understanding of trade-offs
v requirements, practical constraints and computer-based systems (this includes computer systems, information, security, embedded, and distributed systems) in their context: recognise and analyse criteria and specifications appropriate to specific problems, and plan strategies for their solutions
vi critical evaluation and testing: analyse the extent to which a computer-based system meets the criteria defined for its current use and future development
vii methods and tools: deploy appropriate theory, practices and tools for the specification, design, implementation and evaluation of computer-based systems
viii professional considerations: recognise the professional, economic, social, environmental, moral and ethical issues involved in the sustainable exploitation of computer technology and be guided by the adoption of appropriate professional, ethical and legal practices.

3.4 Computing-related practical skills:

i the ability to specify, design and construct reliable, secure and usable computer-based systems
ii the ability to evaluate systems in terms of quality attributes and possible trade-offs presented within the given problem
iii the ability to plan and manage projects to deliver computing systems within constraints of requirements, timescale and budget
iv the ability to recognise any risks and safety aspects that may be involved in the deployment of computing systems within a given context
v the ability to deploy effectively the tools used for the construction and documentation of computer applications, with particular emphasis on understanding the whole process involved in the effective deployment of computers to solve practical problems

vi the ability to critically evaluate and analyse complex problems, including those with incomplete information, and devise appropriate solutions, within the constraints of a budget.

3.5 Generic skills for employability are described below.

i Students are expected to develop a wide range of generic skills to ensure they become effective in the workplace, to the benefit of themselves, their employer and the wider economy. Students who develop generic skills, and are able to evidence and demonstrate such skills, will gain significant advantage when seeking employment. It is the responsibility of higher education providers to provide every student the opportunity to acquire and evidence generic skills; it is the responsibility of the student to make the most of that opportunity.

ii Intellectual skills: critical thinking; making a case; numeracy and literacy; information literacy. The ability to construct well-argued and grammatically correct documents. The ability to locate and retrieve relevant ideas, and ensure these are correctly and accurately referenced and attributed.

iii Self-management: self-awareness and reflection; goal setting and action planning; independence and adaptability; acting on initiative; innovation and creativity. The ability to work unsupervised, plan effectively and meet deadlines, and respond readily to changing situations and priorities.

iv Interaction: reflection and communication; the ability to succinctly present rational and reasoned arguments that address a given problem or opportunity, to a range of audiences (orally, electronically or in writing).

v Team working and management: the ability to recognise and make best use of the skills and knowledge of individuals to collaborate. To be able to identify problems and desired outcomes and negotiate to mutually acceptable conclusions. To understand the role of a leader in setting direction and taking responsibility for actions and decisions.

vi Contextual awareness: the ability to understand and meet the needs of individuals, business and the community, and to understand how workplaces and organisations are governed.

vii Sustainability: recognising factors in environmental and societal contexts relating to the opportunities and challenges created by computing systems across a range of human activities.

4 Teaching, learning and assessment

4.1 Computing courses deploy a diverse range of learning, teaching and assessment methods to enhance and reinforce the student learning experience. This diversity of practice is a strength of the subject. Whichever methods are employed, strategies for teaching, learning and assessment deliver opportunities for the achievement of the learning outcomes, demonstrate the attainment of learning outcomes and recognise the range of student backgrounds.

4.2 Curriculum design is informed by current developments, reflecting appropriate research, scholarship, industrial and business practices, together with an understanding of potential graduate destinations. Students achieve an understanding of Computing through significant exposure to practical coursework and substantial individual and group-project work. Project types include design-and-build, consultancy and research led, which develop both independence of thought and the ability to work effectively in a team. Teaching and learning needs to be placed within the context of social, ethical, legal, professional, environmental and economic factors relevant to computing.

4.3 The following aspects of curriculum delivery have particular relevance to computing degree courses:

- encouraging students to reflect, evaluate, select, justify, communicate and be innovative in their problem-solving
- hands-on learning opportunities, which have particular relevance to many aspects of computing, for example, programming, networking, and physical prototyping
- provision for the development of a range of personal and generic skills
- a major activity allowing students to demonstrate ability in applying practical and analytical skills (as they are present in the course as a whole); this will often take the form of a project carried out in the final year
- computing-related case studies employed to indicate the application of student learning after graduation.

4.4 All forms of work-based learning, including activities such as industrial placements, are seen as a valued part of a course and are properly integrated in terms of preparation of students before this activity, debriefing, building on the experience afterwards and assessment.

4.5 An essential dimension of the educational process is the exposure of students to high-quality software, tools and materials. This conditions the expectations of students and their approach to practice. Access to software and communications facilities enables students to extend their horizons. Apart from exposure to a range of languages covering the different programming paradigms in widespread use, institutions might provide access to tools including graphics packages, computer-aided software engineering (CASE) tools, integrated development environments, theorem provers, project management software, and planning systems, as appropriate to the course of study.

5 Computing degrees as preparation for professional practice

5.1 There are many different types of computing degree course, but all are designed to equip their graduates with knowledge, understanding and skills which will enable them to begin a professional career in some aspect of computing. The possession of a computing degree is seen by many computing employers as an essential indication that these attributes have been achieved. Many computing departments have constituted Industrial Advisory Boards to help them articulate academic and professional practice.

5.2 Some employers may expect graduates to have specific technical skills (for example a particular programming language or tools). Computing courses should not be limited to this. All graduates should have a fundamental ability to adapt and gain additional specific competences after completion of their university learning.

5.3 Not all graduates will proceed with a professional career in computing. The attributes of computing graduates also make them attractive to many non-computing employers: manufacturing, finance, consultancy, public services, creative industries and the arts, as well as entrepreneurs in their own right.

5.4 Within the context of teaching computing to honours degree level, graduates act as the agents to transfer technology and best practice when they ultimately move into employment.

5.5 Computing degrees in the UK may be accredited by UK professional bodies (BCS, The Chartered Institute for IT, and the Institute for Engineering and Technology (IET)) as fulfilling (or partially fulfilling) criteria for professional standing. Such bodies take their own view of the amount of exemption a course delivers to students graduating from it.

5.6 Other professional bodies are involved in accreditation of computing degree courses with particular disciplinary emphasis, for example, Creative Skillset for courses in the creative industries, such as games and visual effects (VFX).

5.7 While neither a curriculum document nor an accreditation requirement, the Skills Framework for the Information Age is a common language for skills and competences in the digital world that many employers use.

6 Benchmark standards

6.1 Benchmark standards are defined at threshold, typical and excellent levels for bachelor's degrees and further comments are made with reference to integrated master's degrees.

The threshold level

6.2 Set here at the bottom of the honours class, the threshold level would be treated by many higher education providers as disappointing performance, given the entry qualifications of their students, and it is not the outcome expected of them.

6.3 On graduating with an honours degree in computing at threshold level, students should be able to:

i demonstrate a requisite understanding of the main body of knowledge for their course of study
ii understand and apply essential concepts, principles and practices of the subject in the context of well-defined scenarios, showing judgement in the selection and application of tools and techniques
iii produce work involving problem identification, the analysis, design and development of a system with accompanying documentation, recognising the important relationships between these stages and showing problem-solving and evaluation skills drawing on supporting evidence
iv produce small, well-constructed programmes to solve well-specified problems
v demonstrate generic skills, an ability to work under guidance and as a team member.
vi identify appropriate practices within a professional, legal and ethical framework and understand the need for continuing professional development.

The typical level

6.4 Set here at the middle of the honours class, this typical level would be treated by many higher education providers as median performance across all students.

6.5 On graduating with an honours degree in computing at typical level, students should be able to:

i demonstrate a sound understanding of the main areas of the body of knowledge within their course of study, with an ability to exercise critical judgement
ii critically analyse and apply essential concepts, principles and practices of the subject in the context of loosely defined scenarios, showing effective judgement in the selection and use of tools and techniques
iii produce work involving problem identification, the analysis, the design or the development of a system, with appropriate documentation, recognising the important relationships between these
iv the work will show problem-solving and evaluation skills, draw upon supporting evidence and demonstrate a good understanding of the need for a high-quality solution
v demonstrate generic skills with an ability to show organised work both as an individual and as a team member and with minimum guidance
vi apply appropriate practices within a professional, legal and ethical framework and identify mechanisms for continuing professional development and lifelong learning.

Excellence

6.6 While the subject benchmark standards in this section are defined for threshold and typical levels, courses in computing will provide opportunities for students to achieve far more highly.

6.7 Such students will be:

i able to contribute significantly to the analysis, design or the development of systems that are complex, recognising the important relationships between these
ii creative and innovative in their application of the principles covered in the curriculum
iii able to exercise critical evaluation and review of both their own work and the work of others
iv able to demonstrate team leadership skills.

Master's level

6.8 Integrated master's degrees (MComp, MEng and MSci) include the outcomes of bachelor's degrees with honours and go beyond them to provide a greater range and depth of specialist knowledge, often within a research and industrial environment, as well as a broader and more general academic base. Such courses provide a foundation for leadership.

6.9 Integrated master's courses of study are designed as an integrated whole from entry to completion, although earlier parts may be delivered in common with a parallel bachelor's degree with honours.

6.10 Further details on integrated master’s degrees can be found in the QAA Subject Benchmark Statement for Computing (Master’s).

Appendix 1: Reference to other curriculum documents

In the decades since the 1960s, roughly every 10 years, the ACM has described curriculum recommendations to the rapidly changing landscape of computer technology. Copies are available at: www.acm.org/education/curricula-recommendations

Computer science
CS 2013: Curriculum Guidelines for Undergraduate Programs in Computer Science

Computer engineering
CE 2016: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering

Information systems
IS 2010 Curriculum Update: The Curriculum Guidelines for Undergraduate Degree Programs in Information Systems

Information technology
IT 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology.

Software engineering
SE2014 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering

Computing Curricula 2005: The Overview Report
CC 2005 provides undergraduate curriculum guidelines for five defined disciplinary areas of computing: computer science, computer engineering, information systems, information technology and software engineering.

CyberSecurity
CSE C2017
Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity

Appendix 2: Membership of the review group for the Subject Benchmark Statement for Computing

Membership of the review group for the Subject Benchmark Statement for Computing (2019)

The fourth edition, published in 2019, was revised by QAA to align the content with the revised UK Quality Code for Higher Education, published in 2018. Proposed revisions were checked and verified by one of the Co-Chairs of the Subject Benchmark Statement for Computing from 2016.

Dr Alan Hayes	University of Bath
Dr Alison Felce	QAA

Membership of the review group for the Subject Benchmark Statement for Computing (2015)

Dr Phil Brooke	Teesside University
Professor Christopher Clare	BCS Academic Accreditation Committee
Dr Tom Crick	BCS Academy
Professor Sally Fincher (Co-Chair)	University of Kent
Dr Alan Hayes (Co-Chair)	University of Bath
Dr Iain Phillips	Loughborough University
Dr Alan Tully	University of Newcastle
Corresponding members	
Professor Quintin Cutts	University of Glasgow
Professor Andrew McGettrick	University of Strathclyde
Employer representative	
Robert Koger	Vision Semantics and IET Academic Accreditation Committee
Student reader	
Emilia Todorova	Glasgow Caledonian University
QAA Officers	
Janet Bohrer	QAA
Dr Tim Burton	QAA

Membership of the review group for the Subject Benchmark Statement for Computing (2007)

Dr Laurence Brooks	Brunel University
Graham Gough	University of Manchester
Alastair Irons	University of Northumbria
Dr Gerry McAllister	University of Ulster
Professor Andrew McGettrick (Chair)	University of Strathclyde
Professor Keith Mander	University of Kent

Membership of the original review group for the Subject Benchmark Statement for Computing (2000)

Professor J Arnott	University of Dundee
Professor D Budgen	University of Keele
Dr PC Capon	University of Manchester
Mr G Davies	The Open University
Professor PJ Hodson	University of Glamorgan
Professor E Hull	University of Ulster
Professor G Lovegrove	Staffordshire University
Professor KC Mander	University of Kent at Canterbury
Professor A McGettrick (Chair)	University of Strathclyde
Mr P McGrath	Leeds Metropolitan University
Dr A Norman	University of Cambridge
Mr SJ Oldfield	University of Plymouth
Ms A Rapley	Kingston University
Professor VJ Rayward-Smith	University of East Anglia

Fourth edition – October 2019
QAA2453

Southgate House, Southgate Street, Gloucester GL1 1UB
Registered charity numbers 1062746 and SC037786

Tel: 01452 557 000
Web: www.qaa.ac.uk