Faculty of
Computing, Engineering
and Mathematical Sciences

BRISTOL

Faculty of Computing, Engineering & Mathematical Sciences

Joint Honours in Internet Technology

Definitive Document – February 2004

Programme Specification

Section 1: Basic Data

Awarding institution/body	UWE
Teaching institution	UWE
Faculty responsible for programme	Computing, Engineering and Mathematical Sciences
Programme accredited by	N/A
Highest award title	BSc (Hons) Internet Technology & ….
Default award title	
Interim award title	BSc Internet Technology & … Dip HE Internet Technology & … Cert HE Internet Technology & …
Modular Scheme title (if different)	Faculty of Computing, Engineering and Mathematical Sciences Modular Scheme
UCAS code (or other coding system if relevant)	
Relevant QAA subject benchmarking group(s)	Computing
On-going/valid until* (*delete as appropriate/insert end date)	
Valid from (insert date if appropriate)	1 September 2004

Authorised by… **Date:…**

Version Code 2

For coding purposes, a numerical sequence (1, 2, 3 etc.) should be used for successive programme specifications where 2 replaces 1, and where there are no concurrent specifications. A sequential decimal numbering (1.1; 1.2, 2.1; 2.2 etc) should be used where there are different and concurrent programme specifications

Section 2: Educational Aims of the Programme

The joint-programme in Internet Technology has the following general aims:

1. To prepare students for computing careers in business, industry, and commerce, in particular where extensive use is made of Internet technology.

2. To develop problem-solving and other transferable skills that will be valuable in any career.

3. To prepare students for progressing to study for higher degrees in Computing.

4. To continue the development of general study skills that will enable students to become independent, lifelong learners.

The joint-programme in Internet Technology has the following specific aims:

1. To provide insight into, and practical skills in, the creation of Internet-based computer systems. This involves understanding the nature of the technology and how it might best be utilized.

2. To enable students to make an immediate contribution to companies engaged in extensive use of the Internet and web-based development.

3. To develop the students' understanding of the importance of problem solving in any domain, though with particular reference to the development of internet-based software.

4. To encourage the discerning use of reference material from a variety of sources.

Section 3: Learning Outcomes of the Programme

The award route 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:	Teaching/Learning Methods and Strategies	Assessment
1. Object-oriented programming concepts; syntax and semantics; programming to satisfy designs. OO Programming Language. 2. Internet tools and Unix / Linux directory management 3. System descriptive notations 4. The characteristics of data, data modeling using XML data type definitions and E-R diagrams and the use of SQL. 5. Program design concepts, methods, and notations; object-oriented design paradigms; algorithms; design patterns. 6. Client-Server architecture and the different models used in developing both clients and servers. CSP 7. The concept of usability and usability standards underpinning the design and evaluation of user interaction with computer systems. 8. The concepts underlying the reuse of components in software development and related research issues.	On all modules, at all levels, students are encouraged to undertake: • independent reading both to supplement and consolidate what is being taught / learnt and to broaden their individual knowledge of the subject. • practice of technical skills so that they become fluent in their use. The programme of study is designed to introduce the knowledge and understanding necessary to engage, from the beginning, in appreciating and solving small-scale problems. At level 1, fundamental principles and techniques are introduced so that students acquire appropriate background knowledge to enable them to start making judgments about suitability of alternate approaches. The in-depth understanding of larger, more complex, real-world problems, essentially starts with level 2 study. Level 3 continues to increase the in-depth knowledge and understanding of technical solutions for larger scale problems. At level 1, knowledge and understanding of Object-oriented programming language concepts; fundamental tools and notations and database construction are introduced on three modules that explore these general approaches. More in-depth knowledge and specific understanding of all topics follows in subsequent levels.	Testing of the knowledge base is through: Assessed written work (topics: 3, 5, 7, 10); Assessed practical work (topics: 1, 2, 3, 4, 6, 7, 9, 11); Examination (topics: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10).

Knowledge and Understanding of:	Teaching/Learning Methods and Strategies	Assessment
9. The state of the art in web software technologies including JavaScript interaction with HTML and web servers. 10. Security threats and approaches to preventing them. 11. How to plan and undertake a small application development project.	At level 2 the deepening of the knowledge and understanding of software development continues with an expansion into design approaches, the use of a client / server architecture and the assessment of designs from a user perspective. The development of specialized and more specific knowledge and understanding continues at level 3. Use of some 'half-modules' for the first time allows in-depth study of sharply focused topics such as Internet Security. Use of some 'standard sized' modules provides fuller coverage of the important topic of Component-based Software development and also allows flexibility by requiring students to bring together their knowledge to develop an appropriate application.	

B. Subject Specific Skills

Subject Specific Skills Students will be able to:	**Teaching/Learning Methods and Strategies**	**Assessment**
1. Write Java programs that conform to UML class diagrams and pseudocode algorithm designs. 2. Use Unix / Linux and Internet tools to build systems. 3. Employ system descriptive notations. 4. Undertake data analysis using appropriate tools and implement the resulting model as a database application. DB 5. Use OO analysis and design techniques together with an OO IDE to construct software. 6. Develop a small client/server application in an OO language using appropriate tools. CSP 7. Apply user-centered design and undertake usability analysis. 8. Develop and document components and integrate them within programs. 9. Design dynamic Web-based information presentation and configure, run and maintain a web server. 10. Identify possible internet security threats, recommend solutions and implement them. 11. Design and implement a significant application. ADP	Throughout the programme, the skills listed are developed through a combination of theoretical discussion, practical laboratory-based work, classroom based tutorial exercises and directed self-study. A number of the fundamental skills listed (1, 2, 3, 4) are introduced at level 1 and then further developed at levels 2 and 3. Much of the work at level 2 and some at level 3 provides generally applicable software development skills (5, 7, 8). As the programme progresses some very specific skills (6, 9, 10) are introduced at both levels 2 and 3. These are underpinned by the more generalized capabilities (1, 2, 3, 4, 5) that are practiced throughout the levels in most of the modules that contribute to the programme. The culmination of skill development is seen with the development of a significant Application Project. (11)	The possession of these skills is demonstrated both by the development of a practical piece of coursework (software) and by examination. The practical nature of many of the skills to be acquired means that particular modules specifically address skills 4, 6, 7, 8, 9, 10 and 11. The more generic skills 1, 2, 3, and 5 are assessed across the modules. For example, the module Client-Server Programming requires the students to design and implement applications using a client server architecture in appropriate programming technologies (6) as part of the coursework assessment. The examination allows students to demonstrate that they have grasped the underlying concepts that inform the development of such an artifact. Skills such as conformance to requirements and design and (1) are fundamental to professional software development of any sort and thus contribute to the assessment of all the practical work produced.

C. Cognitive (Intellectual) Skills

Cognitive (Intellectual) Skills	Teaching/Learning Methods and Strategies	Assessment
1. Critical Thinking 2. Analysis 3. Synthesis of different types of information 4. Evaluation 5. Problem Solving 6. Appreciate problem contexts 7. Balance conflicting objectives	At all levels students are required to bring together knowledge and skills acquired in several modules and hence determine new ways of working. As the student progresses, the need to synthesise (3) ever-greater volumes of information and approaches into a coherent approach is developed and consequently so is their critical thinking (1). At level 1 Analysis (2), Evaluation (4) and Problem Solving (5) are developed on small-scale problems in various programming activities in a number of modules. Here the focus is on understanding the problem and then solving it free from the environmental implications of real-world problems and without the need to examine alternatives and to balance conflicting goals. At level 2 there is a move away from small-scale relatively trivial problems to the consideration of larger scale more complex systems. With this comes the need to evaluate (4) alternative methods and designs and to balance conflicting objectives (7). Level 3 sees the move to yet more sophisticated techniques and more complex examples and with it the need to appreciate problem contexts (6) is developed as well as striking the right balance when facing conflicting objectives (7).	Programming of complex software requires demonstration of all of the intellectual skills. At level 1 the focus in programming coursework assessment, is on the skills of Analysis (2), Evaluation (4) and Problem Solving (5). At levels 2 and 3 this branches out to include all the remaining skills. Many of the coursework assessments and exam papers include elements of programming work. Independent reading is used to enable students to focus on their own areas of interest and in the process asses skills 1-4 in the submitted reports, essays and exam answers. Design-work, even when not implemented in a programming language, requires demonstration of skills 1, 2, 5, 6, 7 and a number of coursework assessments and exam questions are devoted to such work. Finally, all of the examinations assess skills 1-4 whilst skills 5-7 are covered in many exams.

D. Key (Transferable) Skills

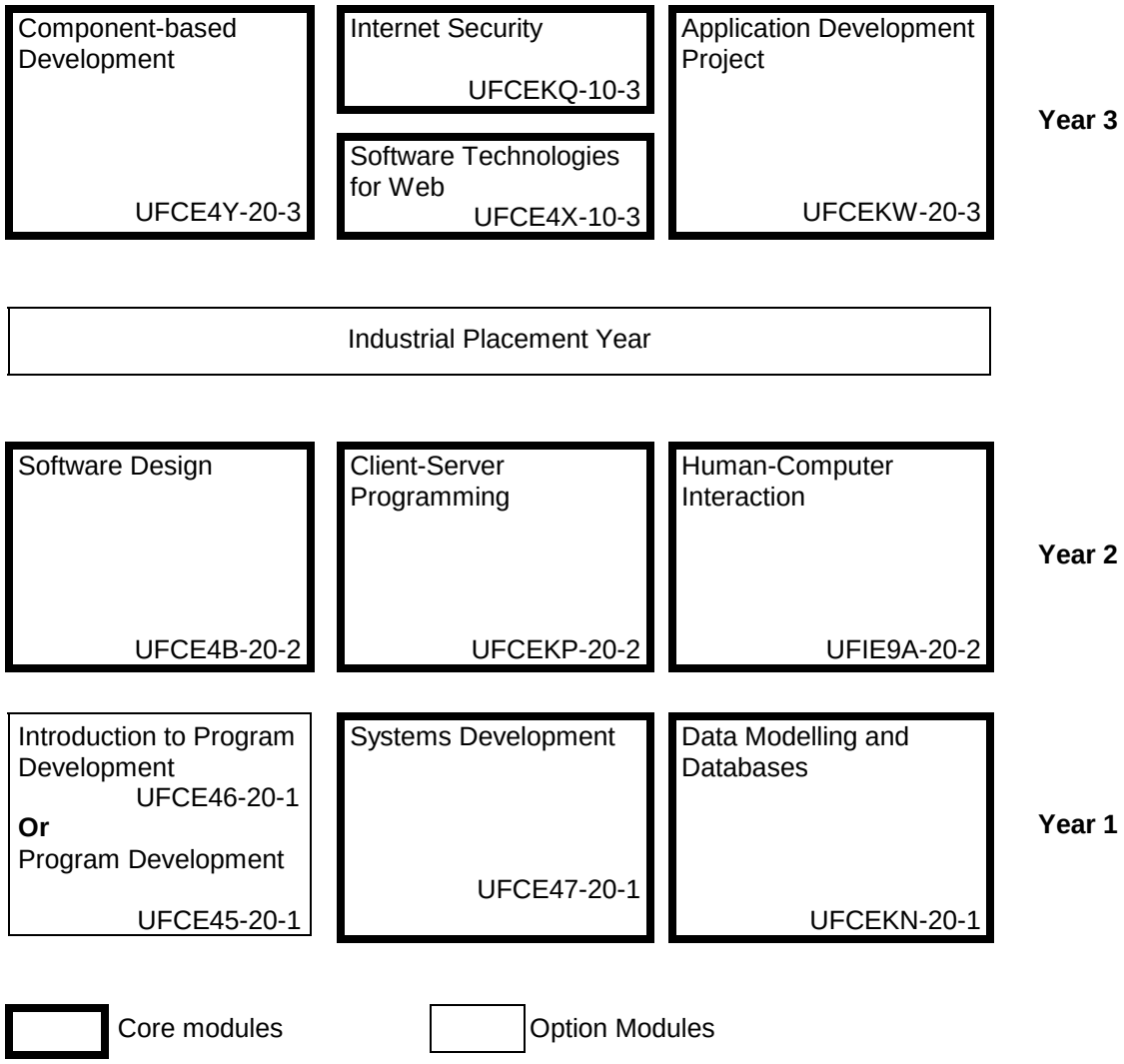
Key (Transferable) Skills	Teaching/Learning Methods and Strategies	Assessment
1. Communication skills**:** to communicate orally or in writing.	1. Skill one is developed through a variety of methods and strategies including the following: ♦ Students participate in electronic conferences, workshops, and groupwork sessions. ♦ Students participate in discussion tutorials ♦ Students present research topic findings in tutorials ♦ Students participate in individual tutorials	All of the skills are demonstrated in varying degrees in all of the employed assessments with the exception of teamwork, which is covered in some of the coursework. It would be impossible to progress to completion on the half-award without demonstrating a basic competence in all of these skills.
2. Self-management skills: to manage one's own time; to meet deadlines; to work with others having gained insights into the problems of team-based software development.	2. Skill two is developed through a variety of methods and strategies including the following: ♦ Students conduct self-managed practical work ♦ Students participate in practically-oriented tutorial laboratory sessions ♦ Students work through practical work-sheets in teams ♦ Students practice design and programming ♦ Students participate in electronic conferencing tutorials and electronic group-working	
3. IT skills in context: to use software tools in the context of application development.	3. Skill three is developed through a variety of methods and strategies including the following: ♦ Students conduct self-managed practical work ♦ Students participate in experimental investigation tutorials ♦ Students work through practical work-sheets in teams ♦ Students make use of online teaching materials ♦ Students are encouraged to practice programming to extend their skills	

Key (Transferable) Skills	Teaching/Learning Methods and Strategies	Assessment
4. Problem Formulation and Decision-Making: To undertake analysis and interpretation of information and express problems in appropriate notations.	4. Skill four is developed through a variety of methods and strategies including the following: ♦ Students develop problem solving programs ♦ Students practice design and programming in a number of different languages ♦ Students sketch designs of larger systems	
5. Progression to independent learning: To gain experience of, and to develop skills in, learning independently of structured class work. For example, to develop the ability to use on-line facilities to further self-study.	5. Skill five is developed through a variety of methods and strategies including the following: ♦ Students are encouraged to practice programming to extend their skills ♦ Students are encouraged to research relevant topics ♦ Students are encouraged to use online facilities to discover information	
6. Awareness of professional literature: to read and to use literature sources appropriate to the discipline to support learning activities	6. Skill six is developed through a variety of methods and strategies including the following: ♦ Students are encouraged to access online material ♦ Students review the literature for discussion in tutorial classes and electronic conferences	
7. Teamwork: to be able to work as a member of a team; to be aware of the benefits and problems which teamwork can bring	7. Skill seven is developed through a variety of methods and strategies including the following: ♦ Students are required to participate in electronic conferences ♦ Students will develop software in small groups	

Section 4: Programme Structure

Joint Honours Internet Technology

Note: this structure is indicative and subject to change



Section 5: Entry Requirements

The university's minimum requirements for entry to a degree apply to this programme. In addition entrants are required to have Mathematics at GCSE Grade C or equivalent.

Section 6: Assessment Regulations

The university's Modular Assessment Regulations apply to this programme.

Section 7: Student Learning: Distinctive Features and Support

Within the Faculty of Computing Engineering and Mathematical Sciences, student learning is supported in the following ways:

Class Activities The mode of delivery of a module is determined by its Module Leader, and typically involves a combination of one or more lectures, tutorials, 'lectorials', laboratory classes, group activities and individual project work.

Academic Support Academic advice and support is the responsibility of the staff delivering the module in question. Staff are expected to be available outside normal timetabled hours, either by appointment or during published "surgery" hours, in order to offer advice and guidance on matters relating to the material being taught and on its assessment.

Pastoral Care The faculty offers pastoral care through its Student Advisers, a team of staff who provide comprehensive, full-time student support service on a drop-in basis and by appointment. All students on the same route are allocated to the same Adviser, who is trained to provide advice on matters commonly of concern, including regulatory and other matters. The Adviser will advise the student to seek advice from appropriate professional services including the university's Centre for Student Affairs or from members of academic staff.

Progression to Independent Study
Many modules require students to carry out independent study, such as research for projects and assignments, and a full range of facilities are available to help students with these. Accordingly, the philosophy is to offer students both guided support and opportunities for independent study. Guided support, mainly in the form of timetabled sessions, which students are expected to attend.

The progression to independent study is also assisted by the nature of the support offered in individual modules. Typically, module leaders will provide a plan for the module indicating the activities to be carried out and the forms of learning to be undertaken during the delivery of the module, with a view to encouraging students to plan ahead and to take responsibility for managing their time and resources.

Computing Facilities The Faculty offers a specialised computing facility along side the general University provision. There are nine general PC computing laboratories of 20 plus seats all running Windows2000, along with four Unix based and 10 specialist computing laboratories. The specialist laboratories are equipped with software for CEMS students including Software Tools and Development Environments.

One of the most popular areas within the Faculty is the Open Access laboratory. This area is never time-tabled and gives students the opportunity to access machines at all times during opening hours. This is a mixed environment consisting of PCs and Unix workstations.

Due to the extensive and specialist nature of the computing facility provided within the Faculty there is a need for user support. The Faculty provides a user support Helpdesk that provides first line support

to users. It is uniquely supported by both permanent staff and students who are in their second or final year of study and are employed on a part time basis. The helpdesk is open from 08.30 hrs until 20.00hrs every day. The general laboratories are available to students up until midnight, seven days per week.

Section 8 Reference Points/Benchmarks

The QAA Subject Benchmark Statement for Computing was published in 2000, and is applicable to this proposal. The design team has considered it in drawing up the structure of the proposed half-degree, and is of the view that the proposal falls clearly within the scope of the benchmarks, as regards curriculum, teaching and learning, and the benchmarking standards themselves.

The benchmarks (paragraph 2.1) identify a range of types of degrees in computing. At one extreme is a programme that "*covers a wide range of topics spanning the entire area of computing".* At the other programmes that "*take one very specific aspect of computing and covers it in great depth"*. This proposal is closer to the second of these extremes.

The benchmarks recognise (paragraph 3.3) that diversity of provision is to be encouraged, and hence joint degrees have an important place. Nevertheless, there are inevitably constraints on the breadth of coverage of the subject possible within a "half-degree". The design team has faced these constraints as part of the course design as set out in the benchmarks (paragraph 3.1), and it believes that it has successfully met them all to the extent that it is possible to do so within the half-degree structure.

The benchmarks also contain (section 5) statements of the standards expected of graduates at both modal and threshold levels. The team is of the view that graduates of the proposed programme will be able to meet the required standards, albeit in some cases with a lesser breadth of knowledge than would be expected of a graduate in a full honours degree in Internet Computing.

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 individual modules are to be found in the module specifications.

The programme will be delivered in accordance with the faculty's Teaching, Learning and Assessment Strategy which has in turn been informed by the university's Learning & Teaching Strategy.