UNIVERSITY OF LEICESTER

Programme Specification (Undergraduate) FOR ENTRY YEAR 2027/28

Date created: 21/01/2024 **Last amended:** 06/06/2026 **Version no.** 1 **Date approved by EQED:** n/a

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

BEng Computer Science (Artificial Intelligence)

BEng Computer Science (Data Science)

BEng Computer Science (Software Engineering)

These are **closed programmes** and available to TAU students only.

a) HECOS Code

HECOS Code	%
100366	100%

b) UCAS Code (where required)

n/a

2. Awarding body or institution:

University of Leicester

3. a) Mode of study

Full-time

b) Type of study

Campus-based

4. Registration periods:

For each degree:

The normal period of registration is 1 years

The maximum period of registration 2 years

5. Typical entry requirements

- All students must have English language IELTs 6.5 (with no individual component less than 6.0).
- The following conditions then apply
  1. If a student has an overall credit weighted average mark of at least 80% from their first two years of study at TAU, they are admitted.
  2. If 1. is not the case, then a student with an overall credit weighted average mark of at least 65%, and a credit weighted average mark across fundamental modules of at least 75% will be admitted.
  3. If 1. is not the case, then a student with an overall credit weighted average mark of at least 60%, and a credit weighted average mark across fundamental modules of between 60% and 75%, will have their application and TAU performance reviewed in detail by the CMS (Computing and Mathematical Sciences) Admissions Team, and the CMS representative on the TAU Board of Studies.

This decision process is summarised in the table below

Overall Average Mark Years 1 and 2	**Fundamental Modules Average Mark Years 1 and 2**	**Admit**
>80	N/A	Y
65-->80	>75	Y
60-->80	60-->75	review
<60	N/A	N

There are 5 fundamental modules at TAU, across years one and two, which are *Programming Fundamentals, Data Structures, Object Orientation through Java, Foundations of Data Science, and Basics of AI.*

Admission decisions will be made in the spring immediately before the academic year of registration.

6. Accreditation of Prior Learning

These programs are based upon APL agreements with The Apollo University (TAU), Chittoor, India, based on 2 years of study at TAU of an agreed diet of modules. These modules reflect those of Leicester BSc Computer Science, in that the overall syllabi and learning outcomes are the same. This ensures that students who meet the required level of achievement at TAU are prepared to enter a level 6 programme at UoL.

7. Programme aims

The programme aims to

- Teach elements of international computing ethics, law and the social impact of computers and computation. Students should be able to critique and analyse typical situations involving ethics and social impact and make informed opinions.
- Create students who can specify, design, implement and test software systems, based on fundamental requirements, and who can write a significant technical dissertation summarising their work. According to the degree programme, the project may have a general focus, or should revolve around work in Artificial Intelligence or Data Science or Software Engineering.
- Teach a range of topics at an advanced level in artificial intelligence, data science or Software Engineering. Students should be able to articulate understanding of fundamental knowledge and be able to solve a range of problems in these areas. Wherever possible, the education of students will be founded on staff research and expertise.
- Teach an advanced understanding of a widely applicable programming language which can substantially underpin students' work throughout the year.
- Create students who can work independently, efficiently organize their work, and be able to meet demanding deadlines, and who have an international perspective of computing. Students should uphold University values of inclusivity and diversity.

8. Reference points used to inform the programme specification

- QAA Benchmarking Statements: Computing 2022 and Mathematics, Statistics and Operational Research 2023.
- Framework for Higher Education Qualifications (FHEQ)
- UK Quality Code for Higher Education
- Education Strategy
- University Assessment Strategy [log-in required]
- University of Leicester previous Periodic Developmental Review Report for the School of Informatics
- External Examiners' reports (annual)
- United Nations Education for Sustainable Development Goals
- Student Destinations Data

UNIVERSITY OF LEICESTER

Programme Specification (Undergraduate) FOR ENTRY YEAR 2027/28

Date created: 21/01/2024 **Last amended:** 06/06/2026 **Version no.** 1 **Date approved by EQED:** n/a

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
Students should be able to: Explain and discuss both foundations and applications of Computer Science together with concomitant scientific knowledge and concepts from logic and mathematics. Students will have expertise in areas of advanced AI or Data Science or Software Engineering.	Lectures, tutorials, computer laboratories, audios & videos, group discussions.	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: Explain, discuss and apply engineering principles, scientific principles and mathematical and logical theories in computing.	Lectures, tutorials, computer laboratories, audios & videos, group discussions.	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

ii) Breadth of knowledge

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type

Students should be able to: Explain the wider social, industrial, cultural and ethical context in which AI or Data Science or Software Engineering are used, through to applying the technical theory and practice of these areas.	Lectures, tutorials, computer laboratories, audios & videos, group discussions.	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

iii) Understanding of source materials

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Explain and apply the theoretical principles, and practical tools of AI, or Data Science, or Software Engineering, together with suitable processes and methodologies, to determine strategies for innovative solutions of large-scale problems.	Lectures, tutorials, computer laboratories, audios & videos, group discussions.	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

b) Cognitive and Practical Skills

i) Selection and analysis of sources

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to:	Lectures, tutorials, computer laboratories, audios & videos, group discussions.	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations,

Apply knowledge of Mathematics and Computing to solve individual problems, both seen and unseen.			individual project written reports, and individual questioning.
Students should be able to: Evaluate and appraise AI or Data Science, in terms of attributes and trade-offs. Identify risks and safety concerns.	Lectures, tutorials, computer laboratories, audios & videos, group discussions.	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

ii) Critical engagement

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Apply the concepts and techniques of abstraction, reification, logical structure and modelling, that pervade AI, Data Science or Software Engineering to specify, design, implement and test small to medium size computer systems.	Lectures, tutorials, computer laboratories, audios & videos, group discussions.	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: Recognise and analyse risks in the deployment and use of software systems.	Lectures, tutorials, audios and videos, group discussions.	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

iii) Presentation of an argument

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type

Students should be able to: Analyse client/customer problems, requirements and criteria, and hence plan an appropriate yet innovative solution strategy.	Lectures, tutorials, group discussions	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: 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.	Lectures, tutorials, group discussions	Project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

iv) Independent research

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Work to advanced levels in the collection, processing and analysis of all forms of data.	Supervisions, computer laboratories, group discussions.	Practicals, project work, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: Explain and react to the rapidity of change in Computer Science, and formulate innovative and creative ideas for future advances.	Supervisions, computer laboratories, group discussions.	Project work, independent reading and research, preparation of assessments.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

v) Relevant technical skills

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Perform software testing, and critically evaluate and analyse test results. Evaluate whether a system of AI or Data Science or Software Engineering meets requirements, for future and for current use.	Computer laboratories, audios & videos	Practicals, project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: Use relevant knowledge to appraise the commercial use and economic and long-term viability of computer systems.	Tutorials, computer laboratories, audios & videos	Practicals, project work, guided independent study, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

vi) Autonomous working

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Formulate an individual analysis of social, legal and ethical issues as required by computing professionals, in western and Asian paradigms.	Tutorials, group discussions.	Project work, background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

vii) Presentation of research findings

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Present information in a variety of forms, chosen to maximise reader/audience impact and understanding, such as reports, seminar presentations, posters, blogs, podcasts, videos and other current media technologies.	Demonstrated through lectures, explained in tutorials.	Project work, tutorials, presentations	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

c) Transferable skills

i) Verbal, written and digital communication

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Respond verbally to technical questions with accurate and concise answers, and also demonstrate fluent and sustained scientific, technical and business communication.	Lectures, tutorials, computer laboratories, audios & videos, group discussions.	Project work, tutorials, presentations.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: Write concise and accurate summaries of computing and scientific knowledge, and solutions to problems, in a variety of different formats. They should also be able to produce properly structured, clear,	Tutorials, group discussions, project supervisions.	Project work, independent research, assessment preparation.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

advanced technical reports or projects.			

ii) Numeracy

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Demonstrate understanding of the concept of number and solve numerical problems.	Lectures, tutorials, computer laboratories.	Project work, guided independent study, independent research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: Use analytical, quantitative, and graphical methods, and deploy elementary statistics. Students in Data Science will have advanced skills in the visualisation of data.	Tutorials, project supervisions.	Project work, guided independent study, independent research, assessment preparation.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

iii) Self-reflection

Intended Learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Reflect on their knowledge and understanding of professional and ethical issues, and aspects of the law, in the context of Computing Professionals.	Tutorials, group discussions.	Preparation of video assignment, tutorial discussions.	Individual video submission on an ethical issue of the students' own choosing.

iv) Problem solving

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Apply knowledge of Mathematics, Logic and Computer Science to solve individual problems, both seen and unseen.	Lectures, tutorials, computer laboratories, group discussions.	Project work, guided independent study, background reading and research, preparation of assessments.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: Use systematic analysis and design methods, and appropriate algorithms, to solve medium scale problems.	Lectures, tutorials, computer laboratories, group discussions.	Project work, guided independent study, background reading and research, preparation of assessments.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.
Students should be able to: Analyse large-scale problems to produce suitable solutions with sensible economic and commercial compromises. Apply management techniques to allocate resources to projects.	Lectures, tutorials, computer laboratories, group discussions.	Project work, guided independent study, background reading and research, preparation of assessments.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

v) Organization and management

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: Demonstrate independence and time management skills.	Tutorials, computer laboratories, group discussions.	Project work, guided independent study, revision for exams, preparation of assignments.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

Students should be able to: Design a personal work plan and be able to improve performance with a clear view of long-term professional development.	Tutorials, project supervisions	Project work, independent research	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

vi) Teamwork

Intended learning Outcome	Teaching methods	Learning Activities	Assessment Type
Students should be able to: 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.	Tutorials, computer laboratories, group discussions.	Project work.	Written examinations, summative and formative coursework, group and individual project presentations, individual project written reports, and individual questioning.

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.

BEng/BSc top up degrees consist of 120 credits at Level 6. Such degrees will be classified on the basis of the level 6 credits studied at the University only.

Senate Regulation **SR5.101-102** will apply.

Since these programmes are comprised of a single year of study at Leicester, there are no mid-year formal progression points. However, the programme director and personal tutors will ensure that standard monitoring processes are applied robustly.

Students taking any one of the listed specialist degree pathways must study specific modules. These modules are therefore core for that pathway as summarised below:

Computer Science (Artificial Intelligence)	Computer Science (Data Science)	Computer Science (Software Engineering)
CO3115 S1	CO3114 S1	CO3102 S1

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

The emphasis is on blending long-term foundational knowledge with state-of-the-art technologies and current programming languages. Individual Projects have structured milestones, and offer comprehensive support during early stages.

12a. Research-inspired Education

Students on all programme specialisms will advance through the four quadrants of the University of Leicester Research-inspired Education Framework as follows:

RiE Quadrant	Narrative
	The programme puts emphasis on blending long-term foundational knowledge with state-of-the-art research-based technologies and current programming languages. Students will take part in Individual Projects with focussed specialisation and with structured milestones.

Research-briefed Bringing staff research content into the curriculum.	**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. Students will be introduced to a range of advanced subjects that reflect the research expertise of the School.
Research-based Framed enquiry for exploring existing knowledge.	**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 AI, Data Science, Software Engineering, Cybersecurity, and more.
Research-oriented Students critique published research content and process.	**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 can evaluate the outcomes of a project, including social, legal and ethical considerations.
Research-apprenticed Experiencing the research process and methods; building new knowledge.	**Research-apprenticed:** Students will work individually to produce professional and complete pieces of software, be it a website, a game demo, an app, from various IT applications. Students can present their work and process and be challenged on choices that were made throughout the development of their project.

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 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 may be able to 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.

Students are welcome to attend the regular Computer Science research seminar programme.

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.

13. Indications of programme quality

The programme director for the Apollo programmes will be supported by the UG Programme Lead who oversees all of our UG programmes. They will both sit on the School Education Committee which reports to Senior Management Group, and the College Education Committee.

14. External Examiner(s) reports

This is an entirely new programme, but once running, students' performances will be scrutinized by an external examiner in conformity with Senate Regulations. There are historical External Examiners' reports for the standard BSc Computer Science and BSc Software Engineering. These provide quality assurance, with external oversight, for existing modules.

UNIVERSITY OF LEICESTER

Programme Specification (Undergraduate) FOR ENTRY YEAR 2027/28

Date created: 21/01/2024 **Last amended:** 06/06/2026 **Version no.** 1 **Date approved by EQED:** n/a

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.

Notes

These programmes have requirements that students must pass specific modules related to their pathway degree specialism. These are listed in Section *10 Progression Points*

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 Final 2027/28

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core (see below and Section 10)	45 credits	30 credits	n/a
Optional	n/a	15 credits	30 credits

120 credits in total

Core modules – All specialisms

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

Notes

Students will study 15-credits of CO3201 in Semester 1 and 30-credits in Semester 2. The teaching and assessment schedule for this module supports this balance.

Option modules

Delivery period	Code	Title	Credits
Semester 1	CO3091	Computational Intelligence and Software Engineering	15 credits

Delivery period	Code	Title	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 1	CO3219	Internet and Cloud Computing	15 credits
Semester 1	*CO3114*	*Data Science and Visualisation*	*15 credits*
Semester 1	*CO3115*	*Advanced Artificial Intelligence*	*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	CyberSecurity	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

Note: Option modules in italics are those which are listed in **Section 10 Progression Points** as core for specialisms and must be available to students.

Otherwise 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.

Appendix 2: Module specifications

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