PROGRAMME SPECIFICATION

BSC COMPUTER SCIENCE

Awarding Institution	University of London
Teaching Institution	Goldsmiths College
Department	Computing
Final Award	BSc (Hons)
FHEQ Award Level	6
QAA subject benchmarking group	Computing
Date	April 2010
Editors	Marian Ursu and Sebastian Danicic

Background to this Programme Specification

This document is the result of the curriculum review undertaken by the Department of Computing which has resulted in a fully revised curriculum starting 2010-11. It is the specification of the BSc Computer Science programme, which is intended to replace the existing BSc Computer Science programme and forms part of the department's integrated suite of undergraduate programmes.

Introduction to the Programme

The BSc in Computer Science aims to give you a clear understanding of the process of developing software systems and some of the most dominant approaches and technologies currently in use. It teaches ways of thinking with the aid of appropriate technologies, rather than just the technologies themselves. This will allow you to become an independent practitioner or researcher, able to adapt to the new technologies that, in this discipline, are developing at a very fast pace.

Programming is the concept at the core of this programme. Essentially, you will be expected to learn how to program. There are various programming approaches and techniques, supported by different theories and abstract models, and implemented in different technologies. You will be exposed to some of the most important ones and will be given the opportunity to specialise in those of your choice.

You will be provided with a wide range of resources for learning, that will make your learning experience engaging, exciting and, not least, effective.

As, upon graduation, you are expected to have strong background in programming and good skills in at least one programming language, you will have a large spectrum of jobs for which you will be qualified and therefore from which you could choose, in more or less any application area (from the media to the financial sector).

As an alternative route, you could consider to continue your professional development with postgraduate studies in any more specialised field of Computing.

What are the admissions criteria?

You will be expected to have at least BBB or ABC at A2 level, or equivalent.

An A2 level qualification, or equivalent, relating to science, technology and mathematics is preferred. However we encourage applications from those without a formal qualification in these areas who can demonstrate relevant knowledge, skills and experience.

All applicants may be called for an interview, at which time they may be asked to take a computer aptitude test.

If you do not have an A2 level qualification, or equivalent, relating to the sciences, you should have a good pass in GCSE Mathematics, or equivalent.

Applicants whose first language is not English must have received a score of 6.0 or more in the IELTS (or equivalent) examination for written English.

What are the **aims** of the programme?

The overall aim of this programme is to produce graduates who are independent, creative and reflective computing practitioners. In particular, the BSc Computer Science programme aims to:

- to provide a stimulating environment which enables students to develop their full academic potential by encouraging them to be creative, critical and responsive to new ideas

- to provide students with a strong conceptual and theoretical understanding of fundamental methods, theories, techniques and technologies leading to the ability to select, apply and evaluate them in the development of software-based systems

- to develop critical, analytical and interpersonal skills that prepares students to become autonomous professionals in industry or research, able to work independently and in groups.

What are the **learning outcomes** of the programme?

Knowledge and understanding

Graduates should have knowledge of:	Taught by:
Fundamental topics underlying software systems and programming. This knowledge will be sufficient for basic application to small-scale real-world problems.	*Introduction to Programming, Data Representation and Architecture, Mathematical Modelling for Problem Solving, Media Computing, Programming with OO Data Structures and Algorithms, Data Modelling [Databases]*, *Software Engineering and Development*.
Mathematical underpinnings of Computing and the use of mathematical and other forms of **abstraction for modelling** systems. [1]	This is taught in particular by the core modelling stream modules (*Data Representation and Architecture Modelling*, *Mathematical Modelling for Problem Solving*, *Network Modelling*, *Data Modelling*) and *Software Engineering and Development* and *Algorithms and Complexity Theory*
The process and consequent problems in **moving from vague** requirements **to relatively tight** specifications. The knowledge will be sufficient for application to small but complete software projects. [1]	*Software Engineering and Development* plus all the programming modules (all the modules that have "programming" in the title)
The necessity, principles and techniques for **decomposing** large problems to make them comprehensible and computationally solvable. This knowledge should be sufficient for application to small but complete software projects. [1]	All the modelling modules (the core modelling stream) and all the programming modules. The latter includes the core programming stream (*Introduction to Programming*, *Programming with OO Data Structures and Algorithms* and *Programming User Interfaces*) and the CS specialist programming modules *CS1*, *Web Programming* and *Network and Distributed Programming*)
A wide range of **classes** of problems and algorithms for their solution. Many will be learned to a basic level but some will be learned in depth. [1]	*Audio-Visual Computing* and all the level 2 and level 3 modules
Methods for **analysing** and **evaluating** reasonably complex abstract models and concrete implementations	*Algorithms and Complexity Theory*, *Software Engineering and Development,* and, partly, all the programming modules

[1] The level 3 modules will also contribute to this learning objective, but in different proportions. In particular they will contribute to providing new **classes** of problems and algorithms for their solution.

Applicative cognitive/intellectual skills.

Graduates should be able to:	Taught by:
Given a specific real world problem, **decide the algorithmic class** in which it lies, and **select** and **apply** the **specific appropriate instances** of this class in specifying the solution	All the modules at level 2 and level 3
Abstract and **generalise** complex problems into **appropriate models**, through **decomposition**, when necessary, in order to facilitate an implementation	This will be taught across the curriculum, but primarily in the core modelling, all the programming modules, and the Final Year Project
Analyse and **evaluate** abstract **models** and concrete **implementations**, in specific (limited) contexts, with reference to efficiency, correctness and suitability to users' needs	*Algorithms and Complexity Theory*, *Software Engineering and Development,* all the programming modules, and the Final Year Project
View computing systems critically, both to verify that they are **correct** and to ensure that they are **well-designed**	Across all the programming modules, the *Data Modelling [Database]* module, and the Final Year Project
Critical awareness and **analysis of own** developed computing models and solutions	Final Year Project and all the practical work assignments in the other modules
Propose, **plan** and **evaluate** a **significant** piece of project work, under supervision of an expert	Final Year Project

Practical skills

Graduates should be able to:	**Taught by:**
Turn an abstract model into a fully implemented software system, using a specific and appropriate programming language	All the programming modules and the Final Year Project module
Apply specific tools and **technologies** in the design and implementation of a solution	*Software Engineering*, all the programming modules, *Data Modelling [Databases]*, and the Final Year Project module
Manage development work on a local distributed system (intranet), with reference to storage, communication and documentation	*Audio-visual Computing*, *Software Engineering*, all the programming modules, *Data Modelling [Databases]*, and the Final Year Project module
Program in a specific OO programming language (e.g. Java) and know in detail some of its libraries (packages)	Some of the core and specialist programming modules, and, in most cases, the Final Year Project module
Manage large collections of data	*Data Modelling [Databases]*
Acquire and **manipulate digital media** to a basic level	*Audio-visual Computing*
Execute a significant piece of work, under supervision of an expert.	Final Year Project module

Transferable skills

Graduates should:	Taught by:
Have core numeracy, literacy and IT skills at graduate level	Numeracy and IT skills are core to a computing degree and will feature throughout the curriculum. Students will be required to document, describe and evaluate their work both in traditional reports and on web pages, culminating in their final year dissertation.
Be able to reflect on and critically evaluate their work	Students will be required to maintain a web page on which they will engage in reflective discussion of their work. Software Engineering and Development and final year project will have specific learning outcomes on reflection and self evaluation
Be independent and creative workers and learners	Our degree programmes have a particular focus, unusual in Computing courses, on independent and creative work, starting with 1st year programming and continuing in [Practice Module] and culminating in the final year project. Students will be expected to tackle complete, independent projects of their own devising from the very beginning and will be expected to independently research and learn specialist topics.
Be able to work effectively in groups	Many modules will include group work but the largest scale will be the group project featured in Software Engineering and Development
Be able to present themselves and their work orally and in writing to a professional level	The 1st year Computing Spectrum module will feature a section on self presentation which will then be reinforced in other modules culminating in the final year project

What courses are offered on this programme?

Level 4

IS51008B (CIS115): Introduction to Programming (30)
[Course Code]: Data Representation and Architecture Modelling (15)
[Course Code]: Mathematical Modelling for Problem Solving (15)
[Course Code]: Computing Spectrum (30)
[Course Code]: Audio-Visual Computing (15)
[Course Code]: Computer Science 1 (15)

Level 5

[Course Code]: Programming OO Data Structures and Algorithms (15)
[Course Code]: Programming User Interfaces (15)
[Course Code]: Network Models (15)
[Course Code]: Data Modelling (15)
[Course Code]: Software Engineering and Development (15)
[Course Code]: Algorithms and Complexity Theory (15)
[Course Code]: Internet and Distributed Programming (15)
[Course Code]: Web Programming (15)

Level 6

Every student in the final year will do a research or practice-based project worth 60 credits. Prior to the final year project each student will choose four options of 15 credits each, from a list approved by the department on a yearly basis. Certain choices could be constrained to be made from subsets of topics and it may even be the case that certain topics would be deemed compulsory.

Ideally, the choices made by students will inform their project topics.

A typical list of options is given below

IS53027A (CC342): INNOVATIVE AUDIOVISUAL PROCESSING
IS53032A (CC349): ADVANCED GRAPHICS AND ANIMATION
IS53002A (CIS311): NEURAL NETWORKS
IS53011A (CIS324): LANGUAGE, DESIGN AND IMPLEMENTATION
IS53012A (CIS326): COMPUTER SECURITY
IS53023B (CIS338B): DATA MINING
IS53030A (CIS350): PHYSICAL COMPUTING
IS53013A (CIS323): ELECTRONIC COMMERCE

The structure of the offered courses

There are two “cores” supporting the programme: *programming* and *modelling*. The former focuses on processes, the latter focuses on data.

The programming core includes: *Introduction to Programming* (level 4), *Programming OO Data Structures and Algorithms* and *Programming User Interfaces* (both at level 5). These include more general aspects of programming and are (at least partly) shared with the other Computing programmes.

The modelling core includes: *Data Representation and Architecture Modelling*, *Mathematical Models for Problem Solving*, *Network Models*, and *Data Modelling*. These include more general aspects of data modelling and are (at least partly) shared with the other Computing programmes.

Good practice in programming is particularly addressed through a combination of three courses: *Programming OO Data Structures and Algorithms*, *Software Engineering and Development* and *Algorithms and Complexity Theory*. The first two give a view grounded in relevant practical examples, including also an element of “house-keeping” good practice, whereas the latter gives a more theoretical view, with a focus on efficiency.

The Computer Science specialist modules include: *Web Programming* (level 1), *Advanced Web Programming* (level 2) and *Network and Distributed Programming* (level 2). This reflects the current dominance of the web and the internet.

How will courses be taught?

The Department of Computing is committed to a diverse and stimulating range of learning and teaching methods that ensure the programme outcomes are addressed rigorously and effectively. Learning emphasises a close synthesis between theoretical understanding and practical application that helps you develop an advanced, critical approach to the subject of computing. In addition, the College’s 3-D graduate scheme and personal tutoring system are opportunities to develop coherent links between seemingly disparate elements in the programme.

The various modules of the programme provide a diverse range of topics across the scope of computing but are designed to form a coherent and cumulative body of knowledge and skills. These are further developed through your independent research and learning activities directed towards course assignments and the large-scale project component. The department is committed to providing a diverse and innovative range of teaching styles across its degree programmes. These include traditional lecture and laboratory sessions but also a range of more interactive and self directed activities focusing on independent, creative work and self presentation. The nature of the learning activities will vary greatly between different modules, but includes, design, programming, analysis, planning, group activity and creative work. In addition students will be expected to engage in considerable independent reading and practical work for all modules culminating in the final year project. This independent work will be supported by library resources, access to lab space and supervision from teaching staff.

The programme provides a range of modules which provide a network of cross-referenced and cumulative knowledge across diverse areas of computing. You achieve the outcomes relevant to your individual pathway, that combines core and optional modules, through the experience of interconnected teaching and learning strategies across the various elements of the programme. All modules provide a weekly lecture-lab or other session, which reinforces preparatory or follow-up reading, and other related learning activities in both group and individual settings to foster new understandings and skills.

The main role of and connections between modules was described in the previous section.

How will my work be assessed?

The department recognises that high quality assessment is a vital part of learning, particular when used formatively, providing valuable feedback for future learning. Our assessment is designed to reflect “real world” skills and activity in order to give our students a strong preparation for the work place.

No single method of assessment can capture all aspects of computing or the full range of skills required by our graduates. For this reason we are committed to providing many diverse styles of assessment and to the development and use of novel forms of assessment. Our methods of assessment are designed to reflect business relevant activities and to encourage independent, creative work. As well as traditional examinations, our assessment includes many different types of “hands on” practical work including software development, planning and group work, and presentations. Students will be required to present their work in a number of different ways that reflect the contemporary work place, including traditional reports but also oral presentations and extensive use of the web for self presentation. Above all we encourage our students to be independent and creative thinkers and include considerable opportunities for open ended assessments that allow students to develop their own ideas.

Feedback is vital to effective continuing learning, the true value of assessment is that it shows students how to improve their work and learn more effectively in future. For this reason we are committed to providing timely and full feedback on all assessed assignments.

Throughout the degree programme assessment will happen in individual modules, each having assignments, each including some of the many diverse styles of assessment listed above, as well as end of year exams for some modules. As well as these small assignments, students will have a major project in their final year. This is a **large scale piece of work** which should integrate what students have learned throughout the programme. It provides students with an opportunity to independently tackle a large project that reflects real world software development. There are many different types of project, but all including the implementation of a substantial software system and a written report.

Assessments are expected to make up roughly half of the workload of a taught course. A 15 credit course corresponds to 150 hours of work. Roughly 80 hours of this should be taken up with assessed coursework and examinations (including revision). The remainder is made up of 40 hours of contact time and a further 30 hours of private study.

Below is a list of the major types of assessment used in the department. Individual courses may vary slightly

Practical Coursework

Most of our courses will include an element of practical coursework that includes programming or otherwise creating a software system based on the material presented in the course. You will work independently, with an opportunity to ask for help in lab sessions. You will submit the finished software together with a written report or other type of documentation (oral presentation, web site, in code comments etc.). The assessment of coursework may also involve an oral examination, typically of a random selection of student or where there is suspicion of plagiarism. A 15 credit course will typically have 1 coursework and a 30 credit course will have 2.

There are five main types of coursework that we set, though individual courses may differ slightly.

Practical Coursework (worth up to 40\% of a 15 credit course). This will involved answering a number of specific questions thatinvolve either creating software or hardware from scratch or editing existing software. It will typically include a report of 1-2000 words or equivalent documentation and require about 30 hours of work.

Extended Practical Coursework (worth between 40\% and 80\% of a 15 credit course). This will involved answering a number of specific questions that involve either creating software or hardware from scratch or editing existing software. The work involved will be more substantial than a normal coursework and will also include scope for extending that software in ways that you choose. It will typically include a report of about 3000 words or equivalent documentation and require about 50 hours of work.

Mini-project (worth between 80\% and 100\% of a 15 credit course). This will involve creating a substantial software system either partially or completely of your own design. It may also involve some formative working similar to a practical coursework. It will typically include a report of about 6000 words or equivalent documentation and require about 80 hours of work.

Group project (worth between 80\% and 100\% of a 15 credit course). This will involve creating a substantial software system in a collaboration with a group of other students. The group will submit the completed software, and each individual will write a report of about 5000 words discussing their own contribution to the software and the working of the group. Your mark will be based on the success of the project as a whole and also your contribution to it.. It will typically require about 80 hours of work.

Examined Coursework (worth 100\% of a 15 credit course). Some of our courses will involve a number of practical courseworks or extended practical courseworks that are either partially or completely assessed by a written examination. This examination will consist of questions relating specifically to the coursework.

Written Coursework

Coursework may also take the form of a written essay. This will involve applying the ideas presented in the course and doing independent research or problem solving. There are four types of written coursework that we may set.

Written Problem Sheet (worth up to 40\% of a 15 credit course). This will involve written answer to a set of clearly defined mathematical or technical questions. They will typically require about 30 hours of work.

Essay (worth up to 40\% of a 15 credit course). This will involve writing in answer to a question about a clearly defined topic. It will typically be about 3000 words and require about 30 hours of work.

Extended Essay (worth between 40\% and 80\% of a 15 credit course). This will involve writing in answer to a question about a clearly defined topic, but with more scope for independent research and choice of topic. It will typically be about 6000 words and require about 50 hours of work.

Mini-dissertation (worth between 80\% and 100\% of a 15 credit course). This will involve extensive independent research on a topic that is at least partially defined by you, within the scope of the course. It will typically be about 10000 words and require about 80 hours of work.

Examinations

The purpose of examinations is to test your understanding and work under timed, controlled conditions. Examinations will consist of a number of questions that you will have to answer in a limited time. They will be held in an examination hall in silence. A typical exam for a 15 credit (1 term) course will be 1hour 30 minutes long and consist of 3 questions with no choice, for a 30 credit (2 term) course it will be 3 hours and consist of 6 questions with no choice. Individual courses may have different examination arrangements. Typically you will not be allowed, notes, books or any internet access, though individual exams may allow access to certain books or web sites. There are four major types of examination used in the department:

Written Examinations. These examinations consist of a number of questions to be answered in writing. Typically this will be hand written on exam scripts provided.

Practical Examinations. These examinations will consist of a number of practical questions whose answers require programming or otherwise creative software systems. These examinations will be held in a computer laboratory with no internet access.

Mixed Written/Practical Examinations. These examinations will consist of both written and practical questions. These examinations will be held in a computer laboratory with no internet access.

Coursework Examinations. These are written examinations where the questions are specifically about practical coursework that you will have done during the course (see above).

What do I need to do to progress between levels?

To be confirmed based on College regulations to be published

What are the grading criteria for a BSc (Hons) degree?

Mark	Descriptor	Grading Criteria
0%	Non submission	Work was not submitted or it was plagiarised
1-9%	Very bad fail	A submission that does not even attempt to address the specified learning outcomes (shall be deemed a non valid attempt and unit must be re-sat).
10-24%	Bad fail	Represents a significant overall failure to achieve the appropriate learning outcomes (shall be deemed a valid attempt and not necessarily required to be re-sat).
25-34%	Fail	Represents an overall failure to achieve the appropriate learning outcomes.
35-39%	Pass	Represents the overall achievement of the majority of the appropriate learning outcomes to a pass level. Does not satisfy the requirements for honours level performance, but displays some understanding of concepts, methodology and content. Students should be able to demonstrate creating a very basic computing system under guidance from tutors. Students achieving an overall mark of between 35-39% at degree level will be awarded a pass (non honours) degree.
40-49%	Threshold	Represents the overall achievement of the appropriate learning outcomes to a threshold level (honours). Demonstration of a limited level of understanding of relevant concepts, methodology and content; clear if limited attempt to tackle problems; display of some skill in organisation of material. Students should demonstrate creation of a basic, complete and working computing system/program.

Mark	Descriptor	Grading Criteria
50-59%	Good	Demonstration of an adequate level of understanding of relevant concepts, methodology and content; display of sufficient skill to tackle some complex problems; appropriate organisation of material. Students should demonstrate the ability to create complex computer software, making use of prior knowledge and material taught within the program
60-69%	Very Good	Demonstration of a sound level of understanding based on a competent grasp of relevant concepts, methodology and content; display of skill in interpreting complex material; organisation of material at a high level of competence. Students should be able to demonstrate the ability to independently design, implement and evaluate a high quality and complex computer systems using knowledge from across the program.
70-79%	Excellent	Demonstration of a thorough grasp of relevant concepts, methodology and content appropriate to the subject discipline; indication of originality in application of ideas, in synthesis of material or in implementation; insight reflects depth and confidence of understanding of the material. Students should be able to design and create computer systems that demonstrate considerable independent thought and are based on independent learning of prior work and existing technologies. Students should be able to critically evaluate their own work.
80-100%	Exceptional	Represents an exceptional achievement beyond the standard requirements of a first class degree. Students' work should demonstrate considerable creative thought and be based on a critical evaluation of prior work. Work is likely to achieve some outcomes that would be expected at a higher level degree

What support can I expect?

Expertise is provided by the Departments' resident staff, who are dedicated and experienced teachers, but also distinguished practitioners and researchers in their own right, working in national and international contexts. The Departments also draw on a large pool of visiting tutors and researchers, to provide a breadth of expertise and contact with current research and practice.

Student learning is supported by the Rutherford Information Services Building, which houses extensive book, score, CD/DVD and electronic resources. All registered students also have access to the University of London libraries network. In addition, the Department of Computing has extensive computer lab facilities. The Department make extensive use of the VLE learn.gold online facility, in order to support student learning in a number of ways, including the dissemination of learning resources and to provide an electronic forum for the exchange ideas and debate.

The BSc curriculum is supported by a wide range of activities that encourage awareness and involvement in the Department's high profile practical and research activities, including termly postgraduate conferences, the Digital Studios' 'Thursday Club', the Whitehead Lectures, workshops, visiting speakers, and various other activities of the Digital Studios. Further information about these groups can be found from the Departments' web pages www.gold.ac.uk.

You are allocated a personal tutor during your period of study who offer advice, guidance or clarification of courses, options, requirements and regulations; and to monitor your progress through the programme. The Personal Tutor can also offer support in cases of academic difficulty. Should further advice be necessary, the Senior Tutor, the Chair of the Sub-Board of Examiners can also be consulted. If you encounter difficulties at any time with your studies, the programme convenor and other course tutors can provide additional academic support whilst the Senior Tutor is available by appointment to discuss welfare-centred issues. Staff members have office hours each week to discuss any matters; outside these hours students may arrange an appointment with staff via email or telephone.

The Department of Computing takes advantage of and pursue the College's Disability Awareness policies. Students with specific needs in this regard are considered on an individual basis. The College also actively supports students with specific learning difficulties (e.g. dyslexia), and provisions are made to ensure that all students, regardless of specific difficulty/disability, derive full benefit from the learning environment. In addition to specialist advice and assistance within the College, the Department ensures that course materials are suitable for all students and, where necessary, these are altered to meet the requirements of individual students.

You will develop and maintain a personal development plan, run by the Goldsmiths 3-D Graduate scheme, during your course of study. This helps you record aspirations, plans and goals, record your achievements, and enables progress to be monitored, in order to help achieve your individual aims. The Senior Tutor is available to discuss the 3-D scheme with students, and the Department will advise you about how best to approach this task.

The medical, counselling and financial services provide support for students when necessary, and in the case of students with special needs (including dyslexia), the Student

Support Office will provide sympathetic advice and help. Goldsmiths also provides a wide range of other support services for students, which can be found on its web site at www.gold.ac.uk. Overseas students whose first language is not English may seek assistance from the Goldsmiths English Language Unit.

The Department is committed to making any reasonable adjustment that allows, as far as possible, for equality of opportunity and access, and to ensuring that students are not substantially disadvantaged because of specific learning difficulties or disability.

What Careers will be open to me?

As, upon graduation, you are expected to have strong background in programming and good skills in at least one programming language, you will be a suitable candidate for most of the jobs in the computing industries that do not require a strong specialisation and/or significant work experience. Jobs such as “software developer”, “programmer”, “web developer”, “system analyst”, “database application developer ,etc. in areas including media industries, the health sector, transport, the financial sector, e-government, etc., are all open to you.

As an alternative route, you could consider to continue your professional development with postgraduate studies in any more specialised field of Computing.