# Programme Specification
# (Undergraduate)

**For students entering in 2020/21**

**Date amended 08/03/2023**

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

MComp Computer Science (G410);

MComp Computer Science with a Year Abroad (G410 – internal programme transfer)

MComp Computer Science with a Year in Industry (G410 – internal programme transfer)

**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 four years (five years for the "Year Abroad" and "Year in Industry" variants).

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

*For Foundation Year Variant:*

The normal period of registration is five years (one year for the Foundation Year, with four years for the MComp) The maximum period of registration is seven years (one year for the Foundation Year, and six years for the MComp)

**5. Typical entry requirements:**

A level: AAB or points equivalent from best three A levels. Computer Science or Mathematics preferred but not essential.

BTEC Diploma: D*D*D* in appropriate subject area, plus a pass in a Departmental UCAS day test.

*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: D*D*D in appropriate subject area.

**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 basis 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 advanced state-of-the-art education in Computer Science that includes both theory and foundations (pure Computer Science), and practical applications (applied Computer Science).
- Provide opportunities for students to learn a wide range of skills in the analysis, specification, design, implementation, testing, maintenance and documentation of computer software systems and to execute these at an advanced level.
- Enable students to become proficient in a variety of modern programming languages, and the underlying principles of programming paradigms (concurrent, imperative, functional, logical, mobile, object oriented and so on).
- Enable students to explain core subjects such as advanced algorithms, computer architecture, operating systems and networks, foundations of computation, databases, web & mobile computing, together with a further range of advanced master's level subjects that reflect the research expertise of the Department.
- *Skills labelled ^ are taught to a high level of insight and complexity:* Enable students to develop skills such as Communication, Teamwork^, Leadership & Supervision, Researching & Analysing^, Problem Solving & Decision Making^, Planning & Organization^; Learning, Improving & Achieving; Resilience, Adaptability & Drive; and Digital Skills^. Skills labelled ^ are taught to a high level of insight and complexity.
- Provide students with experience of both team-based and individual project work.
- Develop an appreciation for computational, mathematical and scientific thinking, along with an appreciation of the necessity for rigorous subject foundations, and the need for mathematical and logical arguments, which will provide a lifelong support for careers.
- Foster students' abilities to learn new and advanced subjects, and conduct research and scholarship, at levels and speeds that are more demanding than those typical of bachelor level programmes.
- Ensure students will have expertise and understanding at a level where they can embark upon a high quality PhD research programme.

*In addition to these aims, MComp 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, MComp Computer Science with a Year in Industry aims to:*

- Enable students to take up industrial placements where they can gain first-hand experience of the requirements, challenges and opportunities of the computing industry in the UK.
- Enable students to use and further 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:**

- QAA Benchmarking Statement for Computing 2016.

- University of Leicester Learning Strategy 2016-2020.
- University of Leicester Periodic Developmental Review Report
- External Examiners' reports (annual).

**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. Explain and discuss both foundations and applications of Computer Science, together with concomitant scientific knowledge and concepts from logic and mathematics, at levels which are equal to typical Master's level programmes. | Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research. | Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations. |
| 2. Explain, discuss and apply advanced engineering principles, scientific principles and mathematical and logical theories in computing. | As above. | As above. |
| 3. Demonstrate mastery of the core of an appropriate foreign language (Year Abroad) | Lectures, language laboratories and learning abroad. | Assessment at host institution. |
| 4. Demonstrate understanding of the core elements of industrial practice and organization (Year in Industry). | Work placement. | Placement Report; presentation. |

| Intended Learning Outcomes | Teaching and Learning Methods | How Demonstrated? |
|---|---|---|
| **(ii) Understanding and application of key concepts and techniques** | | |
| 1. Apply advanced knowledge of Mathematics, Logic and Computer Science to solve individual problems, both seen and unseen. | Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research. | Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations. |
| 2. Apply the concepts and techniques of abstraction, reification, logical structure and modelling that pervade Computer Science and Software Engineering to specify, design, implement and test complex small to medium size computer systems. | As above. | As above. |
| 3. Explain and apply the theoretical principles, and practical tools of Mathematics, Logic, Computer Science, and Software Engineering, together with suitable processes and methodologies, to determine strategies for innovative solutions of large scale and sophisticated problems. | As above, with emphasis on all forms of project work. | As above, with emphasis on project assessments. |
| 6. Demonstrate ability to communicate some aspects of Computer Science in a foreign language. (Year Abroad) | Lectures and language instruction. | As above |
| 7. Work as a computer scientist and computing engineer in an industrial or commercial setting. (Year in industry) | | University report. |

| Intended Learning Outcomes | Teaching and Learning Methods | How Demonstrated? |
|---|---|---|
| **(iii) Critical analysis of key issues** | | |
| 1. Analyse ambiguous client/customer problems, requirements and criteria, and hence plan an appropriate yet innovative solution strategy.<br><br>2. Explain and analyse the constraints of budgets, data, time, staffing and resources in the practical computing domain, undertaking suitable research. Ensure software solutions are fit-for-purpose. Manage the complete engineering process and evaluate the end product, and to work with associated uncertainties.<br><br>3. Be able to recognise risks in the deployment and use of software systems | Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research.<br><br>As above. | Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations.<br><br>As above. |
| **(iv) Clear and concise presentation of material** | | |
| Present information in a variety of forms, chosen to maximise reader/audience impact and understanding, such as reports, dissertations, seminars, posters, blogs, podcasts, videos and other current media technologies. Deliver a departmental research level seminar. | Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research. | Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations. |

| (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.<br><br>2. Perform software testing, and critically evaluate and analyse test results. Evaluate whether a system meets requirements, for future and for current use.<br><br>3. Use relevant knowledge to appraise the commercial use and economic and long- term viability of computer systems. | Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research. | Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations. |
| **Intended Learning Outcomes** | **Teaching and Learning Methods** | **How Demonstrated?** |
| **(vi) Other discipline specific competencies** | | |
| 1. Explain and discuss social, legal and ethical issues as required by computing professionals. Adopt and implement suitable professional and legal practice.<br><br>2. Explain and react to the rapidity of change in Computer Science. Formulate innovative and creative ideas for future advances.<br><br>3. Collect, work with and analyze all forms of data. | Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research. | Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations. |

| *(b) Transferable skills* | | |
|---|---|---|
| **(i) Oral communication** | | |
| 1. Respond to technical questions with accurate and concise answers. | Lectures and tutorials. Project supervisions. | Group and individual project presentations, individual project oral examinations. |
| 2. Demonstrate fluent and sustained scientific, technical and business communication. | As above. | As above. |
| 3. Demonstrate core oral communication skills in a foreign language (Year Abroad). | Language tuition. | Host University assessment. |
| **(ii) Written communication** | | |
| 1. Write concise and accurate summaries of computing and scientific knowledge, and solutions to problems, in a variety of different formats. | Lectures, tutorials, computer laboratories, project work. | Written examinations, assessed coursework. |
| 2. Produce properly structured, clear, advanced technical reports or dissertations. | Lectures and tutorials. 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 (Year Abroad). | 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 uses throughout Computer Science. | Lectures, tutorials and laboratories. | Assessed (laboratory) coursework. |
| 2. Adapt to future programming languages and paradigms, and all varieties of software tools and technology. | As above. | As above. |
| **Intended Learning Outcomes** | **Teaching and Learning Methods** | **How Demonstrated?** |
| **(iv) Numeracy** | | |
| 1. Demonstrate understanding of the concept of number. Solve numerical problems. | Lectures, tutorials, computer laboratories. | 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, organise roles and manage time, undertake assigned tasks, and ensure final completion of a team project. Identify strengths and weaknesses of team members. | Lectures, tutorials and project supervision. | Group project assessed coursework and presentations. Mini projects. |

| (vi) Problem solving | | |
|---|---|---|
| 1. Solve a variety of problems (both simple and advanced) through the integration of knowledge of mathematics, logic, and Computer Science. | Lectures and tutorials. 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 complex 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. |

| (viii) Skills for lifelong learning | | |
|---|---|---|
| 1. Demonstrate knowledge and understanding of professional and ethical issues, and aspects of the law, in the context of Computing Professionals. | Lectures and tutorials. 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:

Regulation 5.10 applies absolutely to CO2201 Software Engineering Project and CO2103 Software Architecture and System Design and CO4210 Personal and Group Skills.

*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 MComp Computer Science.

*For the with industry variant:*

- Students should normally pass the first year at the first attempt; and
- should normally pass the second year at first sitting in January/June, otherwise they will be transferred to the equivalent three year degree.

In year 1 and year 2, students normally need to achieve a CWA of 55%. Exceptional cases may be approved by the appropriate assessment boards.

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

**11. Scheme of Assessment**

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

If regulation 5.23(c) applies in relation to any of the modules CO1102, CO1105, CO1107 then failed marks must be no lower than 35% (rather than the normal 30%) in order for students to proceed and re-sit.

**12. Special features:**

Emphasis on blending long-term foundational knowledge with state-of-the-art technologies and current programming languages; a structured approach to teaching a wide range of programming paradigms; Software Engineering Projects involving an external client wherever possible; Individual Projects with a number of structured milestones.

**13. Indications of programme quality**

British Computer Society Accreditation will be sought, and 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**

**MComp COMPUTER SCIENCE**

**FIRST YEAR MODULES**

**SEMESTER 1**

| **Core Modules** | | **Credits** |
|---|---|---|
| CO1101 | COMPUTING FUNDAMENTALS | 15 |
| CO1102 | PROGRAMMING FUNDAMENTALS | 15 |
| CO1103 | MATHEMATICS FUNDAMENTALS | 15 |
| CO1104 | COMPUTER ARCHITECTURE | 15 |
| | **Semester Total** | **60** |

**SEMESTER 2**

| **Core Modules** | | **Credits** |
|---|---|---|
| CO1105 | INTRODUCTION TO OBJECT ORIENTED PROGRAMMING | 15 |
| CO1106 | REQUIREMENTS ENGINEERING AND PROFESSIONAL PRACTICE | 15 |
| CO1107 | ALGORITHMS, DATA STRUCTURES AND ADVANCED PROGRAMMING | 15 |
| CO1108 | FOUNDATIONS OF COMPUTATION | 15 |
| | **Semester Total** | **60** |

**SECOND YEAR MODULES**

**SEMESTER 1**

| **Core Modules** | | **Credits** |
|---|---|---|
| CO2101 | OPERATING SYSTEMS AND NETWORKS | 15 |
| CO2102 | DATABASES AND DOMAIN MODELLING | 15 |
| CO2103 | SOFTWARE ARCHITECTURE AND SYSTEM DEVELOPMENT [PARTI] | 15 |
| CO2201 | SOFTWARE ENGINEERING PROJECT [PART I] | 15 |
| | **Semester Total** | **60** |

**SEMESTER 2**

| **Core Modules** | | **Credits** |
|---|---|---|
| CO2103 | SOFTWARE ARCHITECTURE AND SYSTEM DEVELOPMENT [PART II] | 15 |
| CO2104 | USER INTERFACES AND HCI | 15 |
| CO2201 | SOFTWARE ENGINEERING PROJECT [PART II] | 15 |

**Optional Modules**

15 credits of options selected from:

| | | |
|---|---|---|
| CO2106 | DATA ANALYTICS | 15 |
| CO2113 | ENTERTAINMENT TECHNOLOGY | 15 |
| CO2114 | FOUNDATIONS OF ARTIFICIAL INTELLIGENCE | 15 |
| | **Semester Total** | **60** |

**THIRD YEAR MODULES**

**SEMESTER 1**

| **Core Modules** | | **Credits** |
|---|---|---|
| CO3201 | COMPUTER SCIENCE PROJECT [PART I] | 15 |
| CO3101 | COMPUTERS, SOCIETY & PROFESSIONALISM | 15 |

**Optional Modules**

30 credits of options selected from:

| | | |
|---|---|---|
| CO3091 | COMPUTATIONAL INTELLIGENCE AND SOFTWARE ENGINEERING | 15 |
| CO3095 | SOFTWARE MEASUREMENT AND QUALITY ASSURANCE | 15 |

| | | | |
|---|---|---|---|
| CO3102 | MOBILE AND WEB APPLICATIONS | | 15 |
| CO3105 | C++ PROGRAMMING | | 15 |
| CO3219 | INTERNET AND CLOUD COMPUTING | | 15+ |
| | | **Semester Total** | **60** |

**SEMESTER 2**

| **Core Modules** | | | **Credits** |
|---|---|---|---|
| CO3201 | COMPUTER SCIENCE PROJECT [PART II] | | 30 |
| **Optional Modules** | | | |
| 30 credits of options selected from: | | | |
| CO3002 | ANALYSIS OF ALGORITHMS | | 15 |
| CO3093 | BIG DATA AND PREDICTIVE ANALYTICS | | 15 |
| CO3099 | FOUNDATIONS OF CYBER SECURITY | | 15 |
| CO3103 | TECHNOLOGY MANAGEMENT | | 15 |
| CO3107 | EMERGING TECHNOLOGIES | | 15 |
| CO3111 | FUNCTIONAL PROGRAMMING | | 15 |
| CO3113 | AI FOR SPACE | | 15 |
| | | **Semester Total** | **60** |

**+ NOTE:**
**If students take CO3219 in year 3 they cannot also take CO4219 in year 4**

**FOURTH YEAR MODULES**

**SEMESTER 1**

| **Core Modules** | | | **Credits** |
|---|---|---|---|
| CO4015 | COMPUTER SCIENCE PROJECT (PART 1)* | | 15 |
| **Optional Modules** | | | |
| 45 credits of options selected from: | | | |
| CO4210 | PERSONAL AND GROUP SKILLS** | | 15 |
| CO4105 | ADVANCED C++ PROGRAMMING | | 15 |
| CO4215 | ADVANCED WEB TECHNOLOGIES | | 15 |
| CO4217 | AGILE CLOUD AUTOMATION | | 15 |
| CO4219 | INTERNET AND CLOUD COMPUTING+ | | 15 |
| CO4223 | INTERACTION DESIGN (ID) AND USER EXPERIENCE (UX) | | 15 |
| CO4224 | MOBILE AND UBIQUITOUS COMPUTING | | 15 |
| | | **Semester Total** | **60** |

**SEMESTER 2**

| **Core Modules** | | | **Credits** |
|---|---|---|---|
| CO4015 | COMPUTER SCIENCE PROJECT (PART 2)* | | 15 |
| **Optional Modules** | | | |
| 45 credits of optional modules selected from: | | | |
| CO4210 | PERSONAL AND GROUP SKILLS** | | 15 |
| CO4200 | ALGORITHMS FOR BIOINFORMATICS | | 15 |
| CO4207 | GENERATIVE DEVELOPMENT+ | | 15 |
| CO4214 | SERVICE-ORIENTED ARCHITECTURES | | 15 |
| CO4225 | SERVICE DESIGN | | 15 |
| | | **Semester Total** | 60 |

*TAKEN OVER TWO SEMESTERS
** ALL STUDENTS ARE REQUIRED TO TAKE CO4210 BUT MAY TAKE IT EITHER IN SEMESTER 1 OR IN SEMESTER 2.
**+ NOTE:**
**If students take CO3219 in year 3 they cannot also take CO4219 in year 4**

## MComp COMPUTER SCIENCE WITH A YEAR ABROAD

**First and Second Year Modules**

As for the first- and second-year of the MComp 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 MComp 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 and Fifth Year Modules**

As for the third and fourth year of the MComp degree in Computer Science.

## MComp COMPUTER SCIENCE WITH A YEAR IN INDUSTRY

**First and Second Year Modules**

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

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

**Fourth and Fifth Year Modules**

As for the third and fourth year of the MComp degree in Computer Science.