



Programme Specification (Undergraduate)

FOR ENTRY YEAR: 2026/27

Date created: 20/01/2025

Last amended: Click or tap to enter a date.

Version no. 1 Date approved

by EQED: Click or tap here to enter text.

1. Programme title(s) and code(s):

G400 BSc Computer Science

G991 BSc Computer Science with Foundation Year

G4G7 BSc Computer Science with Artificial Intelligence

GC49 BSc Computer Science with Cybersecurity

G600 BSc Software Engineering

G410 MComp Computer Science

BSc Computer Systems*

BSc Software Systems*

MComp Computer Systems*

Notes

1. All the above have the following variants
 - a. With a Year in Industry
 - b. With a Year Abroad
2. All the above have the following exit awards apply
 - a. Cert HE
 - b. Dip HE.
3. An award marked with an asterisk is only available as an exit award for students that do not meet PSRB requirements for accredited degrees but do meet the requirements of our Senate Regulations. See Section 10.

a) [HECOS Code](#)

HECOS Code	%
100366 (Computer Science, including MComp and specialisms)	100%
100374 (Software Engineering)	100%

b) UCAS Code (where required)

See programme list 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:

a) All BSc Programmes

The normal period of registration is 3 years. If including a Foundation Year, a Year in Industry, or a Year Abroad the normal period of registration is 4 years. If including a Foundation Year, and either a Year in Industry or a Year Abroad, then the normal period of registration is 5 years.

The maximum period of registration is normally 5 years. If including a Foundation Year, a Year in Industry, or a Year Abroad, then the maximum period of registration is 6 years. If including a Foundation Year and either a Year in Industry or a Year Abroad, then the maximum period of registration is 7 years.

b) All MCOMP programmes

The normal period of registration is 4 years. If including a Year in Industry, or a Year Abroad the normal period of registration is 5 years.

The maximum period of registration is normally 6 years. If including a Year in Industry, or a Year Abroad, then the maximum period of registration is 7 years.

5. Typical entry requirements

a) All BSc Programmes

A level: ABB 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.

b) All MCOMP Programmes

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.

c) GCSE Maths

Software Engineering programmes require 4/C in GCSE Maths, all other programmes require 5/C.

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

- Provide students with a state-of-the-art education in Computer Science that includes both theory and foundations, and practical applications.
- 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.
- Enable students to become proficient in a variety of programming languages, and the underlying principles of programming paradigms.
- Enable students to explain core subjects together with a further range of advanced subjects that reflect the research expertise of the School of Computing and Mathematical Sciences.

- Enable students to develop the skills in the University of Leicester Transferrable Skills Framework.
- Provide students with experience of both team-based and individual project work.
- To 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.

a) In addition to these, BSc programmes have the following aim(s)

- Ensure students will have expertise and understanding at a level where they can embark upon a high quality taught Masters programme in Computer Science.

b) In addition to these, MComp programmes have the following aims(s)

- 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.

c) In addition, for the 'with a Year abroad' variants

- The 'Year Abroad' variant of this programme is offered in accordance with the University's [standard specification for the experiential year abroad variant](#).

d) In addition, for the 'with Industry' variants

- The 'Year in industry' variant of this programme is offered in accordance with the University's [standard specification for year in industry programme variants](#).
- To provide experience of applications of professional and discipline-specific skills in Industry and to reinforce knowledge through its use in different environments.

e) In additional for all specialisms

- Provide students with deeper expertise and knowledge related to their chosen specialism namely Software Engineering, AI, and Cybersecurity

8. Reference points used to inform the programme specification

- [QAA Benchmarking Statement for Computing 2022](#)
- Framework for Higher Education Qualifications (FHEQ)
- UK Quality Code for Higher Education
- [University Education Strategy](#)
- [University Assessment Strategy](#) [login required]
- United Nations Education for Sustainable Development Goals
- Student Destinations Data
- External Examiners' reports (annual)

Programme Specification (Undergraduate)

FOR ENTRY YEAR: 2026/27

Date created: 20/01/2025

Last amended: [Click or tap to enter a date.](#) Version no. 1 Date approved by EQED: [Click or tap here to enter text.](#)

9. Programme Outcomes

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

a) Knowledge and Critical Understanding

i) Competence in an appropriate body of knowledge

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Explain, discuss and apply engineering principles scientific principles. Explain and apply mathematical and logical theories in computing	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, and project dissertations.

ii) Breadth of knowledge

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Explain and discuss both foundations and applications of Computing together with concomitant scientific knowledge and concepts from logic and mathematics.	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, and project dissertations.
[MCOMP] Explain and discuss advanced topics in Computer Science at levels which are equal to typical Master's level programmes.	As above	As above	As above

[Specialism] Discuss a broader understanding of knowledge within a chosen specialism.	As above	As above	As above
---	----------	----------	----------

b) Cognitive and Practical Skills

i) Selection and analysis of sources

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Evaluate and appraise software systems, in terms of attributes and trade-offs. Identify risks including data privacy and cyber security 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, and project dissertations.
[All] 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	As above
[All] Use relevant knowledge to appraise the commercial use and economic and long- term viability of computer systems.	As above	As above	As above
[All] Conduct significant background research and literature surveys, and summarise content from information sources.	As above	As above	As above
[All] Recognise the need for information, and work with fuzzy,	Lectures.	Project work and supervision	Written examinations, assessed coursework, and project reports.

limited and possibly contradictory information.			
---	--	--	--

ii) Critical engagement

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Analyse 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, and project dissertations.
[All] 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 software engineering process and evaluate the end product, and to work with associated uncertainties.	As above	As above	As above
[All] Be able to recognise risks in the deployment and use of software systems.	As above	As above	As above

iii) Presentation of an argument

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Present information in a variety of forms, chosen to maximise reader/audience impact and	Lectures, tutorials, computer laboratories	Audios & videos, group discussions, project work, guided independent	Written examinations, summative and formative coursework, group and

understanding, such as reports, dissertations, seminars, posters, blogs, podcasts, videos and other current media technologies.		study. Also, background reading and research.	individual project presentations, and project dissertations.
---	--	---	--

iv) Independent research

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Explain and react to the rapidity of change in Computer Science. Formulate innovative and creative ideas for future advances.	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, and project dissertations.

v) Relevant technical skills

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Apply knowledge of Mathematics, Logic and Computing 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, and project dissertations.
[SE, AI, Cybersecurity] Apply knowledge from within a chosen specialism to solve individual problems both seen and unseen.	As above	As above	As above
[MCOMP] apply more advanced Mathematics Logic and Computing to solve individual problems.	As above	As above	As above
[All] Apply the concepts and techniques of abstraction, reification, logical structure and modelling that	As above	As above	As above

pervade Computer Science and Software Engineering to specify, design, implement and test small to medium size computer systems.			
[All] 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 all forms of project work.	As above, with emphasis on project assessments.
[SE] Apply sound business and management techniques to the processes of software engineering, and the deployment and maintenance of software. Analyse legacy code and synthesize innovative future software solutions.	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, and project dissertations.
[AI] Design and implement algorithms to automate routine tasks for an IT related problem efficiently and effectively.	As above	As above	As above
[Cybersecurity] Apply secure methodologies to develop, run, and maintain software tools and networks.	As above	As above	As above
[SE] Deploy and maintain software in a commercial environment	As above	As above	As above

[All] Use a very broad range of software and IT tools, and to choose these appropriately for uses throughout Computer Science.	As above	As above	As above
[Specialism] Choose and use a broad range of software and IT tools relevant for the chosen specialism	As above	As above	Above

vi) Autonomous working

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Adapt to future programming languages and paradigms, and all varieties of software tools and technology.	Lectures, tutorials and laboratories.	Project work, guided independent studies.	Assessed (laboratory) coursework.
[All] 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.	Project work and reflection.	Project reports.

c) Transferable skills

i) Verbal, written and digital communication

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Respond to technical questions with accurate and concise answers.	Lectures and tutorials.	Guided independent studies. Project supervisions.	Group and individual project presentations, individual project examinations.
[All] Demonstrate fluent and sustained scientific, technical and business communication.	As Above	As above	As above

[All] 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.	As above	Written examinations, assessed coursework.
[All] 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.

ii) Numeracy

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Collect, work with and analyse 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, and project dissertations.
[All] Demonstrate understanding of the concept of number. Solve numerical problems.	As above	As above	Written examinations, assessed coursework.
[All] Use analytical, quantitative, and graphical methods, and deploy elementary statistics.	As above, together with project work.	As above together with project work.	As above, along with group and individual project presentations and reports.

iii) Self-reflection

Intended Learning Outcome	Teaching methods	Learning Activities	Assessment Type
---------------------------	------------------	---------------------	-----------------

[All] Explain and discuss social, legal and ethical issues as required by computing professionals. Adopt and implement 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, and project dissertations.
[All] Demonstrate knowledge and understanding of professional and ethical issues, equality, diversity and inclusion, and aspects of the law, in the context of Computing Professionals.	As above	As above	Written examinations, assessed coursework, and project reports.

iv) Problem solving

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Solve a variety of problems (both simple and advanced) through the integration of knowledge of mathematics, logic, and Computer Science.	Lectures and tutorials.	Guided independent study, formative assignments, project work and supervision.	Written examinations, assessed coursework, and project reports.
[Specialism] Solve problems using the application of skills and knowledge of the chosen specialism	As above	As above	As above
[All] Use systematic analysis and design methods, and appropriate algorithms, to solve medium scale problems.	As above	As above	As above

[All] 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	As above	Group and individual project presentations and reports.
[MCOMP] Analyse and solve larger scale or more complex problems through the application of advanced Computer Science	As above	As above	As above

v) Organisation and management

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Demonstrate a broad understanding of problems and issues that arise in the location, organisation, processing and evaluation of data.	Lectures and tutorials.	Guided independent study, formative assignments, project work and supervision.	Written examinations, assessed coursework, and project reports.
[All] Demonstrate independence and time management skills.	Project supervisions and research project work. Meeting coursework deadlines.	Planning project work and managing university schedules.	Project Reports

vi) Teamwork

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
[All] Work effectively as part of a team, organise roles and manage time, undertake assigned tasks, and ensure final completion of a team	Lectures and tutorials.	Guided independent study, formative assignments, project work and supervision.	Group project assessed coursework and presentations. Mini projects.

project. Identify strengths and weaknesses of team members.			
---	--	--	--

Year Abroad

[In addition, for the 'with a Year abroad' variants the additional programme outcomes apply](#)

Year in Industry

[In addition, for the Year in Industry' variants the additional programme outcomes apply](#)



Programme Specification (Undergraduate)

FOR ENTRY YEAR: 2026/27

Date created: 20/01/2025

Last amended: Click or tap to enter a date. Version no. 1 Date approved

by EQED: Click or tap here to enter text.

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.

a) Course transfers

Students may transfer between BSc programmes within this specification if all core modules for the destination programme have been or can be passed and entry requirements for the destination programme have been met.

Students will be informed, prior to transfer, if they are moving from an accredited to non-accredited degree.

In cases where a student has failed to meet a requirement to progress, he or she will be required to withdraw from the course.

b) Year abroad

For the Year Abroad variant (for experiential Year Abroad only) [the additional progression points apply](#)

c) Year in Industry

For the Year in Industry variant, the [additional progression points apply](#)

d) For Foundation Year Variant:

For the progression requirements from Foundation Year to Year 1, students should refer to the STEM Foundation Year or International Foundation Year programme specification from their year of entry. In cases where a student has failed to meet a requirement to progress they will be required to withdraw from the course.

e) Progression on a specialist degree

Students taking a specialist degree pathway must study specific modules. These modules are therefore core for that pathway as summarised below:

	Computer Science (BSc and MComp)	Software Engineering	Computer Science with Artificial Intelligence	Computer Science with Cybersecurity
Year 2	CO2118	CO2104	CO2114	CO2303
Year 3	n/a	CO3102 CO3207	CO3115 CO3116	CO3221 CO3099

f) Progression throughout accredited degrees

The progression requirements for the accrediting body are different to those in Senate Regulations 5. To remain on our accredited degrees a student must:

1. Require no more than 30 compensated credits throughout the programme.
2. Pass CO3201 without compensation.

Failing to meet these requirements but otherwise meeting the requirements of Senate Regulation 5 will result in a transfer onto a non-accredited equivalent degree: BSc Computer Systems for BSc Computer Science, BSc Software Systems for BSc Software Engineering and MComp Computer Systems for MComp Computer Science.

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

For the Year Abroad variants (for experiential Year Abroad only) [the additional Special Features apply](#)

For the Year in Industry variants [the additional Special Features apply](#).

By experiencing real-world scenarios and applying skills and knowledge to a professional environment, students can gain a unique insight into how their studies can be utilised in industry. This will not only showcase their abilities to future employers but will also enhance their studies upon returning to university to complete your programme.

To understand the special features for year in industry undergraduate programme variants, this programme specification should be read in conjunction with the [programme specification content which can be found here](#). This outlines details including programme aims, support, progression and duration. Students engage in co-curricular modules (Personal and Professional Development) at level 4, 5, and 6. Co-curricular modules support, recognise and reward the learning that occurs outside of traditional taught modules, for example while volunteering, working, or participating in academic representation. This ensures that students that need or want to engage fully in non-academic activities benefit from this.

12a. Research-inspired Education

Students on the Computer Science, Software Engineering, Computer Science with AI, and Computer Science with Cybersecurity programmes will advance through the four quadrants of the University of Leicester Research-inspired Education Framework as follows:

RiE Quadrant	Narrative
<p>Research-briefed</p> <p>Bringing staff research content into the curriculum.</p>	<p>The programme puts emphasis on blending long-term foundational knowledge with state-of-the-art research-based technologies and current programming languages. It offers a structured approach to teaching a wide range of programming paradigms. Students will take part in Software Engineering Group Projects, and in Individual Projects with structured milestones.</p> <p>[MComp] Students will have the opportunity to engage with modules developed to align with the research expertise of our academics.</p> <p>Research-briefed: Students will be exposed to programming languages and paradigms (Java, Python, ...) all of which are actively being used in research, as well as in industry. For each programming language, the students will be shown, and taught, its most common applications. Students will be introduced to a range of advanced that reflect the research expertise of the School.</p>

<p>Research-based</p> <p>Framed enquiry for exploring existing knowledge.</p> <p>Research-oriented</p> <p>Students critique published research content and process.</p> <p>Research-apprenticed</p> <p>Experiencing the research process and methods; building new knowledge.</p>	<p>Research-based: During computer labs, students will have an opportunity to put their problem-solving and research skills in practice by solving problems with applications to data analysis, machine learning, AI, and more.</p> <p>Research-oriented: Students will be able to search information effectively and organise and present information in the form of an IT literature survey. Students will also have the opportunity to evaluate the outcomes of a project, including social, legal and ethical considerations.</p> <p>Research-apprenticed: Students will work in groups and individually to produce complete pieces of software, be it a website, a game demo, an app, from various IT applications. Students will then have the opportunity to present their work and process and be challenged on choices that were made throughout the development of the project.</p>
--	--

As part of studying at a research-intensive university, students on this programme have the following extra or co-curricular opportunities available to them to gain exposure to research culture:

The School helps organise multiple Hackathons during the academic year where the students can come together and collaboratively work or build new software. These Hackathons often have industrial partnerships and collaborators, for example IBM and Capital One. Students are informed and invited to participate in these events via emails.

Students can apply to join the DriverLeics group, which was invited to demonstrate autonomous technologies at the Royal Society Summer Science Exhibition. Successful candidates will engage in research-inspired learning activities in autonomous systems, such as robotics and autonomous vehicles. They will also have opportunities to participate in national and international competitions, such as F1Tenth, and take part in local outreach and voluntary STEM activities.

Throughout term, subject specific career drop-in sessions are scheduled (and added to the students' timetable), for students to find out more about the subject and research specific paths in Computer Science.

Teaching on this programme will be research-informed (it draws consciously on systematic inquiry into the teaching and learning process itself) in the following way:

The School supports all staff involved in teaching to gain an accredited Higher Education teaching qualification, in which they demonstrate their use of teaching theory to support their own practice and reflect on their current teaching and continuing professional development.

All module convenors are part of teaching pods, which group similar fields together. These pods are designed to provide a forum for discussion between teaching-focussed and teaching/research staff, and as a way for more experienced staff to support others by, for example, peer observation and feedback. This provides a platform for staff to share considerations and observations of their teaching experience and obtain research-based input.

Teaching staff meet once a year for a 'Teaching Away Day', which gives the opportunity to discuss some key issues in depth with the other members within the teaching pods and shared with everyone. This gives a chance to share ideas and experience, and to identify questions that need answers.

Additionally, staff will be paired within their teaching pods to observe each other's teaching sessions then meet to agree actions to participate in UoL's Peer Observation of Teaching scheme.

12b. Work-related learning

The workload of all students on this programme includes the opportunity to engage with *at least* 100 hours of employer informed, work-related learning activity. Further information regarding work-related learning is available [online](#).

13. Indications of programme quality

- PSRB Accreditation
- External examiners reports
- Graduate Outcomes Survey Data

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 exampapers@Leicester [log-in required].

Programme Specification (Undergraduate)

FOR ENTRY YEAR: 2026/27

Date created: 20/01/2025

Last amended: [Click or tap to enter a date.](#) Version no. 1 Date approved by EQED: [Click or tap here to enter text.](#)

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.

Level 4/Year 1 2026/27

Credit breakdown

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

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Semester 1	CO1101	Computing Fundamentals	15 credits
Semester 1	CO1102	Programming Fundamentals	15 credits
Semester 1	CO1103	Mathematics Fundamentals	15 credits
Semester 1	CO1104	Computer Architecture	15 credits
Year long	CO1400	Personal and Professional Development - I	15 credits
Semester 2	CO1105	Introduction to Object Oriented Programming	15 credits
Semester 2	CO1106	Software Lifecycle and Quality	15 credits
Semester 2	CO1107	Algorithms, Data Structures and Advanced Programming	15 credits

Notes

n/a

Level 5/Year 2 2027/28

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core	15 credits	60 credits	30 credits
Optional	n/a	n/a	15 credits

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Semester 1	CO2101	Operating Systems and Networks	15 credits
Semester 1	CO2102	Databases and Domain Modelling	15 credits
Semester 1	CO2115	Information Security Fundamentals	15 credits
Year long	CO2400	Personal and Professional Development - II	15 credits
Semester 1	CO2123	Software Architecture and System Development – I	15 credits
Semester 2	CO2124	Software Architecture and System Development – II	15 credits

Option modules

Delivery period	Code	Title	Credits
Semester 2	CO2106	Data Analytics	15 credits
Semester 2	CO2118	Foundations of Computation	15 credits
Semester 2	CO2104	User Interface Design and Evaluation	15 credits
Semester 2	CO2114	Foundations of Artificial Intelligence	15 credits

Delivery period	Code	Title	Credits
Semester 2	CO2303	Secure Software Development	15 credits

Notes

Some programmes have additional requirements that students must pass specific modules related to their pathway degree specialism. These are listed in Section 9 *Progression on a specialist degree*

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 2028/29

Credit breakdown – Computer Science (BSc and MComp)

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

Credit breakdown – All other pathways

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

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Year long	CO3201	Computer Science Project	45 credits
Semester 1	CO3101	Computers, Society & Professionalism	15 credits

Option modules

Delivery period	Code	Title	Credits
Semester 1	CO3102	Mobile and Web Applications	15 credits
Semester 1	CO3105	C++ Programming	15 credits
Semester 1	CO3219	Internet and Cloud Computing	15 credits
Semester 1	CO3221	Ethical Hacking and Penetration Testing	15 credits
Semester 1	CO3220	AI Security, Ethics, and Management	15 credits
Semester 1	CO3114	Data Science and Visualisation	15 credits
Semester 1	CO3115	Advanced Artificial Intelligence	15 credits
Semester 1	CO3095	Software Measurement and Quality Assurance	15 credits
Semester 2	CO3002	Analysis and Design of Algorithms	15 credits
Semester 2	CO3099	Cryptosystems	15 credits
Semester 2	CO3103	Technology and Innovation Management	15 credits
Semester 2	CO3111	Functional Programming	15 credits
Semester 2	CO3113	Artificial Intelligence for Space	15 credits
Semester 2	CO3116	Deep Learning and Computer Vision	15 credits
Semester 2	CO3207	Generative Development	15 credits

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. Optional modules can be mandatory for a given stream; see Section 9 e.

Level 7/Year 4 2029/30

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

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 2	CO4103	Technology and Innovation Management	15 credits
Semester 1	CO4105	Advanced C++ Programming	15 credits
Semester 1	CO4217	Agile Cloud Automation	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	CO4093	Big Data and Predictive Analytics	15 credits
Semester 1	CO4224	Mobile And Ubiquitous Computing	15 credits
Semester 2	CO4200	Algorithms For Bioinformatics	15 credits
Semester 2	CO4207	Generative Development	15 credits

Delivery period	Code	Title	Credits
Semester 2	CO4214	Service-Oriented Architectures	15 credits
Semester 2	CO4225	Service Design	15 credits

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.

Students may not take level 7 modules if they have already taken their level 6 equivalents: CO3207 → CO4207, CO3105 → CO4105, CO3219 → CO4219 and CO3103 → CO4103.

Appendix 2: Module specifications

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