THE UNIVERSITY OF EDINBURGH

DEGREE PROGRAMME SPECIFICATION FOR BEng (Hons) SOFTWARE ENGINEERING

[TEMPLATE AND SAMPLE FOR PREPARATION OF DEGREE PROGRAMME SPECIFICATIONS.]

[This document provides a template and guidance notes for the production of future Degree Programme Specifications (DPSs) in Informatics, and for future revisions of current ones. It has been designed to conform to both the QAA guidelines for preparing programme specifications and the University's guidelines for producing DPSs compatible with KIS and HEAR requirements. As a sample, the document contains embedded within it a DPS for the BEng in Software Engineering. Meta-comments, like this one, are italicized and in square brackets, and provide guidance and suggestions; they should be deleted for the final version of any given DPS.]

[The following introductory text explains the purpose of the DPS, and may be used as a standard preamble in all Informatics DPSs, with the name of the programme adjusted appropriately.]

This document serves as an overall description of the BSc (Honours) degree programme in Software Engineering at the University of Edinburgh. It provides information on the form, content, character and distinctiveness of the programme, and is intended for the use of current and prospective students, employers, teaching and administrative staff and external examiners. In addition, it provides a basis for Quality Assurance activities such as programme accreditation and review.

1. Awarding Institution: The University of Edinburgh

2. Teaching Institution: The University of Edinburgh

3. Programme accredited by: British Computer Society (BCS), see http://www.bcs.org/. The programme is fully accredited for Chartered IT Professional (CITP), and partially accredited for Chartered Engineer (CEng) and Chartered Scientist (CSci). The BCS is the chartered institute for IT within the UK and its accreditations are recognized throughout the IT profession. For further details, see the School of Informatics accreditation pages at http://www.inf.ed.ac.uk/student-services/quality-assurance/teaching-accreditation.
[For other degree programmes, check the accreditation pages mentioned above, and adjust the text appropriately.]

4. Final award: BEng Honours

5. Programme title: BEng (Honours) Software Engineering

6. UCAS Code: G600
[The codes for other degree programmes can be found by going to http://www.ucas.ac.uk/students/coursesearch/ and searching for computer science degrees at the University of Edinburgh.]

Relevant QAA Subject Benchmarking Group(s): Computing, see http://www.qaa.ac.uk/Publications/InformationAndGuidance/Pages/Subject-benchmark-statement-Computing.aspx.
[For our single degrees, 'Computing' is the only relevant group. For joint degrees, browse the list on the QAA website above.]

7. Postholder with overall responsibility for QA: John Longley

8. Date of production/ revision: August 2012 *[Don't forget to update this!]*

9. External Summary

[The following text should be rewritten or adapted for each degree programme. It should be helpful to a prospective student who is trying to choose a degree: it should explain what the subject is, and what is distinctive about the Edinburgh experience in this degree. Try to make it sound exciting!]

[Further examples of external summaries may be found in other DPSs available from http://www.inf.ed.ac.uk/student-services/teaching-organisation/taught-course-information/degree-programmes (at least those revised from August 2012 onwards). You may also be able to glean useful text from our undergraduate prospectus at http://media.inf.ed.ac.uk/ugbrochure/2011%20version.pdf . For joint degree programmes, check the other school's online documentation to see what they say about their subject area and their approach to it.]

Software Engineering is concerned with the systematic, principled development and evolution of software systems. Software engineers use theoretical and practical ideas from informatics, engineering and the social sciences to ensure that software is suitable for use in its domain. The academic discipline of software engineering studies the current practice of software engineering and works towards improving it.

At Edinburgh, Software Engineering is seen as part of the wider discipline of Informatics (the study of both natural and artificial information processing systems), which also covers Computer Science, Artificial Intelligence and Cognitive Science. Students gain a solid foundation in Software Engineering, including design, programming, testing, architecture and process improvement. They can also studying areas of Computer Science such as operating systems and computer networking, and can explore new and emerging connections with disciplines such as molecular biology, neuroscience and linguistics. As a large department, we are able to offer an exceptionally wide range of courses across all areas of informatics. In addition, the content of our courses is continuously informed and enriched by the latest cutting-edge research in which we have played a leading role for a long time

10. Educational aims of programme:

[Again this should be adapted for each degree programme. The purpose here is to explain what, broadly, the programme is trying to achieve, and what general ***values*** *inform the programme. Again, it should be helpful to a student trying to choose a degree: e.g. why might one choose this degree rather than any of our other programmes? For joint degree programmes, one should again consult the other school's existing documentation, and should seek agreement with their Director of Teaching regarding the educational aims of the joint programme. (Not all the aims of a single degree in Engineering need be aims for a joint degree in CS and Engineering.)]*

The programme aims to give students the opportunity to acquire knowledge, understanding and practical skills in topics spanning Software Engineering and the entire spectrum of Computer Science. It is not simply a vocational training course for programmers; rather, our aim is to develop well-rounded professionals with strong conceptual foundations and wide practical experience, fostering both critical and creative thinking — professionals who can become future leaders, with a sense of the responsibility and ethics associated with this role. The programme places a particular emphasis on the interplay of theory and practice: students will not only acquire proficiency in the use of current computing technologies — which may change rapidly from year to year — but also an appreciation of the underlying theoretical concepts which can be expected to remain applicable to computing systems for a long time. Communication skills, initiative, professionalism and the ability to work with others are also developed as integral parts of the learning process.

11. Programme outcomes.

[Many of the bullet points in this section will apply equally to all of our degrees, though you should check whether they are appropriate to the programme in question and add/subtract points if necessary. For joint degrees, this may involve some level of consultation with the other school to take account of their outcomes, as in point 10.]

Students taking the degree programme will acquire a broad range of knowledge and skills on several levels.

11a. Knowledge and understanding

Students successfully completing the programme should:

- have a knowledge and understanding of how computers work, from hardware up through system software to application programs;
- understand the processes by which high quality software is developed;
- understand the concepts of *modelling* and *abstraction* and their importance to the design of computer systems (e.g. architectures, data structures, control flows);
- understand the underlying mathematical concepts which allow computer scientists to reason about computers and computer based systems, including the nature of algorithms and their computational complexity;
- understand the landscape of current research in Informatics and its relationship to other disciplines;
- understand the social and professional context in which computer systems are developed and used, including the typical requirements and expectations of users;
- are aware of the ethical and legal issues involved in the use of computing systems.

11b. Graduate attributes: Skills and abilities in Research and Enquiry

Graduates from the programme will be able to:

- understand and develop formal models of computer systems;
- explore alternative approaches to a given problem, and integrate different approaches;
- quickly assimilate existing work of relevance to a given problem.

11c. Graduate attributes: Skills and abilities in Personal and Intellectual Autonomy

Graduates will have proven ability to:

- plan complex system development and validation tasks by themselves;
- deploy their logical, analytical and problem solving skills to find solutions to technical problems;
- show initiative in taking on industrial and academic IT roles;
- critically evaluate their own and others' work and ideas.

11d. Graduate attributes: Skills and abilities in Communication

Graduates will have proven ability to:

- communicate effectively through a variety of oral, visual, written, diagrammatic and on-line media;
- communicate technical content to a range of different audiences;
- work effectively as part of a development team.

11e. Graduate attributes: Skills and abilities in Personal Effectiveness

Graduates will have proven ability to:

- make effective use of learning materials, and acquire and apply knowledge from a variety of sources;
- work effectively on large projects, both individually and as part of a team;
- organize their workload and manage their time when working independently, and complete complex tasks under deadline pressure.

11f. Technical/ practical skills

Graduates will have proven ability to:

- develop and implement computer systems, including programs in a range of programming languages (e.g. Java, Haskell, Python);
- understand and apply a range of modern programming paradigms (e.g. procedural; functional; object-oriented);
- make effective use of a wide range of state-of-the-art technologies (e.g. in relation to databases, software engineering or the World Wide Web);
- master new programming languages and technologies quickly as the need arises;
- make well-informed and sensible decisions when designing computer systems;
- design and conduct experiments and evaluate their results;
- operate computing equipment and software systems effectively.

12. Programme structure and features

[This should be reworked for each programme, although the first and second year requirements will be the same for all. For course structure, the content should be abstracted from the Degree Programme Table referenced below; however, do not be too specific as regards names of particular courses and numbers of points, as these may change from year to year. Entry requirement details may be found at http://www.ed.ac.uk/studying/undergraduate/degrees?id=0,9&cw_xml=related.php .]

Course structure: The programme takes four years. In each year, students must gain 120 credit points as follows.

- First year: All students take a set of required Informatics courses (e.g., Computation and Logic; Functional Programming; Object-Oriented Programming; Data and Analysis), and supporting Mathematics courses. They are also required to choose further courses at Levels 7 or 8, which may be in an unrelated subject.
- Second year: All students take a set of required Informatics courses (e.g., Processing Formal and Natural Languages; Algorithms, Data Structures, Learning; Introduction to Computer Systems; Introduction to Software Engineering), and supporting Mathematics courses. They are also required to choose other courses at Levels 7 or 8, which may be in an unrelated subject.
- Third year: Students take some compulsory Informatics courses: the System Design Project (in which students gain practical experience of building a large-scale system as members of a team), a Large Practical (undertaken individually), and a course in Professional Issues. In addition, they study two 10-point courses on core Software Engineering topics and select five 10-point courses from within Informatics and one 10-point course from within or outside Informatics.
- Fourth year: All students undertake an individual Honours Project worth 40 points; this typically involves building a substantial software system over the course of the year supervised individually by a member of staff, and may involve elements of research or original thinking. In addition, students select one course from a specified list of 10-point courses especially relevant to Software Engineering; six other 10-point courses from within Informatics; and one 10-point course from within or outside Informatics.

For further details of compulsory and optional course choices, consult the *Degree Programme Table* at http://www.drps.ed.ac.uk/index.php. The list of Informatics courses at http://www.inf.ed.ac.uk/student-services/teaching-organisation shows which optional courses are suitable for this degree programme.

Entry requirements: For admission to degree programmes in the School of Informatics, students must standardly achieve one of the following combinations.

- SQA Highers: AAAA-ABBB or more if in two sittings, to include Mathematics. Sixth-year work in Mathematics is recommended.
- GCE A Levels: AAA-ABB required in one sitting, to include Mathematics; or AAA-AAB including one of Biology, Chemistry or Physics plus AS level Mathematics at Grade A.

- IB: 32 points overall and award of IB Diploma to include Mathematics HL at Grade 5 plus two HL subjects at Grade 5.

All applicants must also meet our general university entry requirements, see http://www.ed.ac.uk/studying/undergraduate/applications-admissions/entry-requirements, including English language requirements. Note that admissions are highly competitive and attainment of the above minimum requirements does not in itself guarantee the offer of a place.

For direct entry into second year, the above requirements must be exceeded as follows:

- SQA Advanced Highers: AA to include Mathematics, plus Highers at AB in two other subjects. Appropriate relevant computing qualifications or experience is also required.
- GCE A Levels: A*AA in one sitting, to include Mathematics. Appropriate relevant computing qualifications or experience is also required.
- IB: 38 points overall and award to IB Diploma to include Mathematics Higher Level at Grade 6 plus two Higher Level subjects at Grade 6. The Diploma must include Computer Science.

Details of other admission pathways, for instance for overseas and mature students, can be found on the College of Science and Engineering applications page.

Progression requirements: To obtain an Honours degree, students must qualify in their second year for entry to third and fourth year as Honours years. This normally requires passes at 50% or above at the first attempt in at least 40 points' worth of the second-year Informatics courses, together with passes at 40% or above in all remaining compulsory courses. (Permission to proceed to Honours may sometimes be granted by the Head of School when not all these requirements are met.)

Students who are not admitted to Honours at this stage but achieve a pass at 40% or above in first and second year courses normally complete their studies within one more year, aiming at an ordinary (non-Honours) degree.

Exit awards: The final Honours degree classification of the programme is based equally on performance in third and fourth years. Degrees are classified according to the University's standard marking scale with boundaries at 70%, 60%, 50% and 40%. Students can also be awarded an ordinary degree on the basis of their third year marks.

Mode of study: Full time

Language of study: English

13. Teaching and Learning Methods and Strategies

[Points 13 and 14 will probably be mostly the same for all programmes, but check them anyway.]

The material for most Informatics courses is taught through *lectures* (typically around 16 lectures for a 10-point course, and around 32 lectures for a 20-point course). In most cases supporting materials, including notes, slides and sometimes video recordings of the lectures themselves, are made available to students on the web. Lecturers also direct students to *recommended reading* to supplement the lecture material.

For all Informatics courses in the first two years (and some in later years), lectures are supported by weekly scheduled *tutorials*, in which students in groups of 10–15 work through set tutorial exercises with the help of a tutor, and have the opportunity to seek assistance with the course material where required. Many courses in the first two years are also supported by scheduled *laboratory sessions* or supervised drop-in laboratory time, in which they are able to seek help with the practical (e.g. programming) aspects of the course material.

In the third and fourth years, students also learn through undertaking major projects and practicals as detailed in item 12 above. In the case of the fourth year Honours Project, they are guided individually by a member of staff who directs them to the appropriate learning materials and sources.

In addition, *InfBase*, a student help desk, offers further learning support for Informatics students outside scheduled tutorial times. Individual advice on general matters relating to the degree programme (including course choices) can be obtained from the student's *Personal Tutor*, and a range of other student support services is available.

14. Assessment Methods and Strategies

For most courses, the student's achievement is assessed by academic staff via a combination of examinations and coursework assignments. (The balance is typically around 75% for the examination and 25% for coursework, with some variation between courses.) Depending on the course, examinations may be written or online, and assessed assignments may be pen-and-paper or practical programming exercises. Tutorial exercises are not usually assessed directly, but makes an important contribution to preparing students for examinations.

In the final year, the Honours Project, which carries a third of the credit for the year, is assessed via a report on the project work written by the student. This is assessed independently by two members of academic staff, typically in the light of a live demonstration of the project work given by the student. The markers then confer to agree on the final mark for the project.

15. Career Opportunities

[ITO staff (currently Gillian Bell and Effie Dickson) will be able to help with this item. In particular, for each degree programme, they have information on where our graduates were employed six months after graduating.]

Computers are now ubiquitous in modern life. The most interesting opportunities in the future are open to those who really know about computing, software and information systems. Our graduates can choose from a wide range of opportunities in industry, commerce, government and academia; the majority of Informatics graduates enter employment relating to their degree, while others decide to continue within academia to pursue their research interests. For example, graduates from 2010/11 entered employment as programmers or software engineers in companies such as Logica,Roboreus, FDM Group, AV Distribution, and IT University of Copenhagen.

16. Other Items

This Degree Programme Specification conforms to the QAA Guidelines for preparing programme specifications, see
http://www.qaa.ac.uk/Publications/InformationAndGuidance/Documents/guidelines06.pdf.