Computing

2007

ISBN 978 1 84482 678 0

All QAA's publications are available on our website www.qaa.ac.uk

Printed copies of current publications are available from:
Linney Direct
Adamsway
Mansfield
NG18 4FN

Tel 01623 450788
Fax 01623 450481
Email qaa@linneydirect.com

Registered charity number 1062746

Contents

Preface

Subject benchmark statements provide a means for the academic community to describe the nature and characteristics of programmes in a specific subject or subject area. They also represent general expectations about standards for the award of qualifications at a given level in terms of the attributes and capabilities that those possessing qualifications should have demonstrated.

This subject benchmark statement, together with others published concurrently, refers to the **bachelor's degree with honours**[1]. In addition, some subject benchmark statements provide guidance on integrated master's awards.

Subject benchmark statements are used for a variety of purposes. Primarily, they are an important external source of reference for higher education institutions (HEIs) when new programmes are being designed and developed in a subject area. They provide general guidance for articulating the learning outcomes associated with the programme but are not a specification of a detailed curriculum in the subject.

Subject benchmark statements also provide support to HEIs in pursuit of internal quality assurance. They enable the learning outcomes specified for a particular programme to be reviewed and evaluated against agreed general expectations about standards. Subject benchmark statements allow for flexibility and innovation in programme design and can stimulate academic discussion and debate upon the content of new and existing programmes within an agreed overall framework. Their use in supporting programme design, delivery and review within HEIs is supportive of moves towards an emphasis on institutional responsibility for standards and quality.

Subject benchmark statements may also be of interest to prospective students and employers, seeking information about the nature and standards of awards in a given subject or subject area.

The relationship between the standards set out in this document and those produced by professional, statutory or regulatory bodies for individual disciplines will be a matter for individual HEIs to consider in detail.

This subject benchmark statement represents a revised version of the original published in 2000. The review process was overseen by the Quality Assurance Agency for Higher Education (QAA) as part of a periodic review of all subject benchmark statements published in this year. The review and subsequent revision of the subject benchmark statement was undertaken by a group of subject specialists drawn from and acting on behalf of the subject community. The revised subject benchmark statement went through a full consultation with the wider academic community and stakeholder groups.

QAA publishes and distributes this subject benchmark statement and other subject benchmark statements developed by similar subject-specific groups.

---

[1] This is equivalent to the honours degree in the Scottish Credit and Qualifications Framework (level 10) and in the Credit and Qualifications Framework for Wales (level 6).

The Disability Equality Duty (DED) came into force on 4 December 2006[2]. The DED requires public authorities, including HEIs, to act proactively on disability equality issues. The Duty complements the individual rights focus of the *Disability Discrimination Act* (DDA) and is aimed at improving public services and outcomes for disabled people as a whole. Responsibility for making sure that such duty is met lies with HEIs.

The Disability Rights Commission (DRC) has published guidance[3] to help HEIs prepare for the implementation of the Duty and provided illustrative examples on how to take the duty forward. HEIs are encouraged to read this guidance when considering their approach to engaging with components of the Academic Infrastructure[4], of which subject benchmark statements are a part.

Additional information that may assist HEIs when engaging with subject benchmark statements can be found in the DRC revised *Code of Practice: Post-16 Education*[5], and also through the Equality Challenge Unit[6] which is established to promote equality and diversity in higher education.

---

[2] In England, Scotland and Wales

[3] Copies of the guidance *Further and higher education institutions and the Disability Equality Duty*, guidance for principals, vice-chancellors, governing boards and senior managers working in further education colleges and HEIs in England, Scotland and Wales, may be obtained from the DRC at www.drc-gb.org/library/publications/disabilty_equality_duty/further_and_higher_education.aspx

[4] An explanation of the Academic Infrastructure, and the roles of subject benchmark statements within it, is available at www.qaa.ac.uk/academicinfrastructure

[5] Copies of the DRC revised *Code of Practice: Post-16 Education* may be obtained from the DRC at www.drc-gb.org/employers_and_service_provider/education/higher_education.aspx

[6] Equality Challenge Unit, www.ecu.ac.uk

Foreword

The original subject benchmark statement for computing was published in 2000. The present version has been produced in line with an original commitment by the Quality Assurance Agency for Higher Education (QAA) to coordinate a review of existing subject benchmark statements every five years. The original statement has proved to be robust and the changes in this present version are largely a result of some reorganisation and an updating of the original material; the latter tends to be focused on the body of knowledge and that is to be expected in a fast changing discipline like computing. However, the concept of computational thinking has been introduced for the first time and there is now an explicit section on excellence.

The remit given to the original benchmarking group from QAA was to produce a set of benchmark standards that would capture the intellectual and practical attributes that ought to be developed by study of particular disciplines to honours degree level. In formulating these benchmark standards for the discipline of computing, the original benchmarking group took account of a range of related activity emanating from the United Kingdom (UK) and abroad. The original benchmarking group felt that these standards should reflect computing as practised within the UK.

In reviewing the original statement, it seemed appropriate to update the information on the level of penetration of computing degrees. In 2000, some 2,400 computing programmes were on offer. In April 2006, UK universities and colleges offered some 3,344 programmes in computing (with a further 394 for software, 1,434 based on information and 860 on multimedia). This collection of programmes encompasses widely differing titles and differing content. Recognising this diversity, and not wishing to compromise it, the group sought to produce a set of benchmarking standards which would accommodate a wide variety of programmes of study and this would include modular and non-modular courses, both vocational and academic courses, and courses focussing on different aspects of computing. It also wished to encourage both innovation and creativity in programme development, and ensure up-to-date provision. On the other hand, these standards should not date rapidly or be technology dependent, they should not create confusion or uncertainty in the computing academic community and they must not favour any particular approach.

This subject benchmark statement is written to meet the needs of a range of different audiences:

- the academic community who have a particular interest since they must own these standards and engage in relevant programme design and delivery
- academic reviewers and professional bodies who will carry out evaluation of degree programmes
- external examiners whose duties will be informed by aspects of these standards
- and to some extent the public; this is seen to include potential students and their parents or guardians, employers, and careers advisers.

This was done by addressing major topics, namely what is involved in a study of computing; the scope of the discipline; abilities and skills; curricular issues; programme design; learning, teaching and assessment; and finally the benchmark standards themselves.

The title of 'computing' has been chosen as the most representative one for this discipline. This was the overwhelming view of the academic community as expressed at a consultative workshop held during the original benchmarking process.

January 2007

1 Introduction

1.1 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
- combines the ethos of the scholar with that of the professional; it underpins the development of both small-scale and large-scale systems that support organisational goals
- helps individuals in their everyday lives
- is ubiquitous and diversely applied to a range of applications, and yet important components are invisible to the naked eye.

1.2 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 use a creative and dynamic technology. Whatever the perspective, computing can claim characteristics that, while present in other disciplines, are rarely present in such quantities and combinations. Besides being ubiquitous and diversely applied, computing promotes innovation and creativity assisted by rapid technological change. It requires a disciplined approach to problem-solving that brings with it an expectation of high quality; and it approaches design and development through selection from a wide range of alternative possibilities justified by carefully crafted arguments based on insight. 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.

1.3 It is hardly surprising that the diversity of computing is reflected in the varied titles and curricula that higher education institutions (HEIs) have given to their computing-related degree courses. While this subject benchmark statement aims to capture the nature of computing as a discipline, individual HEIs 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 programmes.

2 Nature and scope of computing

2.1 Computing is a discipline that is evolving at a very rapid rate. It is a highly diverse subject and study of different aspects is appropriate to a wide range of student interests and aspirations. A traditional description of computing presents a spectrum of activity, ranging from theory at one end to practice at the other. It also embraces aspects ranging from hardware through to software and from the study of computers and computation per se through to applications-oriented studies.

2.2 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); business (information services)
- philosophy and psychology (human computer interaction and aspects of artificial intelligence)
- physiology (neural networks)
- linguistics.

2.3 This subject benchmark statement is not intended to constrain the development of new courses, rather such initiatives, as well as innovative approaches to the design and development of new degrees, are to be encouraged within the basic principles of the framework that this statement describes. Again, no fundamental difficulty in accommodating joint degrees is perceived within the computing component of such degrees. However, it is expected that the principles set out in this statement will apply. It is not the intention of this document to define curricula or syllabi. A body of knowledge that is broadly indicative of areas that might be considered to come under the remit of the term computing is given at appendix B. These can all contribute to the study and development of computer systems.

2.4 Typically, degree programmes in computing include study of the nature of computation, with effective ways to exploit computation, and with the practical limitations of computation in application terms. The concept of computational thinking captures these concerns. Computational thinking is a basic analytical ability that has relevance (which every computing graduate should recognise) in many aspects of everyday life.

2.5 Its main characteristics include:

- 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 modelling
- recognition of the relationships between the concepts of specification, program and data (in all its forms), as well as the power of the notion 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, safe and critical systems - often in the presence of uncertainty - and approaches to achieving this
- recognising simplicity and elegance as useful concepts and principles on the one hand, but also bad and dangerous practices on the other.

2.6 In degree programmes in computing, there will often be a pervading concern with analysis and design, problem-solving, the nature of information and its processing, and the wide range of levels of abstraction from which computation can be viewed. Courses need to be designed to possess themes that ensure students are equipped to contribute to the development of major components of computer systems in a manner that ensures they are fit for the purpose for which they were originally intended. The latter implies an understanding of the mechanisms that will ensure quality in both process and product, and this will often mean a comprehension of how systems should be designed for use by humans. Such themes will exhibit an appropriate balance of theory and practice, and will include the methodologies that ensure students will adopt a disciplined approach to their tasks. This will include:

- problem definition, specification, design, implementation and maintenance
- the necessary knowledge, including an understanding of the range of possible options for these tasks
- the data structures and algorithms
- the related practical and transferable skills, including relevant approaches to group activity
- access to the appropriate resources, including tools
- the necessary underpinning to guide practice to ensure the sustainability of their knowledge and to provide an appropriate framework that will accommodate rapid technological change. The necessary underpinning may come also from diverse disciplines including mathematics, empirical or experimental insight, engineering, psychology, aesthetics, organisational theory, management or graphical design.

2.7 HEIs are likely to offer more than one degree programme in computing. A degree programme, or a programme component in the case of a joint degree, will count as lying within the area of computing if the existence of computers and associated technology is seen as a central driving force in its motivation. 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. HEIs will produce aims and objectives that characterise their programmes and indicate that their curricula are at honours degree level. Honours degree programmes in computing can take various forms, each of which could prepare their students for different but valid careers. Certain areas within the field such as artificial intelligence, computer science, information systems, software engineering, multimedia, communications, networks and computer games, form familiar domains of activity that are represented strongly within computing.

2.8 At one extreme, an honours degree programme in computing might provide opportunities for its students to attend modules on a wide range of topics spanning the entire area of computing. Graduates from such courses would have great flexibility and might be of especial value, either in emerging areas where specialist courses may not be established or in contexts where their ability to span the field would be useful. At another extreme there can be programmes that take one very specific aspect of computing and cover it in great depth. The graduates from such programmes will typically tend to seek opportunities in the specialist area they studied, whether it be the development of multimedia systems, network design, the formal verification for safety-critical systems, electronic commerce or whatever other specialities emerge and become important. The overall field is wide-ranging and it is important that those working in unusual and innovative areas recognise that they also reside within the field of computing.

2.9 Programme designers, students and employers will need to be aware of this spectrum of programme identities, but behind such variation there are three key ideas which constitute a certain ethos that can be expected to characterise any honours degree programme in computing:

i the concept of computational thinking, the recognition of its main elements and the relevance of these to everyday life

ii the computing system (and this includes systems such as information systems), 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

iii there is a balance of practice and theory, appropriate to the aims of the particular degree programme, such that practical activity can be supported by an understanding of underlying principles.

3 Abilities and skills

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

i computing-related cognitive abilities and skills, ie abilities and skills relating to intellectual tasks

ii computing-related practical skills

iii additional transferable skills that may be developed in the context of computing but which are of a general nature and applicable in many other contexts.

3.2 Cognitive, practical and generic skills need to be placed in the context of the programme of study as designed by the institution and/or the possible pathways selected by the individual student. The implicit interplay between these identified skills both within and across these three categories is recognised.

Subject-related cognitive abilities

- Computational thinking including its relevance to everyday life.
- Knowledge and understanding: demonstrate knowledge and understanding of essential facts, concepts, principles and theories relating to computing and computer applications as appropriate to the programme of study.
- 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.
- Requirements, practical constraints and computer-based systems (and this includes computer systems, information systems, embedded systems and distributed systems) in their context: recognise and analyse criteria and specifications appropriate to specific problems, and plan strategies for their solution.
- Critical evaluation and testing: analyse the extent to which a computer-based system meets the criteria defined for its current use and future development.
- Methods and tools: deploy appropriate theory, practices and tools for the specification, design, implementation and evaluation of computer-based systems.
- Reflection and communication: present succinctly to a range of audiences (orally, electronically or in writing) rational and reasoned arguments that address a given information handling problem or opportunity. This should include assessment of the impact of new technologies.
- 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.

Subject-related practical abilities

- The ability to specify, design and construct computer-based systems.
- The ability to evaluate systems in terms of general quality attributes and possible trade-offs presented within the given problem.
- The ability to recognise any risks or safety aspects that may be involved in the operation of computing equipment within a given context.
- 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.
- The ability to operate computing equipment effectively, taking into account its logical and physical properties.

3.3 The extent to which students acquire these abilities will depend on the emphasis of individual degree programmes. It is expected, however, that the student will be able to deploy these abilities to a greater and deeper extent than someone who is merely an interested practitioner.

Additional transferable skills

- Effective information-retrieval skills (including the use of browsers, search engines and catalogues).
- Numeracy and literacy in both understanding and presenting cases involving a quantitative and qualitative dimension.
- Effective use of general information technology (IT) facilities.
- The ability to work as a member of a development team, recognising the different roles within a team and different ways of organising teams.
- Managing one's own learning and development including time management and organisational skills.
- Appreciating the need for continuing professional development in recognition of the need for lifelong learning.

4 Principles of course design

4.1 An honours degree programme in computing can legitimately address a wide range of topic areas and associated practical skills and applications from the broad field. Although many degree titles are possible, any such title should reflect the content of the degree programme. Because of the permissible variability of courses, this subject benchmark statement does not prescribe any core of material guaranteed to be present in all courses. However, each HEI should be expected to be able to show that:

- the course is designed as a coherent whole with theory, practical skills and applications integrated in a harmonious manner. It should be up-to-date in terms of developments in computing and current thinking on curriculum development and delivery and it should take appropriate account of issues such as the employability of its graduates and the needs of employers
- it has clear and achievable aims, objectives and intended learning outcomes which match its title and the programme specification
- courses are imaginatively designed to meet as effectively as possible the needs of the full range of intended students in terms of course length/duration, modes of attendance including part-time possibilities, location, structure and sequence, and optional elements
- on each pathway, every student will have exposure to those key topics and practices most relevant to its central objectives and title and the design of this should be informed by considerations articulated below
- the course shows progression with later parts complementing, extending or building upon earlier ones
- the programme presents coherent underpinning theory appropriate to the aims of the course, and this is further developed and used throughout the course. This should be such as to enable graduating students to adapt to future developments in the field. Overall, the course should reflect the rapid rate of change in the field and ensure that coverage is given to a selection of emerging topics so that students are aware of likely future developments in the subject together with their potential impact

- courses need to be designed to possess themes that ensure students are equipped to contribute to the development of major components of computer systems in a manner that ensures they are fit for the purpose for which they were intended. The latter implies an understanding of the mechanisms that will ensure quality in both process and product and this will often mean a comprehension of how systems should be designed for use by humans
- in those parts of the curriculum that relate to an engineering approach to the subject, the concepts of requirement, specification, design, implementation, evolution and maintenance are pervasive and an appropriate engineering ethos is present
- in those parts of the curriculum that have a mathematical, scientific, psychological, aesthetic, systems, management or organisational orientation, there is appropriate underpinning which ensures that students acquire well-founded insight into the range of possible approaches
- in practical coursework there is an opportunity for students to gain experience of working both in groups and as an individual
- in relevant parts of the course students are encouraged to reflect, evaluate, select, justify, communicate and be innovative in their problem-solving
- there is provision for the development of a range of personal and transferable skills generic to all graduates
- there is 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 but individual HEIs are free to use alternative arrangements where that would best fit their particular course structure or content
- where appropriate, in terms of meeting the overall objectives of a course, such activity 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 and building on the experience afterwards, and assessment
- the assessment strategy associated with the course is clearly documented and will allow the HEI to show that graduating students meet the criteria set in this subject benchmark statement.

4.2 The different themes will also need to include attention to evaluation methods and an assessment of the experience within the context of their course coupled with a clear understanding of what constitutes a high-quality system in this context and how such systems can be developed.

5 Teaching, learning and assessment

Teaching and learning

5.1 Rapid changes in computing are having a considerable impact on the educational environment in HEIs. Assessing the significance of new technical developments, mastering them and their related tools and integrating them into the curriculum is often challenging and time-consuming. HEIs need to assure themselves that there are opportunities for both academic and support staff to keep up to date in terms of their discipline and pedagogy.

5.2 In a HEI offering honours degrees in computing, it would be expected that:

- the teaching staff are appropriate for the curriculum on offer and they are effectively deployed
- there are mechanisms for both academic and support staff being kept up to date in their discipline and in relevant pedagogy
- computing is presented as a discipline whose character is outlined in other sections of this subject benchmark statement
- approaches to teaching stimulate learning and the development of skills in lifelong learning, with students being encouraged to take responsibility for their own learning
- teaching and learning are based on explicit objectives, which are consistent with course aims and objectives as set out in the programme specification, and are well-planned with teaching methods being effective and varied and making efficient and effective use of available facilities and equipment
- learning is enriched by appropriate underpinning, current research, industrial applications and the development of transferable skills.

Assessment

5.3 Students should be provided with opportunities to engage in debate about many aspects of their studies, including assessment, for which they should ultimately take ownership. Students should receive appropriate and timely feedback on all work, and this includes constructive feedback on coursework and project work. In a sense, students should experience a range of approaches to assessment. Over-assessment should be avoided, and in this regard new approaches to assessment are encouraged.

5.4 In general terms, all major activities on an honours degree programme in computing should receive an assessment with credit contributing to student progress or an award. On the grounds that assessment guides learning, all important skills ought to be formally assessed in a manner that ensures students acquire these skills. Any assessment must ensure fairness and reliability. In this respect, the assessment of sandwich placements offers challenges and there is scope here for innovative forms of assessment. There are important issues of comparability and quality control across the wide range of potential environments in which placement occurs, and the roles which students take within these environments.

5.5 In teaching computing to honours degree level, it is expected that:

- an assessment strategy appropriate to the required balance of knowledge and skills exists to ensure that students achieve the requisite standards on all distinctive aspects of a course. This should include addressing the range and scope of assessment methods (and this might include, for example, the problem-solving nature of formal examination questions), the scheduling and weighting of assessments, quality control issues, with theory and practice being assessed in some combination
- all student assessments are set at an appropriate standard and the assessment of this work taking full account of the circumstances under which it was done and the available resources is fair, valid and reliable and ensures that appropriate

standards are achieved. Passes obtained at each level reflect an appropriate level of student achievement and students should be encouraged to develop a facility in self-assessment

- a range of other external and internal measures are used to safeguard standards eg accreditation activities involving professional bodies, internal/external reviews and students have ready access to reasonable appeal procedures.

Learning, environments and resources

5.6 In general terms, HEIs must assure themselves that the resources, including human resources, are appropriate for their courses.

5.7 The computer systems that students use are an important factor in the teaching and learning environment. Increasingly, they are used for the purposes of learning. Material such as notes, assessments, case studies, access to digital libraries and other website information, videos, software, standards, dedicated laboratories and special purpose laboratories (eg for configuring networks) can all contribute to an environment that supports appropriate learning. High-quality software, tools and materials generally condition the expectation of students and their approach to practice and this is an essential dimension of the educational process. Apart from the usual range of languages, basic software including software libraries, graphics packages, computer-aided software engineering (CASE) tools, environments, network analysers, multimedia development tools, theorem provers, project management software, and planning systems, as appropriate to the programme of study, might be provided. Departments often support students by making available software and communications facilities that enable them to extend their horizons. Thus, up-to-date software tools as well as modern facilities such as digital libraries and simulations all have a role to play.

5.8 Such systems must be of high quality and this includes being free of hostile software, being properly managed and supported in a manner that reflects best practice in the discipline, and there being in place a regime that ensures excellence in terms of computer use.

5.9 Beyond this, learning in the future is likely to revolve around the use and exploitation of electronic material, as well as electronic discussion and communication. Accordingly, there is a challenge to teach the basic principles of best practice in terms of these approaches and this will include electronically mediated group work, electronically supported collaborative learning, learning from electronic resources, and so on. For the present, academics face the challenge of how to teach this and how to assess it.

5.10 In summary, the following would be expected of an environment in which an honours degree in computing is taught:

- the academic environment (and this includes support for teaching as well as the computer and software environments) is conducive to learning, and the nature of case studies, applications and other resource materials is appropriate to the aims of the course and the level of teaching
- there are sufficient resources to deliver the curriculum, including learning support materials, equipment and IT, and these are up-to-date, readily available with easy access, well-managed and effectively deployed
- provision of high-quality teaching staff with appropriate specialist interests and support staff.

Enhancing the student experience

Student motivation

5.11 HEIs should assure themselves that their students are appropriately motivated. Different students are motivated to learn in different ways, and a range of possibilities exist in the context of computing. Some students are very content with traditional environments while others respond to industrially relevant activity perhaps in an industrial setting. Others relish international competitions, some want to be associated with research activity, others respond to being given responsibility, and so on. It is proper that the diversity exists, not necessarily within the one institution. But in all these different possible settings serious consideration needs to be given to ensure the quality of the learning environment and of the student experience. Opportunities need to exist to challenge, stimulate and motivate students and to involve them in active learning. Associated with this is the assessment process and there should be an appropriate mix of feedback, encouragement and rewards to generate self-confidence in students.

5.12 Within the context of teaching computing to honours degree level, it should be expected that:

- students are appropriately motivated, bearing in mind the emphasis and the character of their course, and this should be reflected in applicants and progression rates
- graduates should act as the agents of technology transfer (when they ultimately move into employment) and to this end receive education in appropriate high-quality environments
- students should gain some feeling through illustrations and case studies of the range of applications appropriate to their discipline and of the career outlets that meet their interests and aspirations.

Student induction

5.13 It is important to recognise that students need to receive induction into higher education and into the practices of their discipline. Students need to be told what is expected of them in a HEI; what is expected of them within the discipline of computing in terms of tasks such as writing reports, writing essays, giving presentations, writing programs; how to work with others yet avoid plagiarism; and what constitutes excellence and how to achieve it. Throughout a course, this needs to be reinforced eg through regular feedback on coursework or more generally formative assessment. Amounts of feedback and support should reflect student needs. In all cases, students need to be encouraged to accept, as they go through their course, increasing amounts of responsibility for their own learning and the assessment related to this.

5.14 The expectation on all HEIs offering honours degrees in computing is that:

- students receive proper induction into their course and this is reinforced throughout the course in a manner that reflects appropriate progressive development
- students receive appropriate and timely feedback on all student work, and this includes constructive feedback on coursework and project work.

6 Benchmark standards

6.1 Benchmark standards are defined at threshold and typical levels and some further comments are made about excellence.

Threshold level

6.2 The threshold level is interpreted to mean that all students (taken over all years), graduating with an honours degree in the discipline of computing will have achieved this level of performance.

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

- demonstrate a requisite understanding of the main body of knowledge for their programme of study
- understand and apply essential concepts, principles and practice of the subject in the context of well-defined scenarios, showing judgement in the selection and application of tools and techniques
- produce work involving problem identification, the analysis, the design or the development of a system with appropriate documentation, recognising the important relationships between these. The work will show some problem-solving and evaluation skills drawing on some supporting evidence, and demonstrate a requisite understanding of the need for quality
- demonstrate transferable skills and an ability to work under guidance and as a team member
- identify appropriate practices within a professional, legal and ethical framework and understand the need for continuing professional development
- discuss applications based upon the body of knowledge.

Typical level

6.4 This is the average standard attained (taken over all years) of all the students graduating with an honours degree in the discpline of computing.

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

- demonstrate a sound understanding of the main areas of the body of knowledge within their programme of study, with an ability to exercise critical judgement across a range of issues
- critically analyse and apply a range of concepts, principles and practice of the subject in an appropriate manner in the context of loosely defined scenarios, showing effective judgement in the selection and use of tools and techniques
- produce work involving problem identification, the analysis, the design or the development of a system, with accompanying documentation, recognising the important relationships between these. The work will show problem-solving and evaluation skills, draw upon supporting evidence and demonstrate a good understanding of the need for quality

- demonstrate transferable skills with an ability to show organised work as an individual and as a team member and with minimum guidance
- apply appropriate practices within a professional, legal and ethical framework and identify mechanisms for continuing professional development and lifelong learning
- explain a wide range of applications based upon the body of knowledge.

Excellence

6.6 While the benchmark standards above are defined for just threshold and typical levels, it is nevertheless expected that programmes in computing will provide opportunities for students of the highest calibre to achieve their full potential.

6.7 Such students will be:

- creative and innovative in their application of the principles covered in the curriculum, and may relish the opportunity to engage in entrepreneurial activity
- able to contribute significantly to the analysis, design or the development of systems which are complex, and fit for purpose, recognising the important relationships between these
- able to exercise critical evaluation and review of both their own work and the work of others.

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

Appendix A - Glossary of terms

The following terminology is used in this report.

Module: a unit of learning and teaching which is assessed as a single unit and for which credit or a pass is given; for example, a set of 20 lectures and associated tutorials and practical activity.

Pathway: a set of modules taken by a student and normally leading to the award of a degree; note that this may involve the selection of optional modules or electives.

Course: a collection of one or more different pathways leading to a single award, for example, of a degree.

Programme of study: normally synonymous with course.

Cognate area: a collection of programmes of study leading to an award within a single discipline, in this case computing.

Field of study: normally synonymous with cognate area.

Level: a hierarchical categorisation of material within a discipline, such as computing; in full-time courses this is likely to be closely related to the year of study.

Benchmark standards: measures of achievement on conclusion of a programme of study, for which this report defines threshold and typical (modal) in the context of an honours degree course.

Appendix B - Body of knowledge

The following list of topics is seen as defining the scope of the broad area of computing. It is not intended to define curricula or syllabi, it is merely provided as a set of knowledge areas indicative of the technical areas within computing.

Architecture

The computer processing unit/memory/input-output (IO) model, representation of data and programs in memory, arithmetic/logic unit, fetch-execute cycle, registers, stacks, data-paths, special IO-support hardware, support for protection and virtual memory, instruction sets, implementation constraints and trade-offs, historical, current and future trends. Peripheral devices, interfacing. Cache memory and memory hierarchies. Measuring performance. High performance and parallel architectures; pipeline processors, array processors and single instruction multiple data (SIMD) architectures, shared-memory multiple instruction multiple data (MIMD) machines (symmetrical multi-processor and non-uniform memory access), message-passing MIMD machines.

Artificial intelligence

This is a discipline with two strands. The scientific strand attempts to understand the requirements for and mechanisms enabling intelligence of various kinds in humans, other animals and information processing machines and robots. The engineering strand attempts to apply such knowledge in designing useful new kinds of machines and helping us to deal more effectively with natural intelligence, eg in education and therapy. Knowledge elicitation and representation. Constraint-based programming. Uncertainty. Cognitive modelling. Logics. Reasoning. Deduction and theorem proving. Search. Machine learning. Agent technology. Planning. Vision systems, robotics. Speech and language technology.

Comparative programming languages

The variety of languages and the motivation for this variety, including languages such as scripting languages. Design criteria for languages. Desirable properties of languages and their implementations. Different programming paradigms: imperative, object-oriented, functional, logic, visual. Concurrency, parallelism and distributed computing. Strengths, weaknesses of different language features including types and data modelling, control structures, structuring concepts, abstraction mechanisms, parameterisation, exception handling, separate compilation, generics. Declarations, naming conventions, storage allocation strategies; parameter passing mechanisms.

Compilers and syntax directed tools

Aims in compiler construction. Variation in possible users. Features of languages. Phases of development. Lexical analysis. Parsing: LL grammars, LR parsing, parse trees and abstract syntax trees. Symbol tables. Type checking. Semantic analysis. Run-time storage organisation. Code generation. Code optimisation. Illustration of other syntax directed tools.

Computational science

Combinatorics, integer and linear programming, optimisation, operational research, e-science, computer algebra.

Computer-based systems

Definition of computer-based systems. Different kinds of systems: to include embedded systems, real time systems, distributed systems, client-server systems. Safety critical and other high-integrity systems: risk analysis and assessment. Systems approach. Modelling. Needs, goals and objectives; requirements definitions; functional analysis and derivation of non-functional requirements; specification development; evaluation of trade-offs and alternatives leading to formulation of system architecture; allocation of responsibilities leading to sub-system design and interface definitions. Co-design issues. Problem of integration, configuration management, quality assurance, operations and maintenance. Performance measures. Forensics.

Computer communications

Digital communication: standards, media, signalling, reliability, error handling and performance. Device management, input/output considerations. Communications management. Communications software.

Computer hardware engineering

Specification, design (using electronic computer-aided design and hardware description languages), simulation, verification, construction and testing of the hardware of computer systems using appropriate technologies for logic, memory, storage and communication (with users and other machines). Understanding future technology trends and the requirements placed by software systems on computer hardware.

Computer networks

Networks: topologies, protocols and standards. Different communication media and data, and related requirements. Reference models, switching, access, security, compression, encryption, mobile operation, quality of service, performance, management, interconnection and architectural models, routing, congestion, firewalls, proxy servers. Network operating system design. Future trends: emerging technologies and applications.

Computer vision and image processing

The design of computer algorithms and hardware to model the structure and properties of visual data. Modelling techniques and algorithms: human vision system based, engineering perspective based. The extraction and application of information from these models. Image processing: pattern recognition, the manipulation of the image signal to include image analysis: the extraction of semantic data, animation, manipulating images.

Concurrency and parallelism

Nature of concurrency, problems. Examples, including IO. Concurrent processes, interprocess communication. Low level synchronisation primitives. Language primitives for shared memory. Concurrency at operating system, language level. Atomic actions. Scheduling and real-time systems. Performance issues. Resource allocation and deadlock. Concurrency control and recovery. Classification of parallel machines. Algorithms and algorithm design in the context of parallelism. Complexity and performance metrics associated with algorithms in the context of concurrent systems.

Databases

The concept of a database and database management. Database development. Illustrations. Entity-relationship model. Database design: logical design and the relational model, physical design. Normalisation; different normal forms. Client-server model. Structured query language (SQL) and database servers. Database access and client applications. eXtensible mark-up language (XML). Object oriented systems, multimedia database systems, distributed database systems. Spatial databases and geographic positioning systems. Database administration. Data mining, data warehousing. Databases and websites. Tools.

Data structures and algorithms

Data types, structures and abstract data types. Efficiency measures (average and worst case), rates of growth, asymptotic behaviour. Algorithmic paradigms (including enumeration, divide-and-conquer, greedy, dynamic programming, tree search, probabilistic). Algorithm design and analysis with correctness proofs. Data processing algorithms (sorting, searching, hash tables etc). Data mining. Numerical algorithms and analysis; statistical algorithms and simulation. Graph theory and graph theoretic algorithms (shortest paths, spanning trees, etc). Symbolic computation. Other application areas, eg sequencing, scheduling and assignment. Parallel and distributed algorithms, implementation issues and efficiency measures.

Developing technologies

For example, quantum computing, bio-informatics, medical computing, ubiquitous computing, computer forensics, e-science, autonomic computing, grid computing, high performance computing.

Distributed computer systems

Characteristics of distributed systems, client-server model, interprocess communication, remote procedure calls, distributed operating systems, naming and protection, file service design, shared data and transactions, concurrency and control, time coordination and time-stamping, replication, fault handling and recovery, distributed system security. Computer supported collaborative work.

Document processing

Word processing systems, design and development. Related tools: editors, spelling checkers, grammar checkers. Mixed systems including tables, diagrams, pictures. Presentation systems. Web-based documents: the benefits and drawbacks. Design and development of web-based documents. e-Publishing, digital typography. Document engineering. Mark-up languages. Multimedia presentation. Contents and index generation. Copyright and other legal issues.

e-Business

Nature of e-business: the particular cases of e-commerce, e-learning. Impact on practices and on organisations. Business process re-engineering. Levels of automation. Distributed transactions, security and privacy. Particular problems. Major components in such a system. Hard and soft e-commerce. Business-to-business and business-to-customer technologies. Digital signatures and authentication issues. Legal and ethical issues. Need for deep knowledge in certain application areas.

Empirical approaches

Experiments: their role and purpose. Experimental design, hypothesis formulation and hypothesis testing. Empirical methods.

Games computing

Computer games programming, games design, interaction, relevant aspects of advanced computer graphics, relevant mathematics for computer games, artificial intelligence for computer games, online gaming and networking, console architectures, professional issues for games. More generally, computers in entertainment.

Graphics and sound

Human perception of images, display and image-capture technology, storage formats and algorithms for the manipulation of two-dimensional (2D) and three-dimensional (3D) representation, transformations on images, geometric modelling, animation, rendering with realistic lighting and texture effects. Human perception of sound, frequency versus time domain representations, sound compression, synthesis, sound analysis. 2D and 3D modelling, animation, virtual reality, multimedia. Scientific and information visualisation. Computational geometry. Object modelling. Animation.

Human-computer interaction (HCI)

User interface engineering: user-centred design and evaluation methodologies, architectures, IO modes (including multi-modal) and devices, development environments, interface managers, construction skills; HCI guidelines, principles and standards; interaction styles, metaphors and conceptual models. User models: human psychology and actions, ergonomics, human information processing. Human-computer applications: including virtual and connected environments (including mobile), games, visualisation, multimedia, affective computing, systems for users with special needs. Usability engineering and evaluation.

Information retrieval

Information and its management. Transmission issues. Text, graphics, speech, sound, documents and other kinds of content. Methods of retrieval for different content. Methods based on logic, and probability theory, situation theory, computational logistics. Web-based considerations. Interactive considerations including feedback issues. Case-based reasoning, hypertext, visualisation.

Information systems

Theoretical underpinnings. Data, information and knowledge management. Information in organisational decision making. Integration of information systems with organisational strategy and development. Information systems design. Systems approaches. Compression technologies. Development, implementation and maintenance of information systems. Information and communications technologies (ICT). Decision support. Management of information systems and services. Content management systems. Organisational and social effects of ICT-based information systems. Economic benefits of ICT-based information systems. Personal information systems. Digital libraries.

Intelligent information systems technologies

Theory, design and development of database systems, database applications, data warehouses, data mining principles, decision support system development including intelligence density (quality, models, constraints, organisational factors), decision trees, genetic algorithms, neural networks, fuzzy logic, case based reasoning, information presentation.

Management issues

Software project management, management and economic aspects of outsourcing and off-shoring. Aspects of managing IT systems, eg documents, resources, networks, websites. The management of risk.

Middleware

Examples of objects, and object libraries. Characteristics of well-designed and high-quality objects. Design guidelines. Methods of ensuring quality. Building new classes in accord with the guidelines. Design patterns. Design languages. Tool support. Mechanisms for interconnection of classes and modules. Integration as a concept and as a vehicle for system enhancement. Mechanisms for achieving integration. Interconnection languages: scripting languages. Building systems in this environment, including distributed systems. Verification and validation of such systems. Applications.

Multimedia

Multimedia seen as the capabilities of modern computer technology to employ multiple-media communication forms (including data, text, graphics, still and video images and sound) integrated into single applications. Distinguished from other forms of multiple-media by the fact that the computer reduces all information into a digital form that can be reproduced, manipulated, stored and transmitted electronically. Consideration of the representation, storage and transmission issues for different digital forms, and the subsequent transformations of these forms. Operations. Design and development issues. User interface and presentational matters. Tools support.

Natural language computing

Advanced computing techniques to enhance the capabilities of systems providing text and speech. Communication. Language generation, language models, parsing and understanding, machine translation. Advanced models of interpersonal and human-computer dialogue; advanced methods for language processing by providing robust, accurate and efficient treatment of language in a range of applications and of user-situations. Speech recognition and synthesis. Text analysis.

Operating systems

Role, functions of operating systems. Characteristics, capabilities of single user systems, multi-user systems. Illustrations. Process concept. Architecture of an operating system: influences of networks, multimedia, security. File processing systems. Resource management. Basic services - memory management, interrupt handling, process scheduling. Concurrency mechanisms. Scheduling. System processes - spooler, network interface; utilities. Security and protection issues including access control, virus protection. Shell programming. Relationship to window systems.

Professionalism

Ethics: consideration of the individual, organisational and societal context in which computing systems are planned, developed and used; deployment of technical knowledge and skills with a concern for the public good. Ethical responsibilities. Techniques of ethical analysis. Social aspects of computing. Computer crime. Law: awareness of relevant law and processes of law eg data protection, computer misuse, copyright, intellectual property rights, basic company and contract law. Systems: development and operational costs; safety/mission criticality; consequences and liability issues of failure; risk analysis; security; recovery. Professional bodies: structure, function, restriction of title, licence to practice, codes of ethics/conduct/practice.

Programming fundamentals

The nature of programming. Use of some well-designed and appropriate programming language. The idea of syntax and semantics, and related errors. Programming paradigms. Problem analysis, program design, coding including interface considerations. Simple programs and simple algorithms. Abstraction mechanisms, parameter passing. Simple quality considerations, including strategies for testing and debugging. Use of libraries. Different kinds of documentation serving different purposes.

Security and privacy

Security and privacy: the problems. Illustrations of how problems arise. Physical and logical security. Machine access. Protection mechanisms. Encryption and encryption building blocks. Virtual private networks. Legal issues. Firewalls and internet security. Monitoring of traffic and computer use. Digital signatures. e-commerce, e-banking and related applications.

Simulation and modelling

Uses of modelling and simulation. Benefits and drawbacks. Model classification, systems theory. Continuous and discrete simulation. Applications. Simulation languages. Model building. Model validation. Different approaches and different types of simulation.

Software engineering

Development paradigms; requirements elicitation/specification; analysis and design (including architectural design and design patterns); system models; programming paradigms; prototyping and evolution; testing; verification and validation; assessment and evaluation; software reuse; software measurement and metrics; operation and maintenance; quality assurance and management; configuration management; formal description techniques; software dependability; tools (including CASE) and environments; software process models; implementation; documentation. Agile methods, open-source software development.

Systems analysis and design

Systems theory. Systems within an organisation. Different kinds of systems serving different purposes, eg autonomic, distributed, ubiquitous. Systems in support of an enterprise which is potentially complex and may have to adapt. Typical computer systems lifecycles. Systems requirements and specification. Feasibility concerns. System design: strengths and weaknesses of relevant methodologies and techniques. Fault tolerance. People and interface issues. Compliance with legal and ethical standards. Development, implementation and maintenance. Quality considerations.

Theoretical computing

Models of computation, computability, automata theory, formal language theory, analysis of algorithms, computational complexity, mathematical aspects of programming language definition, logic and semantics of programming languages, foundations of programming, theorem proving, software specification, data types and data structures, coding theory, theory of databases and knowledge based systems, models of concurrency, statistical models of system performance, formal methods of system development. The subject also includes the development of the mathematical techniques used in the list above.

Web-based computing

The specification, design, implementation and operation of web-based technologies and services: currently wired and wireless internet protocol (IP) protocol-based technologies, mark-up languages, HCI, branding and brand loyalty. Web scripting languages/systems. Mobile computing. Enterprise systems: intranets and extranets: access control, security, authentication, encryption, intellectual property rights (IPR), costing, pricing, charging and funding. Tools. Server selection, installation, configuration and administration. Logs and traffic analysis. Searching and search engines. IPR and copyright. Impact of networked economy at regional, national and international levels.

Appendix C - Membership of the review group for the subject benchmark statement for computing

Dr Laurence Brooks	Brunel University
Graham Gough	The 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

Appendix D - Membership of the original benchmarking group for computing

Details below appear as published in the original 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	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

The Quality Assurance Agency for Higher Education
Southgate House
Southgate Street
Gloucester
GL1 1UB

Tel 01452 557000
Fax 01452 557070
Email comms@qaa.ac.uk
Web www.qaa.ac.uk

QAA 170 03/07