PROGRAMME SPECIFICATION

Programme title:	MSc Computer Science
Final award (BSc, MA etc): (where stopping off points exist they should be detailed here and defined later in the document)	MSc
Cohort(s) to which this programme specification is applicable: (e.g. from 2008 intake onwards)	From 2008 intake onwards
Awarding institution/body:	University College London
Teaching institution:	University College London
Faculty:	Engineering Sciences
Parent Department: (the department responsible for the administration of the programme)	Computer Science
Departmental web page address: (if applicable)	http://www.cs.ucl.ac.uk/teaching/msccs/
Method of study: Full-time/Part-time/Other	Full-time
Criteria for admission to the programme:	Please see http://www.cs.ucl.ac.uk/admissions/pg_msccs.html
Length of the programme: (please note any periods spent away from UCL, such as study abroad or placements in industry)	One calendar year
Level on Framework for Higher Education Qualifications (FHEQ) (see Guidance notes)	M
Relevant subject benchmark statement (SBS) (see Guidance notes)	Not applicable. There is currently no Master's level benchmark statement for this subject area.
Brief outline of the structure of the programme and its assessment methods: (see guidance notes)	Please see http://www.cs.ucl.ac.uk/teaching/msccs/msccs_exam.html
Board of Examiners:	Board of Examiners in Computer Science

Professional body accreditation (if applicable):	N/A	Date of next scheduled accreditation visit:

EDUCATIONAL AIMS OF THE PROGRAMME:

The programme aims to give graduates of any discipline the technical skills and understanding required either to pursue a career in the computing/information technology sector or to continue their education in specialized Masters programmes or PhD programmes. Recent graduates of degree programmes that contain a significant computer science component (generally anything more than a couple of introductory units) are not admitted to the programme. The majority of the intake comprises graduates with little or no computing experience from either their previous education or postgraduate employment. The remainder (about a quarter of the intake) are students with limited postgraduate experience of computing who want to deepen their understanding and extend their computing skills. The programme emphasizes the rigorous scientific aspects of computing. In response to the large number of well-qualified applicants without a hard science background, particular attention has been placed on the design and horizontal integration of the first term basic courses so that they constitute a sound but accessible basis for the advanced options for students from disciplines as different as English or Engineering, Music or Mathematics. In the second term, students choose from a wide range of optional courses that reflect the research activity of the department; the major student projects typically explore individual interests in these research areas and are often conducted in collaboration with an active research group.

PROGRAMME OUTCOMES:

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

A: Knowledge and understanding

Knowledge and understanding of: 1. the architecture of computer systems including hardware and systems software 2. the programming of computer systems 3. at least one programming language 4. some of the issues involved in implementing a compiler for a programming language 5. software engineering principles 6. principles of interaction design 7. standard data structures and algorithms 8. fundamental issues of computability and complexity of algorithms	→	**Teaching/learning methods and strategies:** Knowledge and understanding is imparted via traditional lectures, via small-group tutorials, and via individual tuition for the summer project. Throughout, students are encouraged to undertake independent reading both to supplement and consolidate what is being taught/learnt and to broaden their individual understanding and knowledge of the subject. Students are given direction for their independent reading, and lectures are often supported by laboratory work with help from demonstrators. Substantial use is made of the World Wide Web for making lecture notes and other supplementary course material available; the Web is also used to support some coursework submission, and email is used extensively as a forum for students and course-specific emailing lists are used as a communication channel for individual course material.
	→	**Assessment:** Knowledge and understanding in all areas (1-8) is assessed through a variety of courseworks, unseen examination papers and a substantial (3 month) individual project (see below).

B: Skills and other attributes		
Intellectual (thinking) skills: 1. comprehend multiple levels of abstraction in analysis, synthesis and description 2. develop precision of thought and precision in the expression of thought 3. identify problems and apply Computer Science concepts to solve problems 4. reason critically 5. analyse and interpret 6. demonstrate and exercise independence of mind and thought	→	**Teaching/learning methods and strategies:** Intellectual skills are developed through the teaching and learning programme outlined above and elsewhere in this document. Each course, whatever the format of teaching, involves discussion of key issues, practice in applying concepts, and feedback on submitted coursework. The individual project involves weekly supervision with an individual tutor.
	→	**Assessment:** All intellectual skills(1-6) are assessed through a variety of courseworks, unseen examination papers and a substantial (3 month) individual project. The individual project requires each students to analyse requirements and goals (skills 1 and 5), to design a solution that meets the requirements and achieves the goals (skill 2), to implement that solution (normally in the form of a computer program, which assesses skills 2 and 3) and to write a substantial dissertation (typically 60 to 80 pages plus extensive appendices) that includes an appraisal of different techniques (this dissertation assesses skills 2, 4, 5 and 6).
C: Skills and other attributes		
Practical skills (able to): 1. construct correct and reasonably efficient programs using a modern programming language 2. plan, undertake and report on a substantial computer-science-based project	→	All students receive extensive tuition (lectures, laboratory sessions, tutorials etc) in the development of programming skills throughout the first term (where it is a dominant factor) and the second term (where students learn about data structures and algorithms) and the summer (where students normally undertake a programming project).
	→	**Assessment:** Skill 1 is assessed by a combination of practical coursework, the writing of computer programs in answer to unseen examination questions, and in the summer project. The assessment of skill 2 is achieved by the third element of the degree programme – the summer project. This is assessed by dissertation and students may be required to demonstrate their work.

D: Skills and other attributes

Transferable skills (able to):

1. structure and communicate ideas effectively both in writing
2. manage time and work to deadlines
3. participate constructively in groups
4. work independently
5. find information and use IT
6. be self-reliant
7. assess the relevance and importance of the ideas of others

Teaching/learning methods and strategies:

The degree programme contains a substantial amount of coursework, requiring students to develop their time management skills (skill 2) and their communication skills (skill 1). Several of the optional course also include group work (skill 3) and throughout the course the students are encouraged to investigate beyond the limits of the lectures, so developing skill 5. The programme culminates in the summer project, which develops skills 1 – 2, and 4 – 7.

Assessment:

Skill 1 is assessed in unseen examination. Skill 2 is assessed in the courseworks and the summer project. Skill 3 is indirectly assessed within the courses that include group work. Skills 5 and 6 are not directly assessed. Skill 7 is assessed in the dissertation for the summer project.

The following reference points were used in designing the programme:

- the Framework for Higher Education Qualifications (http://www.cs.ucl.ac.uk/financialcomputing/mscfc_programme.htm http://www.qaa.ac.uk/academicinfrastructure/benchmark/default.asp;
- the programme specifications for UCL degree programmes in relevant subjects (where applicable);
- UCL teaching and learning policies;
- staff research.

Please note: This specification provides a concise summary of the main features of the programme and the learning outcomes that a typical student might reasonably be expected to achieve and demonstrate if he/she takes full advantage of the learning opportunities that are provided. More detailed information on the learning outcomes, content and teaching, learning and assessment methods of each course unit/module can be found in the departmental course handbook. The accuracy of the information contained in this document is reviewed annually by UCL and may be checked by the Quality Assurance Agency.

Programme Organiser(s) Name(s):	Dr R D Hirsch
Date of Production:	November 2002
Date of Review:	April 2008
Date approved by Head of Department:	TBA
Date approved by Chair of Departmental Teaching Committee:	TBA
Date approved by Faculty Teaching Committee	TBA