UNIVERSITY OF LEICESTER

Programme Specification (Undergraduate)

Date amended: 25 Feb 2016

1. Programme Title(s) and UCAS code(s):

BSc Computer Science (G400);
BSc Computer Science with a Year Abroad (G401);
BSc Computer Science with a Year in Industry (G402)
BSc Computer Science (with Foundation Year)
(G499)

2. Awarding body or institution:

University of Leicester

3. a) Mode of study:

Full-time

b) Type of study:

Campus-based

4. Registration periods:

The normal period of registration is three years (four years for the "Year Abroad" and "Year in Industry" variants).

The maximum period of registration is five years (six years for the "Year Abroad" and "Year in Industry" variants).

For Foundation Year Variant:
The normal period of registration is four years (one year for the Foundation Year, with three years for the BSc) The maximum period of registration is six years (one year for the Foundation Year, and five years for the BSc)

5. Typical entry requirements:

320 points normally including ABB from 6 or 12 unit awards.

For Foundation Year Variant:

A level: ABB or points equivalent from best three A levels. Typically in subjects outside of the 'usual' A levels expected by the department.

BTEC Diploma: DDM in appropriate subject area.

Access to HE courses in Science and Engineering: 45 L3 credits, including 30 at Distinction and remaining L3 credits at least at Merit.

6. Accreditation of Prior Learning:

APL will not be accepted for exemptions from individual modules, however may be considered for direct entry to year 2, on a case by case and subject to the general provisions of the University APL policy.

For Foundation Year Variant:

n/a

7. Programme aims:

The programme aims to:

- Provide students with an education and training in computer science that includes both fundamental concepts and state-of-the-art trends, and also provides a good indication of the breadth of the subject.
- To provide opportunities for students to learn a wide range of skills in the analysis, design, specification, implementation, testing and documentation of computer software systems.
- To develop powers of critical analysis, skills in problem solving, written communication, and abilities in presentation.
- Provide students with experience of both team-based and individual project work.
- To develop skills that will enhance employment prospects, especially in the IT industry or other numerate disciplines.
- Allow students to gain familiarity with a variety of modern programming languages, and the underlying principles of programming paradigms (functional, object oriented, logical and so on).
- To develop scientific problem solving abilities, along with an appreciation for mathematical and scientific methods, which will provide a lifelong support for your career.
- To develop an appreciation of the necessity for rigorous subject foundations, and the need for logical arguments, which will also contribute to your lifelong skills.
- Ensure students will have expertise and understanding at a level where they can embark upon a high quality taught Masters programme in computer science.

In addition to these aims, G401 BSc Computer Science with a Year Abroad aims to:

- Enable students to experience modern Computer Science from an international perspective.
- Develop students' working knowledge of a language other than English.
- Provide students with an environment that will encourage a thoughtful and mature approach to all aspects of study and life, creating graduates with broad experiences and horizons.

In addition to these aims, G402 BSc Computer Science with a Year in Industry aims to:

- Help place students on relevant industrial placements where they can gain first-hand experience of the requirements and opportunities of the computing industry in the UK.
- Enable students to use and develop the knowledge and skills gained during the first two years of the degree programme.

For Foundation Year variant, see Foundation Year Programme Specification

8. Reference points used to inform the programme specification:

- Accreditation reports from the BCS.
- QAA Frameworks for Higher Education Qualifications in England Wales and Northern Ireland
- QAA Benchmarking for Computing 2007.
- QAA Developmental Engagement.
- QAA Institutional Report.
- University Learning Strategy.
- University Employability Strategy
- PDR report (January 2010).
- First Destination Survey.
- Student Feedback 2014.
- External Examiners' reports.

9. Programme Outcomes:

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
(a) Discipline specific knowledge and competencies		
(i) Mastery of an appropriate body of knowledge		
1. Demonstrate both recollection and understanding of computing factual knowledge, together with relevant scientific knowledge and concepts from logic.	Lectures, surgeries, computer laboratories and problem classes. Also background reading and research.	Written examinations, assessed coursework, group and individual project presentations, individual project oral examinations and project reports.
2. Demonstrate recollection and understanding of engineering principles, scientific principles and mathematical and logical theories in computing.	As above.	As above.
3. Demonstrate appropriate management techniques.	Group work in modules, group project work.	Assessed coursework and
4. Demonstrate mastery of the core of an appropriate foreign language (G401)	Lectures, language laboratories and learning abroad.	project work. University
5. Demonstrate understanding of the core elements of industrial practice and organization (G402).	Work placement.	report.
		Placement Report.

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
(ii) Understanding and application of key concepts and techniques		
1. Demonstrate knowledge and understanding of both computing at a mathematically and logically abstract (conceptual) level, and to apply this to the design and modeling of computing systems.	Lectures, surgeries, computer laboratories and problem classes, worksheets, project work.	Written examinations, assessed coursework, group and individual project presentations, individual project reports.
2. Demonstrate and apply the concept of logical structure and organization that pervade computing, and be able to generalize and specialize to achieve such structure. Apply these ideas in modeling and design.	As above.	As above.
3. Understand and apply the theoretical principles, practices and tools of both mathematics/logic/science, and also software engineering, together with suitable processes and methodologies, to determine strategies for solution; and create requirements, specifications and designs.	As above, with emphasis on all forms of project work.	As above, with emphasis on project assessments.
4. Design and construct, test & verify, and deliver medium scale software systems. Maintain systems. Make appropriate use of tools for such tasks.	Computer laboratories and project work.	Assessed laboratory work, group and individual project presentations, individual project reports
5. Undertake mathematical/scientific problem solving and practical engineering style problem solving; utilize general scientific knowledge and principles where appropriate.	Lectures, surgeries, computer laboratories and problem classes, worksheets, project work.	Written examinations, assessed coursework, group and individual project presentations, individual project reports.
6. Demonstrate ability to communicate some aspects of Computer Science in a foreign language. (G401)	Lectures and language laboratories.	University report.
7. Work as a computer scientist and computing engineer in an industrial or commercial setting. (G402)		

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
(iii) Critical analysis of key issues		
1. Analyze customer problems, requirements and criteria, and hence plan and select an appropriate solution strategy.	Lectures, surgeries, problem classes, worksheets, group and individual project work.	Written examinations, assessed coursework, group and individual project presentations, group reports and individual project reports.
2. Understand customer needs, and budgets, undertaking suitable research. Ensure software solutions are fit-for- purpose. Be able to manage the complete engineering process and evaluate the end product, and to work with associated uncertainties.	As above.	As above.
3. Be able to recognize risks in the deployment and use of software systems.		
(iv) Clear and concise presentation of material		
1. Write short accounts of computing and scientific knowledge.	Worksheets, group and individual project work, private study.	Written examinations, assessed coursework., group and individual project presentations, and project reports.
2. Produce written and visual information in a variety of forms, chosen to maximize reader/audience impact and understanding.	As above.	As above
(v) Critical appraisal of evidence with appropriate insight		
1. Evaluate and appraise software systems, in terms of attributes and tradeoffs. Identify risks and safety concerns.	Lectures, surgeries, computer laboratories and problem classes. Also background reading and research.	Written examination, assessed coursework, group and individual project presentations, individual project oral examinations and reports.
2. Perform software testing, and critically evaluate and analyze test results. Evaluate whether a system meets the requirements, for future and for current use.	Computer laboratories and project work.	Assessed laboratory work, group and individual project presentations, individual project reports
3. Use relevant knowledge to appraise the commercial use and economic and long- term viability of computer systems.	As above.	As above.

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
(vi) Other discipline specific competencies		
1. Demonstrate knowledge and understanding of social, legal and ethical issues as required by computing professionals. Adopt and implement suitable professional and legal practice.	Lectures, surgeries, problem classes, worksheets, group and individual project work.	Written examination, assessed coursework, group project presentations and coursework.
2. Demonstrate knowledge and understanding of functional programming and logical principles of induction and type inference.	Lectures, surgeries, computer laboratories and worksheets.	Written examination, assessed coursework.
3. Solve problems using principles from propositional and predicate logic, demonstrating a clear understanding of the scientific method of proving an hypothesis.	As above.	As above.
(b) Transferable skills		
(i) Oral communication		
1. Respond to technical questions with accurate and concise answers.	Lectures and surgeries. Project supervision. Problem classes.	Group and individual project presentations, individual project oral examinations.
2. Demonstrate fluent and sustained scientific, technical and business communication, supported by a variety of audio-visual aids.	Lectures and project supervision. Use of student learning center.	As above.
3. Demonstrate core oral communication skills in a foreign language (G401).	Lectures and language laboratories.	University Report.
(ii) Written communication		
1. Demonstrate ability to write concise and accurate summaries of computing and scientific knowledge, and solutions to problems, in a variety of different formats.	Lectures, surgeries, computer laboratories and problem classes, worksheets, project work.	Written examinations, assessed coursework.
2. Produce properly structured, clear, advanced technical reports or dissertations.	Lectures and surgeries. Discussed in both group and individual project supervisions.	Group project assessed coursework and individual project reports.
3. Demonstrate core written communication skills in a foreign language (G401).	Lectures, tutorials, language laboratory work.	University report.
(iii) Information technology		
1. Use a very broad range of software and IT tools, and to choose these appropriately for retrieval and management of information.	Lectures, surgeries and laboratories.	Assessed (laboratory) coursework.
2. Demonstrate a broad and deep understanding of many IT tools, and be able to adapt to new programming paradigms in the future.	As above.	As above.

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
(iv) Numeracy		
1. Demonstrate understanding of the concept of number.	Lectures, surgeries, computer laboratories and problem classes, worksheets.	Written examinations, assessed coursework.
2. Use analytical, quantitative, and graphical methods, and deploy elementary statistics.	As above, together with project work.	As above, along with group and individual project presentations and reports.
(v) Team working		
1. Work effectively as part of a team, and demonstrate ability to organize roles and manage time, undertake assigned tasks, and ensure final completion of a team project. Identify strengths and weaknesses of team members.	Lectures and project supervision. Use of student learning center.	Group project assessed coursework and presentations. Mini projects.
(vi) Problem solving		
1. Solve a variety of short problems through the integration of knowledge of mathematics, logic, algorithms and basic computing.	Lectures, surgeries and problem classes. Also covered in project supervisions.	Written examinations, assessed coursework, and project reports.
2. Use systematic analysis and design methods, and appropriate algorithms, to solve medium scale problems.	As above.	As above.
3. Analyze large-scale problems to produce suitable solutions with sensible economic and commercial compromises. Apply management techniques to allocate resources to projects.	As above.	Group and individual project presentations and reports.
(vii) Information handling		
1. Conduct significant background research and literature surveys, and summarize content from information sources.	Taught in lectures. Also covered in project supervisions.	Individual project reports.
2. Demonstrate a broad understanding of problems and issues that arise in the location, organization, processing and evaluation of data.	As above.	Written examinations, assessed coursework, and project reports.
3. Recognize the need for information, and work with fuzzy, limited and possibly contradictory information.	As above.	As above.
1. Demonstrate knowledge and understanding of professional and ethical issues, and aspects of the law, in the context of Computing Professionals.	Lectures, surgeries and problem classes. Also covered in project supervisions.	Written examinations, assessed coursework, and project reports.
2. Demonstrate independence and time management skills.	Project supervisions and research project work. Meeting coursework deadlines.	Project reports.
3. Design a personal work plan and be able to improve performance with a clear view of long-term professional development.	Project supervisions and research project work.	As above.

10. Progression points:
This programme follows the standard scheme of award and classification set out in Senate Regulation 5 modified as follows:

For Foundation Year Variant:
Progression from Foundation Year to year 1: In cases where a student has failed to meet a requirement to progress he or she will be required to withdraw from the course.

Students will be required to pass Foundation Year in order to progress to Year 1 with an average module mark of at least 60%. Students are required to have a mark of at least 60% in FS0031 and FS0032 to progress onto the BSc Computer Science.

British Computer Society Accreditation requires that individual projects be passed at the first attempt.

11. Scheme of Assessment

The programme follows the standard Scheme of award and classification set out in Senate Regulation 5.

If regulation 5.14(c) applies in relation to any of the modules CO1003, CO1005, CO1019 then failed marks must be no lower than 35% (rather than the normal 30%) in order for students to proceed and re-sit.

Regulation 5.10 applies absolutely to CO2015 Software Engineering Project.

12. Special features:

Emphasis on blending long-term foundational knowledge with state-of-the-art technologies and current programming languages; Group Projects involving an external client wherever possible;; Individual Projects with a number of structured milestones.

13. Indications of programme quality

British Computer Society Accreditation requires that individual projects be passed at the first attempt.

14. External Examiner

The details of the External Examiner(s) for this programme and the most recent External Examiners' reports can be found here.

Appendix 1: Programme structure (programme regulations) overleaf

Appendix 2: Module specifications

See module specification database http://www.le.ac.uk/sas/courses/documentation

Appendix 3: Skills matrix

See skills matrix

Appendix 4: Foundation Year Programme Specification

BSc COMPUTER SCIENCE

FIRST YEAR MODULES

SEMESTER 1

Core Modules			**Credits**
CO1003	PROGRAM DESIGN		20
CO1008	REQUIREMENTS ENGINEERING AND PROFESSIONAL PRACTICE		10
CO1012	DISCRETE STRUCTURES		10
CO1016	COMPUTER SYSTEMS		20
		Semester Total	**60**

SEMESTER 2

Core Modules			**Credits**
CO1001	LOGIC AND PROBLEM SOLVING		20
CO1005	DATA STRUCTURES AND DEVELOPMENT ENVIRONMENTS		20
CO1019	DATABASES AND WEB APPLICATIONS		20
		Semester Total	**60**

SECOND YEAR MODULES

SEMESTER 1

Core Modules			**Credits**
CO2001	USER INTERFACES AND HCI		10
CO2006	SOFTWARE ENGINEERING AND SYSTEM DEVELOPMENT		20
CO2012	SOFTWARE PROJECT MANAGEMENT AND PROFESSIONALISM		10
CO2011	AUTOMATA, LANGUAGES AND COMPUTATION		20
		Semester Total	**60**

SEMESTER 2

Core Modules			**Credits**
CO2008	FUNCTIONAL PROGRAMMING		10
CO2015	SOFTWARE ENGINEERING PROJECT		20
CO2016	MULTIMEDIA AND COMPUTER GRAPHICS		10
CO2017	OPERATING SYSTEMS, NETWORKS AND DISTRIBUTED SYSTEMS		20
		Semester Total	**60**

THIRD YEAR MODULES

SEMESTER 1

Core Modules			**Credits**
CO3015	COMPUTER SCIENCE PROJECT (PART 1)*		20
Optional Modules			
	40 credits of options selected from:		
CO3007	COMMUNICATION AND CONCURRENCY		20
CO3091	COMPUTATIONAL INTELLIGENCE AND SOFTWARE ENGINEERING		20
CO3095	SOFTWARE MEASUREMENT AND QUALITY ASSURANCE		20
CO3015	C++		20
CO3098	WEB TECHNOLOGIES		20
		Semester Total	**60**

SEMESTER 2

Core Modules			**Credits**
CO3015	COMPUTER SCIENCE PROJECT (PART 2)*		20
Optional Modules			
	40 credits of options selected from:		
CO3002	ANALYSIS AND DESIGN OF ALGORITHMS		20
CO3090	DISTRIBUTED SYSTEMS AND APPLICATIONS		20
CO3093	BIG DATA AND PREDICTIVE ANALYTICS		20

CO3096	COMPRESSION METHODS FOR MULTIMEDIA	20
CO3099	CRYPTOGRAPHY AND INTERNET SECURITY	20
	Semester Total	**60**

BSc COMPUTER SCIENCE WITH A YEAR ABROAD

First and Second Year Modules

As for the first- and second-year of the BSc degree in Computer Science.

Third Year Modules

The third year will be spent abroad taking approved courses either in an institution associated with the Computer Science Department via an ERASMUS bilateral agreement or in a university that has a Study Abroad exchange partnership agreement with the University of Leicester. Students will normally be required to complete the year and to reach a pass level of attainment in 60 credits of Computer Science modules Failure to do so will result in the student reverting to the three year BSc Computer Science degree. The marks awarded during the year abroad do not contribute to the final degree classification.

Note: Transfer will be confirmed only after successful completion of the first year.

Fourth Year Modules

As for the third-year of the BSc degree in Computer Science.

BSc COMPUTER SCIENCE WITH A YEAR IN INDUSTRY

First and Second Year Modules

As for the first- and second-year of the BSc degree in Computer Science.

Third Year Modules

1. Students will work within a sponsoring company for one year between 1 July of the second year of the course and the start of the following year.
2. During their one-year placement students will undertake a programme of training and work experience which will be agreed by the sponsoring company and the University.
3. Students will be expected to keep a logbook recording their training and experience that is to be presented for approval to the sponsoring company and the University.
4. Students will be issued with a *Certificate of Industrial Studies* indicating successful completion of their placement. Students who do not satisfactorily complete their industrial placement will be transferred to the B.Sc. Computer Science degree.

   The Year in Industry does not contribute to the final degree classification.

Fourth Year Modules

As for the third-year of the BSc degree in Computer Science.