UNIVERSITY OF LEICESTER

Programme Specification (Undergraduate) FOR ENTRY YEAR: 2023/24

Date created: n/a **Last amended:** 26/06/2026 **Version no.** 4

1. Programme title(s) and 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)

Diploma of Higher Education in Computer Science*

Certificate of Higher Education in Computer Science*

Notes

* An award marked with an asterisk is only available as an exit award and is not available for students to register onto.

a) HECOS Code

HECOS Code	%
100366	100%

b) UCAS Code (where required)

Codes listed above.

2. Awarding body or institution:

University of Leicester

3. a) Mode of study

Full-time

b) Type of study

Campus-based

4. Registration periods:

MComp Computer Science

The normal period of registration is 4 years

The maximum period of registration 6 years

MComp Computer Science with a Year Abroad

The normal period of registration is 5 years

The maximum period of registration 7 years

MComp Computer Science with a Year in Industry

The normal period of registration is 5 years

The maximum period of registration 7 years

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.

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
- Framework for Higher Education Qualifications (FHEQ)
- UK Quality Code for Higher Education
- University Education Strategy
- University Assessment Strategy
- University of Leicester Periodic Developmental Review Report
- External Examiners' reports (annual)
- United Nations Education for Sustainable Development Goals
- Student Destinations Data

9. Programme Outcomes

Unless otherwise stated, programme outcomes apply to all awards specified in 1. Programme title(s).

a) Discipline specific knowledge and competencies

i) Mastery of an appropriate body of knowledge

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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.

ii) Understanding and application of key concepts and techniques

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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.
4. Demonstrate ability to communicate some aspects of Computer Science in a foreign language. (Year Abroad)	Lectures and language instruction.	As above
5. Work as a computer scientist and computing engineer in an industrial or commercial setting. (Year in industry)	Work placement.	University report.

iii) Critical analysis of key issues

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Analyse ambiguous client/customer problems, requirements and criteria, and hence plan an appropriate yet innovative solution strategy.	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 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.	As above.	As above.
3. Be able to recognise risks in the deployment and use of software systems.	As above.	As above.

iv) Clear and concise presentation of material

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Evaluate and appraise software systems, in terms of attributes and tradeoffs. Identify risks and safety concerns.	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?
2. Perform software testing, and critically evaluate and analyse test results. Evaluate whether a system meets requirements, for future and for current use.	As above.	As above.
3. Use relevant knowledge to appraise the commercial use and economic and long-term viability of computer systems.	As above.	As above.

vi) Other discipline specific competencies

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Explain and discuss social, legal and ethical issues as required by computing professionals. Adopt and implement suitable professional and legal practice.	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 and react to the rapidity of change in Computer Science. Formulate innovative and creative ideas for future advances.	As above.	As above.
3. Collect, work with and analyse all forms of data.	As above.	As above.

b) Transferable skills

i) Oral communication

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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.

iv) Numeracy

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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. Analyse 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

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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.

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
3. Recognize the need for information, and work with fuzzy, limited and possibly contradictory information.	As above.	As above.

viii) Skills for lifelong learning

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
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 Progression set out in Senate Regulations – see the version of Senate Regulation 5 governing undergraduate programmes relevant to the year of entry.

The following additional progression requirements for this programme have been approved:

Regulation 5.10 applies absolutely to CO2301/CO2302 Software Engineering Project and CO2123/CO2124 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.

a) Course transfers

Students can transfer from BSc Computer Science to MComp Computer Science if they have achieved at least 55% overall for year 2. If students meet this condition they can transfer up until the second semester of year 3.

b) Year Abroad

The Year Abroad variant of this programme is offered in accordance with the University's standard specification for year abroad programme variants.

c) Year in Industry

The Year in Industry variant of this programme is offered in accordance with the University's standard specification for year in industry programme variants.

Students will have the opportunity to take a Year in Industry either between levels 5 and 6 OR levels 6 and 7. Student can only take a Year in Industry on one occasion and cannot take both a Year in Industry AND a Year Abroad.

To take a Year in Industry after level 5, students would need to meet standard University eligibility requirements to progress to the next level of study. Students who obtain a level 5 CWA of less than 55.00% will be permitted to take a Year in Industry but will not be eligible for progression to level 7, and therefore would revert to a BSc (with a Year in Industry). See 'Progression from levels 4 to 5 and 5 to 6' for more information.

To take a Year in Industry after level 6, students would need to have met the criteria to remain on the MComp programme.

11. Criteria for award and classification

This programme follows the standard scheme of undergraduate award and classification set out in Senate Regulations – see the version of *Senate Regulation 5 governing undergraduate programmes* relevant to the year of entry.

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(s) reports

The details of the External Examiner(s) for this programme and the most recent External Examiners' reports for this programme can be found at exam-papers [log-in required]

UNIVERSITY OF LEICESTER

Programme Specification (Undergraduate) FOR ENTRY YEAR: 2023/24

Date created: n/a **Last amended:** 26/06/2026 **Version no.** 4

Appendix 1: Programme structure (programme regulations)

The University regularly reviews its programmes and modules to ensure that they reflect the current status of the discipline and offer the best learning experience to students. On occasion, it may be necessary to alter particular aspects of a course or module.

MComp Computer Science

Level 4/Year 1 2023/24

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core	n/a	60 credits	60 credits
Optional	n/a	n/a	n/a

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Sem 1	CO1101	COMPUTING FUNDAMENTALS	15 credits
Sem 1	CO1102	PROGRAMMING FUNDAMENTALS	15 credits
Sem 1	CO1103	MATHEMATICS FUNDAMENTALS	15 credits
Sem 1	CO1104	COMPUTER ARCHITECTURE	15 credits
Sem 2	CO1105	INTRODUCTION TO OBJECT ORIENTED PROGRAMMING	15 credits
Sem 2	CO1106	REQUIREMENTS ENGINEERING AND PROFESSIONAL PRACTICE	15 credits
Sem 2	CO1107	ALGORITHMS, DATA STRUCTURES AND ADVANCED PROGRAMMING	15 credits

Delivery period	Code	Title	Credits
Sem 2	CO1108	FOUNDATIONS OF COMPUTATION	15 credits

Notes
N/A

Level 5/Year 2 2024/25

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core	n/a	60 credits	45 credits
Optional	n/a	n/a	15 credits

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Sem 1	CO2123	SOFTWARE ARCHITECTURE AND SYSTEM DEVELOPMENT – I	15 credits
Sem 1	CO2301	PROJECT MANAGEMENT	15 credits
Sem 1	CO2101	OPERATING SYSTEMS AND NETWORKS	15 credits
Sem 1	CO2102	DATABASES AND DOMAIN MODELLING	15 credits
Sem 2	CO2124	SOFTWARE ARCHITECTURE AND SYSTEM DEVELOPMENT - II	15 credits
Sem 2	CO2302	SOFTWARE ENGINEERING GROUP PROJECT	15 credits
Sem 2	CO2104	USER INTERFACES AND HCI	15 credits

Notes
N/A

Option modules

Delivery period	Code	Title	Credits
Semester 2	CO2106	DATA ANALYTICS	15 credits
Semester 2	CO2113	ENTERTAINMENT TECHNOLOGY	15 credits
Semester 2	CO2114	FOUNDATIONS OF ARTIFICIAL INTELLIGENCE	15 credits

Choose one optional module in Semester 2

Notes
This is an indicative list of option modules and not definitive of what will be available. Option module choice is also subject to availability, timetabling, student number restrictions and, where appropriate, students having taken appropriate pre-requisite modules.

Level 6/Year 3 2025/26

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core	45 credits	15 credits	n/a
Optional	n/a	30 credits	30 credits

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Year long	CO3201	COMPUTER SCIENCE PROJECT	45 credits
Sem 1	CO3101	COMPUTERS, SOCIETY & PROFESSIONALISM	15 credits

Notes
N/A

Option modules

Delivery period	Code	Title	Credits
Semester 1	CO3091	COMPUTATIONAL INTELLIGENCE AND SOFTWARE ENGINEERING	15 credits
Semester 1	CO3095	SOFTWARE MEASUREMENT AND QUALITY ASSURANCE	15 credits
Semester 1	CO3102	MOBILE AND WEB APPLICATIONS	15 credits
Semester 1	CO3105	C++ PROGRAMMING	15 credits
Semester 2	CO3002	ANALYSIS AND DESIGN OF ALGORITHMS	15 credits
Semester 2	CO3093	BIG DATA AND PREDICTIVE ANALYTICS	15 credits
Semester 2	CO3099	FOUNDATIONS OF CYBER SECURITY	15 credits
Semester 2	CO3103	TECHNOLOGY MANAGEMENT	15 credits
Semester 2	CO3111	FUNCTIONAL PROGRAMMING	15 credits
Semester 2	CO3113	AI FOR SPACE	15 credits

Notes
Choose 30 credits (2 modules) of options in each semester.

Level 7/Year 4 2026/27

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core	30 credits	n/a	n/a
Optional	n/a	45 credits	45 credits

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Year long	CO4015	COMPUTER SCIENCE PROJECT	30 credits

Notes
N/A

Option modules

Delivery period	Code	Title	Credits
Semester 1	CO4210	PERSONAL AND GROUP SKILLS*	15 credits
Semester 2	CO4210	PERSONAL AND GROUP SKILLS*	15 credits
Semester 1	CO4105	ADVANCED C++ PROGRAMMING	15 credits
Semester 1	CO4219	INTERNET AND CLOUD COMPUTING	15 credits
Semester 1	CO4223	INTERACTION DESIGN (ID) AND USER EXPERIENCE (UX)	15 credits
Semester 1	CO4224	MOBILE AND UBIQUITOUS COMPUTING	15 credits
Semester 2	CO4103	TECHNOLOGY AND INNOVATION MANAGEMENT	15 credits
Semester 2	CO4200	ALGORITHMS FOR BIOINFORMATICS	15 credits
Semester 2	CO4207	GENERATIVE DEVELOPMENT	15 credits
Semester 2	CO4214	SERVICE-ORIENTED ARCHITECTURES	15 credits
Semester 2	CO4225	SERVICE DESIGN	15 credits

Notes
Choose 45 credits (3 modules) per Semester

*CO4210 must be taken, but can be taken in either semester 1 or 2. Students will choose 45 credits of optional modules in one semester, and 30 credits in the other, depending on the semester in which students take CO4210.

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.

Academic year affected	Module Code(s)	Update

2024/5	CO2201 and CO2103	Modules have been adjusted from being 30 credit year long modules, and split into separate semester versions. There is virtually no change to the delivery of the module otherwise.
2026/7	CO4215	Removal of optional module
2026/7	CO4103	Addition of optional module
2025/6	CO3107	Removal of CO3107 as an optional module.

Appendix 2: Module specifications

See undergraduate module specification database [login required] (Note - modules are organized by year of delivery).