Programme Specification

<table>
<thead>
<tr>
<th>Programme award and title:</th>
<th>Bachelor of Science with Honours in Computing Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCAS code:</td>
<td>G400</td>
</tr>
<tr>
<td>SCQF Qualification Level:</td>
<td>10</td>
</tr>
<tr>
<td>SCQF Credit Value:</td>
<td>484</td>
</tr>
</tbody>
</table>

**Educational aims of the programme:**
Concise (e.g. a few sentences), general statement of aims and broad purposes of the programme

The programme aims to produce graduates who will:
- be able to demonstrate a detailed understanding, knowledge and experience of the theory, practice and applications of computing science,
- be equipped to apply appropriate scientific and engineering practices in the development and construction of computer-based systems for the solution of practical problems,
- be adaptable, have good communication skills and have a firm understanding of computing in order to meet the changing demands of the real world and a modern technological society,
- appreciate the need for continuing professional development in recognition of the need for lifelong learning,
- have an understanding of the professional standards and responsibilities demanded of the computer professional by industry, commerce and society.

**Intended programme learning outcomes:**
Outline (e.g. one or two paragraphs) of what the student will know, understand and be able to do as a result of their learning, expressed in the categories below. Please consider the contribution made to the student’s personal development planning (PDP) and future employability.

**Knowledge and understanding**
- the principles and practice of computer programming,
- the concepts, principles and practice of Computing, including software development, data communications, databases, networking, computer systems, software engineering, multimedia and HCI, and an ability to exercise critical judgement across these areas,
- the importance of systematic software design methodology,
- the operational principles of computer networks, and the ability to design and implement web based applications,
- the impact of legislation on their professional work.

**Subject-specific skills and other attributes**
- critically analyse concepts, principles and practice, in the context of loosely defined scenarios,
- bring effective approaches to bear in solving problems,
- evaluate and understand requirements and specifications,
- produce work involving problem specification, analysis, design and implementation of software systems,
- prepare technical documentation and prepare and give technical presentations,
- use and develop verification, validation and testing procedures for computer-based systems,
- apply good HCI design principles to computer applications, multimedia presentations, and web sites,
- apply formal methods in the development of computer-based systems,
- show effective judgement in the choice and use of tools and techniques.

**Generic skills (e.g. information skills, communication skills, critical, analytical and problem-solving abilities) and other attributes**
- work as a member of a development team, recognising the different roles within a team and different ways of organising teams,
- apply appropriate practices within a professional and ethical framework,
- organise and manage their time and prioritise workloads.
• reflect on and assess their professional development,
• communicate effectively, both orally and in writing.

**Learning, teaching and assessment strategies:**

Outline (e.g. one or two paragraphs) on overall approach taken to develop and assess learning outcomes, including any distinctive features

The teaching methods are lectures, tutorials and practical laboratory sessions. The different modules making up the degree use these in differing ways: they need to be selected appropriately for each module. This is achieved partly by the lecturer in charge of each module selecting what is in their opinion appropriate, and these decisions are reviewed regularly by the learning and teaching committee, and by the subject committee.

Assessment techniques used are class tests, practical assignments, reports/essays, examinations, and, for the final project, an interim and a final project dissertation. A random sample of all items of assessed work is reviewed by a member of staff not involved in the teaching of the unit. For the final Honours project, all pieces of assessment are marked independently by two members of staff. Communication skills are developed in many courses through coursework reports and project demonstrations and additionally in the Honours project through a poster session and oral project presentation.

Class tests are used in the first semester to ensure that these students have grasped the basic elements of computer applications. Practical assignments are used extensively particularly on units that have a major programming component. This is the only realistic method for providing formative feedback to students in this area of work. Practical work is also important in units where design is a major element, as is true in both software engineering and Multimedia and HCI. Two software engineering units contain a group project and this enables students to be assessed in a situation where they are working together. Essay type assignments are used where the nature of the material lends itself to this: for example, essays may be used where a number of different techniques can be compared.

Examinations are used as a summative assessment (though we also use the marks gained in earlier class tests, practical assignments and essays to produce final grades in each unit). The grades for all advanced units taken in semesters 5 - 8 are combined to give the final Honours classification. In a number of units, we require students to gain at least a certain level in examinations: this is because we are aware that collusion between students can lead to less able students doing better than they could unaided in practical work. The final year project is different: it lasts two semesters, and is assessed by two reports. One is an interim report, produced at the end of the 1st semester of the project, and the other is the final project dissertation, produced near the end of the 2nd semester. Students also have to demonstrate their project to the staff assessing the project. In order to gain an Honours degree, the student must pass the project at the first attempt.

As can be seen from the above, students are enabled to demonstrate achievement through their practical work, and through their performance in examination. The practical work is set in such a way that the more able students can demonstrate their abilities while the less able students are not disadvantaged. This is achieved by making it clear what is required in order to achieve different levels of grade in the practical work.

Students with special learning needs are supported by the University’s special educational needs advisor and by the Department’s disability co-ordinator. Student progress is assessed by marking their practical assignments, and essays, and these are returned to students in less than 21 days.

Both the examinations and assessments are designed to test the students’ ability to exercise critical judgement. Working as part of a team in the context of Software Engineering is assessed through the use of group projects in two Software Engineering units. Testing of the abilities to produce work involving problem specification, analysis, design and implementation of software systems occurs partly through assessment in practical work in units involving programming, and in the assessment of the two units in Software Engineering. The ability to analyse critically concepts, principles and practice, in the context of loosely defined scenarios, and to show effective judgement in the choice and use of tools and techniques is tested in an integrative way in the final project that the student undertakes.

The units in successive semesters are designed to develop the students’ computing skills in a natural way. The progression is

• Control over the basic tools of a windowing environment
• Skills in basic IT tools
• Ability to develop and to understand simple applications in languages such as Java and SQL
• Ability to design and implement computer-based solutions to well-defined problems
• Ability to analyse loosely defined problems and design and implement computer-based solutions to them.
<table>
<thead>
<tr>
<th>Professional/statutory body accreditation or recognition:</th>
<th>British Computer Society</th>
</tr>
</thead>
</table>

**Further details:**

- Entry requirements: [http://www.external.stir.ac.uk/undergrad/entry_reqs/index.php](http://www.external.stir.ac.uk/undergrad/entry_reqs/index.php)
- Programme structure: [http://www.calendar.stir.ac.uk/](http://www.calendar.stir.ac.uk/)
- Relevant Subject Benchmark statement: [http://www.qaa.ac.uk/academicinfrastructure/benchmark/default.asp](http://www.qaa.ac.uk/academicinfrastructure/benchmark/default.asp)
- Introduction/revision date: February 2007