Additional Admissions Form master Program in Computer Science and Engineering

Instructions for filling out this form:

- Please fill out **all** fields within this form, unless indicated otherwise.
- Please submit PDF.
- **Please list all courses that prove you have passed one or more courses on the following subjects:**
    1. Calculus (topics: polynomials, exponentials, limits, Taylor expansion, l'Hôpital's rule, differential equations, integrals)
    2. Linear Algebra (topics: calculation with matrices and vectors, Gauss(-Jordan) elimination, rank, orthogonality, (in)dependency, eigenvalue decomposition, eigenvectors)
    3. Logic & Set Theory (topics: predicate logic, formal deduction system, sets, relations, functions, proof techniques, induction, contradiction)
    4. Programming (topics: write programs from scratch in imperative and object-oriented languages, apply principles of code quality and software engineering)
    5. Algorithms and data structures (topics: algorithm design techniques, standard data structures, prove correctness of algorithms, reason about asymptotic complexity of algorithms)
    6. Automata, formal languages and complexity (topics: deterministic and nondeterministic finite automata, regular expressions, pushdown automata, context-free grammars, Turing machines, (un)decidability, complexity theory)
    7. Data modelling and databases (topics: design data models and query data in relational databases based on natural, both based on natural language requirements)
    8. Machine learning & data mining (topics: feature selection and extraction, supervised learning, unsupervised learning, evaluation methods and overfitting)
- Copy the detailed and comprehensive course descriptions of all the relevant courses you have taken and the courses you are currently taking. Copy this from your study guide and **highlight** in it (in yellow) all the key words mentioned in our admission criteria and where you think you meet these requirements. You may add a **brief** explanation if you think the official course description of your university is incomplete, but it should be very clear what is the official course description and what is your own addition. **Copy all information in this form and safe as PDF; We use this form primarily to determine whether you meet the admission requirements so study very carefully all the courses relevant to the master's that you think you meet! This will benefit your assessment!**
- **Please also indicate in case you do not meet a requirement. Not meeting a requirement does not necessarily imply rejection. The admission board will look at your application.**

Personal details:

Full name (as on passport)	Zeng Qin Tao
TU/e student number	2277522

Information on bachelor's degree:

Name of bachelor's degree program	B.Sc. Computer Science
(Expected) graduation date	2025/06/23
Name of university	University of Birmingham
City and country of university	Birmingham, United Kingdom

Required TU/e subject	**Course name, code and description of course contents (as given in the course syllabus/course catalogue)**	**Credits of the course/total credits of bachelor** *(for example 5 out of 180)* **and final grade**
Calculus	**LI Artificial Intelligence 2, 06 34255** Artificial Intelligence (AI) and Machine Learning are often applied in situations characterised by various kinds of uncertainty, for example uncertainty in data measurements, missing data or uncertainty in our prior knowledge about the problem. This module will provide principles that enable AI to treat uncertainty consistently in inference, search, optimisation and learning. Learning Outcomes: • Conceptually understand frameworks for consistent treatment of uncertainty in AI • Understand principles of inference and fitting to data in AI models under uncertainty. • Describe and understand randomised search and optimisation techniques utilised in AI. Self-Reflection: This course includes foundational probability and statistics concepts relevant to calculus. Some topics such as differential equations and Taylor expansion are not explicitly covered but are applied in AI inference methods.	Credits: 20 / 360 Final Grade: 43/100

Linear algebra	**1. LC Mathematical and Logical Foundations of Computer Science, 06 35324** Mathematical and logical reasoning underpins almost all of Computer Science, from linear algebra in graphics and machine learning, to algebra in cryptography and logic in verification. This module introduces essential topics in mathematics and logic, including linear algebra, abstract algebra, set theory (a fundamental language of mathematics), and propositional and predicate logic. The key ideas will be illustrated with applications across a range of topics in computer science. Learning Outcomes: • Solve mathematical problems in algebra and set theory • Understand and apply algorithms for key problems in logic such as satisfiability. • Write formal proofs for propositional and predicate logic • Apply mathematical and logical techniques to solve a problem within a computer science setting **2. LH Machine Learning, 06 38965** Machine learning studies how computers can autonomously learn from available data, without being explicitly programmed. The module provides a solid foundation in machine learning by giving an overview of the core concepts, theories, methods, and algorithms for learning from data. The emphasis will be on the underlying theoretical foundations, illustrated through a set of methods used in practice. This will provide the student with a good understanding of how, why, and when various machine learning methods work. Learning Outcomes: • Demonstrate knowledge and understanding of core ideas and foundations of automated learning from data. • Demonstrate understanding of broader issues of learning and generalisation in machine learning. • Demonstrate the ability to apply the main approaches to unseen examples. Self-Reflection: This course heavily involves linear algebra in various aspects of machine learning. Key operations such as matrix multiplication, dot products, and vector addition are fundamental in algorithms like gradient descent, linear regression, and neural networks. Gauss elimination is utilized to solve optimization problems, enabling efficient solutions in model training. These linear algebra principles form the foundation for understanding and implementing machine learning models effectively.	1. Credits: 20 / 360 Final Grade: 60/100 2. Credits: 20 / 360 Final Grade: Not Finished Yet

Logic & Set Theory	**LC Mathematical and Logical Foundations of Computer Science, 06 35324** Mathematical and logical reasoning underpins almost all of Computer Science, from linear algebra in graphics and machine learning, to algebra in cryptography and logic in verification. This module introduces essential topics in mathematics and logic, including linear algebra, abstract algebra, set theory (a fundamental language of mathematics), and propositional and predicate logic. The key ideas will be illustrated with applications across a range of topics in computer science. Learning Outcomes: • Solve mathematical problems in algebra and set theory • Understand and apply algorithms for key problems in logic such as satisfiability. • Write formal proofs for propositional and predicate logic • Apply mathematical and logical techniques to solve a problem within a computer science setting	Credits: 20 / 360 Final Grade: 60/100
Programming	**1. LC Object Oriented Programming, 06 34229** Object-oriented programming is one of the most popular techniques in industrial software development. This module introduces students to the principles of object-oriented programming, imperative algorithms, and data structures. Learning Outcomes: • Explain and apply the fundamental constructs of imperative and object-oriented programming, and data structures. • Analyse computer programs, for example, by determining the behaviour of a program from its source code or by completing and/or correcting partially written programs. • Write, test, and debug computer programs, where appropriate making effective use of an integrated development environment (IDE) and other programming aids. • Design and document complete computer programs to solve given software problems, including some open-ended tasks. **2. LI Software Engineering, 06 40098** This module introduces students to the field of software engineering and the principles of systematically engineering large-scale software systems. The module covers widely used techniques for engineering requirements, designing, and modelling, and architecting dependable and evolvable software. The module discusses Software Quality Assurance, testing, and project management with an appreciation of the economical, legal, and ethical aspects. Students will apply their software engineering knowledge in the development of a team project.	1. Credits: 20 / 360 Final Grade: 91/100 2. Credits: 20 / 360 Final Grade: 76/100 3. Credits: 20 / 360 Final Grade: 71/100

	Learning Outcomes: • Describe and apply techniques for systematically engineering requirements, designing, architecting, and testing software systems, along with the technical, economical, legal, and ethical trade-offs involved. • Describe and apply the basic principles of software project management, metrication, and quality assurance. • Understand advances in industrial software engineering. • Demonstrate teamwork and leadership in developing a group project using good software engineering practices. **3. LI Functional Programming, 06 34253** This module develops practical programming skills in a typed functional programming language. It will strengthen the algorithmic and design skills of the students within the functional framework and will introduce them to some advanced programming language features. Learning Outcomes: • Present the basic ideas of functional programming language. • Demonstrate the main elements of good functional programming style. • Illustrate some of the uses and applications of functional programming. • Understand and apply more advanced features of typed functional programming.	
Algorithms and data structures	**LC Data Structure & Algorithms, 06 30175** Algorithms lie at the heart of Computer Science and software development. They embody the way in which we solve problems using computers. This module introduces the fundamentals of data structures and algorithms. Data structures will be formulated to represent information in such a way that it can be conveniently and efficiently manipulated by the algorithms that are developed. The ideas will be presented both abstractly, and via problem-solving and implementations. Learning Outcomes: • Design and implement data structures and algorithms. • Argue that algorithms are correct and derive time and space complexity measures. • Explain and apply data structures in solving programming problems. • Make informed choices between alternative data structures, algorithms, and implementations, justifying choices on grounds such as computational efficiency.	Credits: 20 / 360 Final Grade: 45/100

Automata, formal languages and complexity	**LC Theories of Computation, 06 35393** Computers have been used to solve an astonishing range of different problems, but this does not mean that they can be used to solve all possible problems: some cannot be solved efficiently, and some cannot be solved at all. In this module, we will introduce a set of principles and techniques for formalising computation and computability to understand what problems can be solved, how efficiently they can be solved, and what problems cannot be solved. We will develop mathematical models of computations using ideas such as automata theory (including Turing machines), of formal languages using ideas such as regular expressions and grammars and will conclude by considering the notions of non-computability and complexity. Learning Outcomes: • Explain and apply mathematical models of computations. • Describe and use the connection between finite automata and regular language. • Explain and apply concepts from automata theory, formal language theory, computability theory and complexity theory • Explain non-computability and undecidability issues	Credits: 20 / 360 Final Grade: 53/100
Data modelling and databases	**1. LC Full Stack Application Development, 06 34252** Many modern software applications involve a full software stack – a client-side front-end, a server to interpret and process client requests, and a database to store the application's content. This module will introduce a range of tools and techniques for full stack software development. Learning Outcomes: • Design, populate, and query a database. • Design and implement a front-end interface. • Design and implement server-side application software. • Integrate a database, server, and front-end into a full software stack. • Use modern development tools effectively. **2. LI Security and Networks, 06 30195** As computers are embedded in everyday life, protection against the criminal or unauthorized use of electronic data is essential and measures must be taken to achieve this. The module will introduce a range of topics in the theory and practice of computer security, including attacks, vulnerabilities, and defences. Cloud and Web services will be used as a motivating example for the importance of security. Learning Outcomes:	1. Credits: 20 / 360 Final Grade: 62/100 2. Credits: 20 / 360 Final Grade: 84/100 3. Credits: 20 / 360 Final Grade: 76/100

	• Understand basic concepts of cryptography and SQL. • Understand basic concepts of cloud services, in particular storage. • Demonstrate an understanding of the threats to data stored on a computer, locally or in the cloud. • Demonstrate an understanding of the threats to data sent on the network. • Identify risks and use techniques to eliminate or mitigate them. **3. LI Software Engineering, 06 40098** Self-Reflection: In this module, we implemented UML diagrams to design and model large-scale software systems. The use of UML diagrams, such as use case diagrams, class diagrams, and sequence diagrams, allowed us to visualize and document system requirements, architecture, and interactions systematically.	
Machine learning & data mining	**1. LH Machine Learning, 06 38965** Machine learning studies how computers can autonomously learn from available data, without being explicitly programmed. The module provides a solid foundation in machine learning by giving an overview of the core concepts, theories, methods, and algorithms for learning from data. The emphasis will be on the underlying theoretical foundations, illustrated through a set of methods used in practice. This will provide the student with a good understanding of how, why, and when various machine learning methods work. Learning Outcomes: • Demonstrate knowledge and understanding of core ideas and foundations of automated learning from data. • Demonstrate understanding of broader issues of learning and generalisation in machine learning. • Demonstrate the ability to apply the main approaches to unseen examples. Self-Reflection: This module provided hands-on experience in applying supervised learning, such as classification and regressions. Overfitting was addressed by implementing techniques like cross-validation and regularization (L1 and L2 norms) during model evaluation. **2. LH Neural Computation, 06 32167** This module introduces the basic concepts and techniques of neural computation, and its relation to automated learning in computing machines. It covers the main types of formal neuron and their relation to neurobiology, showing how to construct large neural networks and study their learning	1. Credits: 20 / 360 Final Grade: Not Finished Yet 2. Credits: 20 / 360 Final Grade: Not Finished Yet 3. Credits: 20 / 360 Final Grade: Not Finished Yet

	and generalization abilities in the context of practical applications. It also provides practical experience of designing and implementing a neural network for a real-world application. Learning Outcomes: • Understand the relationship between real brains and simple artificial neural network models. • Describe and explain some of the principal architectures and learning algorithms of neural computation. • Explain the learning and generalisation aspects of neural computation. • Demonstrate an understanding of the benefits and limitations of neural-based learning techniques in context of other state-of-the-art methods of automated learning. • Apply neural computation algorithms to specific technical and scientific problems. **3. LH Natural Language Processing, 06 37810** Natural Language Processing (NLP) enables computers to understand and reason about human languages such as English and has resulted in many exciting technologies such as conversational assistants, machine translation, and intelligent internet search. This module provides the theoretical foundations of NLP as well as applied techniques for extracting and reasoning about information from text. Learning Outcomes: • Demonstrate an understanding of the major topics in Natural Language Processing. • Understand the role of machine learning techniques in widening the coverage of NLP systems. • Demonstrate an ability to apply knowledge-based and statistical techniques to realworld NLP problems.	
Other	**1. LI Artificial Intelligence 1, 06 34238** Covers AI fundamentals, search algorithms, problem-solving techniques. **2. LI Operating Systems and Systems Programming, 06 38059** Memory management, OS structure, systems programming.	1. Credits: 20 / 360 Final Grade: 52 / 100 2. Credits: 20 / 360 Final Grade: 46 / 100

	3. LH Computer Science Project, 06 26581 Capstone project involving advanced computer science topics.	3. Credits: 20 / 360 Final Grade: Not Finished Yet