University of the West of England
BRISTOL

ACADEMIC SERVICES

PROGRAMME SPECIFICATION

Part 1: Basic Data		
Awarding Institution	UWE, Bristol	
Teaching Institution	UWE, Bristol	
Delivery Location	UWE, Bristol; Villa College, The Maldives	
Faculty responsible for programme	Environment and Technology	
Department responsible for programme	Computer Science and Creative Technologies	
Modular Scheme Title		
Professional Statutory or Regulatory Body Links	BCS (The Chartered Institute for IT) – accreditation for UWE, Bristol only	
Highest Award Title	BSc(Hons) Computer Science	
Default Award Title		
Interim Award Titles	BSc Computer Science Dip HE Computer Science Cert HE Computer Science	
UWE Progression Route		
Mode(s) of Delivery	Full time, Part time, Sandwich	
Codes	**UCAS: G400**	**UCAS:**
	ISIS2: G50013(FT); G500 (SW); G50F43 (PT); I1012 (Villa)	**ISIS2:**
Relevant QAA Subject Benchmark Statements	Computing	
CAP Approval Date	November 2017 v5	
Valid From	September 2018 v5	
Periodic Curriculum Review	June 2013	
Valid until Date	June 2019	
Version	5	

Part 2: Educational Aims of the Programme

The BSc in Computer Science has the following general aims:

1. To prepare students both for entry into the computing profession, and for the more general challenges of professional and personal life.
2. To inculcate in students problem-solving and other transferable skills that will be valuable to them in any career.
3. To prepare students for progression to study for higher degrees in Computer Science.
4. To continue the development of those general study skills that will enable students to become independent, lifelong learners.

The BSc in Computer Science has the following specific aims:

1. To enable a student to obtain employment in any area of computer science, such as artificial intelligence, systems development, algorithm design, or networking.
2. To provide a coherent and broad based coverage of the theory of computer science and its application to practical problems.
3. To enable students to appreciate the problems that can arise in computer science and to provide them with the appropriate skills to select and apply appropriate methods and technologies to solve them.
4. To encourage students to uphold professional, ethical and social standards and to keep up to date with recent technological and theoretical developments.

Part 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:

Learning Outcomes	**Teaching, Learning and Assessment Strategies**
A Knowledge and Understanding	
A Knowledge and understanding of 1. Object-oriented programming language concepts; other programming paradigms; syntax and semantics; programming to satisfy designs. 2. Program design concepts, methods, and notations; object-oriented design and other design paradigms; algorithms; design patterns. 3. Databases; logical and physical database design; database query languages. 4. The concepts underpinning distributed systems and networks. 5. The concepts underpinning World-Wide Web technology and web-based application development. 6. Design and analysis of a variety of classes of algorithms. 7. The architecture and main components of computers. 8. A range of software development lifecycle methods, e.g.: OOA, OOD, and OOP. 9. Discrete mathematics, propositional logic, and predicate logic. 10. Professional, ethical, legal and social issues 11. The commercial context of software development 12. Formal systems: syntax and semantics 13. Artificial intelligence concepts, notations, and	**Teaching/learning methods and strategies:** The award is designed to introduce both the main concepts and topics of computer science, such as the design and analysis of algorithms, object oriented systems development, and AI, and the knowledge and understanding necessary to engage, from the beginning, in appreciating and tackling computer-science problems. Students are introduced at each level to modules that develop a gradually increasing appreciation of the main concepts of computer science (1, 2, 6, 12, 13). At level-1, the context in which these issues reside is introduced, but the in-depth understanding of large, complex, problems essentially starts with level-2 study and then continues into level-3. At level-one, knowledge and understanding of topics 1, 2, 4, 6 and 12 (object-oriented programming language concepts; program design concepts; design and analysis of algorithms; AI; and networks and operating systems) is introduced on modules which explore the general concepts, components and issues, positioning them in the computing environment. The general understanding of topics 1, 2 and 6 is built on with more in-depth knowledge and

Part 3: Learning Outcomes of the Programme	
methods: for example, agents, machine learning and evolutionary algorithms	specific understanding of application at higher levels. Topic 7, “The architecture and main components of computers.” is taught only at level 1 providing, for this programme, fundamental technical knowledge and understanding. At level-2 the knowledge and understanding of computing continues with an expansion into broader and larger issues, such as, the design of, and methods of building larger software systems. The complexity and design of such systems is addressed in all level-2 modules. Moreover, in-depth knowledge and understanding of topics 2-4, 6 and 13 (Program design concepts; concepts underpinning distributed systems and networks; design and analysis of algorithms; and AI) is delivered in these modules. At level-2, knowledge of topic 1 is assumed but will be consolidated by constant review and usage. At level-3 students continue to broaden and deepen their knowledge of, and practical skills with, object-oriented systems development and AI. They also study ethical, legal, social and professional issues in computing. And they are able to obtain in-depth knowledge in an area of computer science of their own choosing. On all modules, at all levels, the learner is encouraged to undertake independent reading both to supplement and consolidate what is being taught/learnt and to broaden their individual knowledge of the subject.
	Assessment: Most of the knowledge and understanding outcomes are assessed by examination (1, 2, 3, 4, 5, 6, 8, 9, 12 and 13). In addition, a variety of other assessment instruments are used to assess these outcomes, including the following: • Extended individual project (1, 2, and any of 3-13) • In class test (7 and 9) • Portfolio of practical work (12) • Programming exercises (6) • Coursework assignment (1,2) • In class demonstration (1,2, 5, 8, 13) • Programme assignment (4, 7) • Online problems (13) • Portfolio of tasks (9, 12) • Weekly worksheets (4, 7)
B Intellectual Skills	
B Intellectual Skills 1. Critical Thinking 2. Analysis 3. Synthesis of different types of information 4. Evaluation	**Teaching/learning methods and strategies:** 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-

Part 3: Learning Outcomes of the Programme	
5. Problem Solving 6. Appreciate problem contexts 7. Balance conflicting objectives 8. Construction of logical arguments 9. Discussion and debate about technical 10. subjects with peers	greater volumes of information and approaches into a coherent approach is developed and consequently so is their critical thinking (1) as well as their ability to discuss and debate technical subjects with peers (9). At level-1 Analysis (2), Evaluation (4) and Problem Solving (5) are developed on small-scale problems in various 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 problems to the design of larger scale 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 larger scale application 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). The skill of constructing a logical argument is inculcated in students in part when they develop cases to support decisions they have made to resolve conflicting objectives (8).
	Assessment: Programming of complex software requires demonstration of all of the intellectual skills. At level-one, the focus in programming coursework assessment, undertaken in a number of modules, is on the skills of Analysis (2), Evaluation (4) and Problem Solving (5). At level-2 and level-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, subsequent reports and examinations allow skills 1 – 4 to be assessed. 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. Many of the assignments and the individual project require students to express logical arguments, 8. Finally, all of the examinations assess skills 1-4 whist skills 5-7 are covered in many exams.
C Subject, Professional and Practical Skills	
C Subject, Professional and Practical Skills 1. Write programs that conform to designs 2. Create high-level and low-level designs that correspond to stated requirements 3. Design, build, and deploy databases to meet application requirements	Teaching/learning methods and strategies: Throughout the program, the skills listed are developed through a combination of the following devices: • Theoretical discussion • Practical laboratory-based work

Part 3: Learning Outcomes of the Programme	
4. Perform adequate tests on programs 5. Elicit and express requirements for software systems 6. Build web-based systems 7. Employ a range of tools and notations to support the activities listed here: e.g. editors, compilers, software development environments, HTML, CGI, Java, etc. 8. Design algorithms using standard techniques; evaluate and compare algorithms with regard to domain problems; use mathematical techniques to analyse algorithm complexity; apply algorithms appropriately to real-world tasks.	• Classroom-based tutorial exercises • Directed self-study Many of the skills listed are introduced at level-1 and then drawn into sharper focus at level-2, and deepened at level-3. The general teaching/learning method is to impart these practical/professional skills by a process of moving from an overview of what is required to a specific application of an individual skill at a higher level. **Assessment:** The two main forms of assessment of the subject, professional and practical skills are as follows: • Extended individual project (1, 2, 4, 5, 7 and optionally any of 3, 6 and 8) • Practical component of individual assignment projects (1 – 8) In addition, other assessment instruments are used to assess some of the skills: • Examination (1, 2, 3, 5, 8)
D Transferable Skills and other attributes	
D Transferable Skills and other attributes 1. Communication skills: to communicate orally or in writing. 2. Self-management skills: to manage one's own time; to meet deadlines; to work with others. 3. IT skills in context: to use software tools in the context of application development. 4. Logical reasoning skills: to undertake analysis and interpretation of information in the context of the computer science discipline. 5. Problem formulation: to express problems in appropriate notations. 6. 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. 7. Comprehension of professional literature: to read and to use literature sources appropriate to the discipline to support learning activities.	**Teaching/learning methods and strategies:** 1. Communication skills are developed through a variety of methods and strategies including the following: • Students maintain laboratory log books • Students participate in electronic workshops, and group work sessions. • Students participate in discussion tutorials • Students present research topic findings in tutorials • Students participate in individual tutorials 2. Self-management skills are 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 group-working tutorials 3. IT skills in context are 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 use a range of system development

Part 3: Learning Outcomes of the Programme	
	tool, methods, and packages • Students are encouraged to practice programming to extend their skills • Students make sustained use of the internet 4. Logical reasoning skills are developed through a variety of methods and strategies including the following: • Students develop problem-solving programs • Case-Studies are used to explore design issues with students • Students practice design and programming • Students sketch designs of larger systems 5. Problem formulation skills are developed through a variety of methods and strategies including the following: • Students develop problem solving programs • Students practice design and programming 6. Progression to independent learning 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 the library, the internet and other online facilities to discover information and broaden knowledge • Students are encouraged to articulate and reflect upon their own ideas and experiences • Students negotiate the content and structure of their individual projects with tutors 7. Comprehension of professional literature is developed through a variety of methods and strategies including the following: • Students are encouraged to access online material • Material is recommended to the students in module syllabi and by tutors • Students are required to research and refer to appropriate literature in assignments and the individual project
	Assessment: 1. Communication skills are assessed mainly by examination, but also by in-class tests, essays, presentations and poster presentations. 2. The other skills are assessed through a number of similar instruments including the following: • Individual and group projects • Practical assignments • Portfolio of exercises 3. In addition self-management skills are assessed by both peers and tutors through GDP sessions and generally throughout the course.

Part 4: Programme Structure (UWE)				
This structure diagram demonstrates the student journey from Entry through to Graduation for a **full time student**, including: level and credit requirements, interim award requirements, module diet, including compulsory and optional modules				
ENTRY ↓	Year 1	Compulsory Modules **UFCFC3-30-1** Introduction to OO Systems Development **UFCF93-30-1** Computer and Network Systems **UFCFD3-30-1** Introduction to Artificial Intelligence **UFCFA3-30-1** Principles of Computing	Optional Modules None	Interim Awards Cert HE in Computer Science 120 credits, of which not less than 100 are at Level 1 or above.
	Year 2	Compulsory Modules **UFCFB6-30-2** OO Systems Development 2 **UFCFB4-30-2** Intelligent Systems **UFCFW4-30-2** Design and Analysis of Data Structures and Algorithms	Optional Modules **30 credits from:** **UFCFV4-30-2** Data, Schemas and Applications **UFCFK6-30-2** Software Engineering **UFCFVK-15-2** Internet of Things **UFCFWK-15-2** Operating Systems	Interim Awards Dip HE in Computer Science 240 credits, of which not less than 100 are at Level 2 or above and a further 120 are at Level 1 or above.
	Year Out: Students on the Sandwich route complete a placement year. Students on placement complete a professional practice module and are awarded 15 level 3 credits. The professional practice module is shown in the option list for year 3 but is actually completed during the year out.			

	Year 3	Compulsory Modules **UFCF85-30-3** Enterprise Systems Development **UFCFY3-15-3** BioComputation **UFCFB5-15-3** Ethical and Professional Issues in Computing and Digital Media **New Level 1 students registered from 2017/18 must take:** **UFCFXK-30-3** Digital Systems Project **Students registered before 2017/18 must take one of the following:** **UFCFXK-30-3** Digital Systems Project **OR** **UFCFR4-45-3** Computing Project	Optional Modules **Students must take 30 credits from the lists below, of which a maximum of 15 credits can be at Level 2** **EITHER** one from List-one and one from List-two; **OR** two from List-two **List-one:** **UFCFE6-15-3** Professional Experience OR **UFCFWJ-15-3** International Experience (The above two modules are only available to students who have taken a placement year) OR **UFCFVJ-15-3** Professional Development **List-two:** **UFCFM6-15-3** Requirements Engineering **UFCFU3-15-3** Advanced Databases **UFCFT4-15-3** Cryptography **UFCF95-15-3** Entrepreneurial Skills **UFCF7H-15-3** Mobile Applications **UFCFR5-15-3** Advanced Topics in Web Development 2 **UFCFHC-15-3** Usability and Interaction Design **UFCFVK-15-2** Internet of Things **UFCFWK-15-2** Operating Systems **UFCFJK-15-M** Cloud Computing **UFCFYK-15-3** Readings in Artificial Intelligence	Interim Awards BSc Computer Science 300 credits with at least 60 credits at level 3, plus a further 100 credits at level 2 or above and a further 120 credits at level 1 or above Highest award BSc(Hons) Computer Science 360 credits, of which at least 100 must be at Level 3 or above, at least a further 100 at Level 2 or above and a further 140 at Level 1 or above.

GRADUATION

Part 4: Programme Structure (Villa College, The Maldives)

This structure diagram demonstrates the student journey from Entry through to Graduation for a **full time student**, including: level and credit requirements, interim award requirements, module diet, including compulsory and optional modules

ENTRY

	Compulsory Modules	Optional Modules	Interim Awards
Year 1	**UFCFC3-30-1** Introduction to OO Systems Development **UFCF93-30-1** Computer and Network Systems **UFCFD3-30-1** Introduction to Artificial Intelligence **UFCFA3-30-1** Principles of Computing	None	Cert HE in Computer Science 120 credits, of which not less than 100 are at Level 1 or above.
Year 2	**UFCFB6-30-2** OO Systems Development 2 **UFCFB4-30-2** Hybrid Intelligent Systems **UFCFW4-30-2** Design and Analysis of Data Structures and Algorithms	**Options should be selected up to a value of 30 credits (subject to change and availability).** **UFCFVK-15-2** Internet of Things **UFCFWK-15-2** Operating Systems **UFCFV4-30-2** Data, Schemas and Applications **UFCFK6-30-2** Software Engineering	Dip HE in Computer Science 240 credits, of which not less than 100 are at Level 2 or above and a further 120 are at Level 1 or above.

Year Out: Students on the Sandwich route complete a placement year. Students on placement complete a professional practice module and are awarded 15 level 3 credits. The professional practice module is shown in the option list for year 3 but is actually completed during the year out.

	Year 3	Compulsory Modules **UFCFXK-30-3** Digital Systems Project **UFCF85-30-3** Enterprise Systems Development **UFCFY3-15-3** BioComputation **UFCFB5-15-3** Ethical and Professional Issues in Computing and Digital Media	Optional Modules **Students must take 30 credits from the lists below, of which a maximum of 15 credits can be at Level 2** **EITHER** one from List-one and one from List-two; **OR** two from List-two **List-one:** **UFCFE6-15-3** Professional Experience OR **UFCFWJ-15-3** International Experience (The above two modules are only available to students who have taken a placement year) OR **UFCFVJ-15-3** Professional Development **List-two:** **UFCFM6-15-3** Requirements Engineering **UFCFU3-15-3** Advanced Databases **UFCFT4-15-3** Cryptography **UFCF95-15-3** Entrepreneurial Skills **UFCF7H-15-3** Mobile Applications **UFCFR5-15-3** Advanced Topics in Web Development 2 **UFCFHC-15-3** Usability and Interaction Design **UFCFVK-15-2** Internet of Things **UFCFWK-15-2** Operating Systems **UFCFJK-15-M** Cloud Computing **UFCFYK-15-3** Readings in Artificial Intelligence	Interim Awards BSc Computer Science 300 credits with at least 60 credits at level 3, plus a further 100 credits at level 2 or above and a further 120 credits at level 1 or above Highest award BSc(Hons) Computer Science 360 credits, of which at least 100 must be at Level 3 or above, at least a further 100 at Level 2 or above and a further 140 at Level 1 or above.

GRADUATION

Part 5: Entry Requirements

The University's Standard Entry Requirements apply with the following additions:

Successful applicants normally require a minimum of (the equivalent of) 300 UCAS points

Part 6: Assessment

Approved to University Regulations and Procedures

Part 7: Student Learning

Teaching, learning and assessment strategies to enable learning outcomes to be achieved and demonstrated

At UWE, Bristol there is a policy for a minimum average requirement of 12 hours/week contact time over the course of the full undergraduate programme. This contact time encompasses a range of face-to-face activities as described below. In addition a range of other learning activities will be embedded within the programme which, together with the contact time, will enable learning outcomes to be achieved and demonstrated.

On the BSc (Hons) Computer Science programme, teaching is a mix of scheduled, independent and placement learning.

Scheduled learning includes lectures, seminars, tutorials, project supervision, demonstration, practical classes, external visits and field trips. Scheduled sessions may vary slightly depending on the module choices made

Independent learning includes hours engaged with essential reading, case study preparation, assignment preparation and completion etc. These sessions constitute an average time per level as indicated in the table below. Scheduled sessions may vary slightly depending on the module choices made.

Placement learning may include a practice placement. If a student opts to take a placement year, they are required to takethe level-three module, Professional Experience, while they are undertaking the placement. They may alternatively take a study year abroad, in which case they take the International Experience module.

Description of Distinctive Features and Support

Class-based Activities Classes use a range of activities. The particular mode of delivery of a module is determined by its Module Leader, and typically involves a combination of one or more lectures, practical sessions, group activities and group project work. Modules on the programme that require laboratory classes are commonly delivered by means of a combination of lecture and practicals/tutorials.

Academic Support Academic advice and support is the responsibility of the staff delivering the module. Outside of normal timetabled hours, advice and guidance on matters relating to the material being taught and on its assessment can be obtained either by arranging an appointment with academic staff or during published "surgery" hours. Appointments are most commonly arranged by email.

In addition all students are allocated Academic Personal Tutor (APT) to whom they can turn for general academic advice related to their studies. From time to time students can expect their APT to invite them to meet to discuss their progress.

As a supplement to this formal academic support, all modules at level 1 (i.e. first year modules) include timetabled Peer-Assisted Learning (PAL) sessions. These classes are extra to the sessions timetabled with academics and provide new students with a significant additional resource, over and above the normal 12 hours contact time. PAL sessions are led by trained PAL leaders; second and final year

Part 7: Student Learning

students who are able to use their experience during the first year to help the newer students overcome barriers to success in their studies.

On-line Academic Support Extensive on-line support for this programme is provided through the University portal (myUWE). This provides access to the University's e-library, which allows students to read academic journals and study-skills material. Of particular interest to students of this programme is access to the ACM, IEEE and British Standards Online databases. The portal also gives entry to UWE's Virtual Learning Environment (Blackboard) which is used by academics to make available general information about the module delivery, handbooks, lecture notes and other materials. In addition, the portal publishes individual student timetables, marks and other aspects of the operation of the programme and University life.

Pastoral Support Pastoral care is provided through the University-wide Student Advisers, a team of staff who provide comprehensive, full-time student support service. Advisers are trained to provide advice on matters commonly of concern, including regulatory and other matters; the Adviser will, when necessary, direct the student to specialist professional services including the University's counselling service, careers, financial services etc.

Independent Study
All modules require students to carry out independent study, such as preparation for classes, research for projects and completion of assignments. A full range of facilities are available at all sites to help students with these. The philosophy is accordingly to offer students both guided support and opportunities for independent study. Guided support is mainly in the form of timetabled sessions. Students are expected to attend all sessions on their timetable.

The habits and practice of independent study is then developed through 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 In 2012 the Faculty has undertaken a major new build of computing facilities in which it offers a specialised computing facility alongside the general University provisions. There are multiple computing laboratories of 20 plus seats running Windows, Linux and dual-boot systems required for this program. Computers within the specialist laboratories include the standard University build augmented by software resources and hardware equipment necessary for the delivery of the modules. For example, the specialist Forensic and Security laboratory runs virtual machine and industry-standard specialist software.

In addition, one of the most popular areas within the Faculty is the Open Access laboratory. This area is never timetabled and gives students the opportunity to access machines at all times during opening hours.

Part 8: Reference Points and Benchmarks

QAA subject benchmark statements
The QAA Subject Benchmark Statement for Computing was revised in 2007, and is applicable to this proposal. The design team has considered them in drawing up the structure of the proposed 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 (para. 2.7) recognise that HEIs are likely to offer a range of programmes in computing. In paragraph 2.8 they refer to programmes, at one extreme, which provide "*a wide range of topics spanning the entire area of computing*" providing great flexibility. At the other extreme, the benchmarks recognise that there will be programmes which "*take one very specific aspect of computing and cover it in great depth*". This proposal is in the middle of these extremes. Nevertheless it does allow students to recognise the importance of specialist areas, in particular through the choice of a Level 3 module.

The benchmarks (para. 3.1) expects students to develop a wide range of abilities and skills, divided into three broad categories:

Part 8: Reference Points and Benchmarks

1. Computing related cognitive abilities and skills relating to intellectual tasks.
2. Computing related practical tasks.
3. Transferable skills that may be developed in the context of computing but which are of general value.

This proposal extends these categories into extensively defined learning outcomes.

The benchmarks also contain (section 6) statements of the standards expected of graduates at threshold, typical and excellence levels. The team is of the view that graduates of the proposed programme will be able to meet the threshold standards and are given full opportunities to achieve excellence.

University strategies and policies
The development of this programme reflects institutional policies and is fully consistent with the University's commitment to 'make a positive difference to our students, business and society'.

Staff research projects
The thread of artificial intelligence modules in the programme has been informed and developed by a team of world-class researchers in this area. Much of the module material is based upon their actual research work.

The thread of object-oriented systems development modules in the programme has been informed and developed by members of staff who are members of the faculty's Software Engineering Research Group and active in the field of software engineering research for example in research automating business process with service oriented architectures and web services.

Employer interaction and feedback
The programme benefits from close collaboration with local companies. These liaisons influence the curriculum and also provide professional mentors, placement opportunities and guest speakers.

FOR OFFICE USE ONLY

First CAP Approval Date	May 2013			
Revision CAP Approval Date		Version	1	
			1.1	
			1.2	
			2	
			3	
			4	
	6 November 2017		5	Link to RIA (ID 4506)
Next Periodic Curriculum Review due date	September 2018			
Date of last Periodic Curriculum Review				