UNIVERSITY OF LONDON

Programme Regulations 2022–2023

Computer Science

MSc
PGDip
PGCert
Individual modules

Important document – please read
This document contains important information that governs your registration, assessment and programme of study

Contents

Important information regarding the Programme Regulations

Last revised 30 June 2022

As a student registered with the University of London you are governed by the current General Regulations and Programme Regulations associated with your programme of study.

These Programme Regulations are designed and developed by the University of London which is responsible for the academic direction of the programme. The Programme Regulations will provide the detailed rules and guidance for your programme of study.

In addition to Programme Regulations you will have to abide by the General Regulations. These regulations apply to all students registered for a programme of study with the University of London and provide the rules governing registration and assessment on all programmes; they also indicate what you may expect on completion of your programme of study and how you may pursue a complaint, should that be necessary. Programme Regulations should be read in conjunction with the General Regulations.

The relevant General Regulations and the Programme Regulations relating to your registration with us are for the current year and not the year in which you initially registered.

On all matters where the regulations are to be interpreted, or are silent, our decision will be final.

Further information about your programme of study is outlined in the Programme Specification which is available on the relevant Courses page of the website. The Programme Specification gives a broad overview of the structure and content of the programme as well as the learning outcomes students will achieve as they progress.

Terminology

The following language is specific to the **Computer Science** programme:

Module: Individual units of the programme are called module. Each module is a self-contained, formally structured learning experience with a coherent and explicit set of learning outcomes and assessment criteria.

Study session: There are four study sessions in a year, each lasting 10 weeks. Sessions begin in October, January, April and July. Each session is following by an assessment submission point.

Resitting the assessment of a failed module: When you resit a failed module you will not be allocated a tutor group but you will have access to the learning materials on the VLE and you will be required to resubmit your summative assessment.

Repeating a failed module: When you repeat a failed module you will be allocated a tutor group, you will have access to the learning materials on the VLE and you will be required to resubmit your summative assessment. If you repeat a module, you will have to pay the full module fee when you re-register for the module.

Throughout the Regulations, ‘we’ ‘us’ and ‘our’ mean the University of London; ‘you’ and ‘your’ mean the student, or where applicable, all students.

If you have a query about any of the programme information provided please contact us. You should use the *ask a question* button in the student portal.

Significant changes made to the programme regulations 2022-2023

Added 30 June 2022:

Advice has been added to **Regulation 6.1** to note that it is strongly recommended that you attempt the assessment of the *Object-Oriented Programming* module before commencing study of the *Software Design and Programming* module.

1 Structure of the qualifications

Appendix A gives the qualification structures and Appendix B gives the module descriptions.

Qualifications

1.1

The following named qualifications are awarded under the Computer Science programme:

- Master of Science in Computer Science (MSc)
- Postgraduate Diploma in Computer Science (PGDip)
- Postgraduate Certificate in Computer Science (PGCert)

Qualification structures

1.2

The MSc Computer Science consists of:

- ten core modules (15 credits each)
- one Project module (30 credits)

1.3

The PGDip Computer Science consists of:

- eight modules (15 credits each)

1.4

The PGCert Computer Science consists of:

- four modules (15 credits each)

Individual modules taken on a standalone basis

1.5

The following module from the Computer Science programme is available to study on a stand-alone basis, subject to module availability:

- Applied Machine Learning

See course page for information about the modules available for study and when they run.

2 Registration

Effective date of registration

2.1

Your effective date of registration will be either:

- 1 October, if you first register before the September registration deadline
- 1 April, if you first register before the March registration deadline

Date of first assessments

2.2

If your effective date of registration is:

- 1 October, you will take your first assessment(s) in December of the same year
- 1 April, you will take your first assessment(s) in June of the same year

Study sessions

2.3

The programme has two registration points in the year. There are four study sessions in a year, each lasting 10 weeks. Sessions begin in October, January, April and July. Each session is followed by an assessment submission point.

Further information about ratification of grades can be found in Section 6: Progression within the programme

2.4

The 15 credit modules will be taught over one 10-week session.

2.5

The Project is 30 credits and will be taught over two 10-week sessions, beginning in April only.

Module availability

2.6

Where the learning experience may be compromised due to low student registrations, we may consider deferring the module to a later session.

Not all modules will run in every study session.

We will inform you of any such changes as early as possible and provide you with reasonable alternative arrangements.

Period of registration

See the Programme Specification for the minimum and maximum periods of registration applicable to this programme.

2.7

The minimum and maximum periods of registration to complete the programme are counted from your effective date of registration.

See Section 6: Progression within the programme for information on Maximum and minimum number of modules you can study in a study session for.

3 Recognition of prior learning and credit transfer

To be read in conjunction with the General Regulations, Section 3.

Recognition of prior learning

Recognition of prior learning is a generic term for the process by which we recognise and, where appropriate, award credit for learning that has taken place elsewhere, before entry onto a programme of study. Where the prior learning covered a similar syllabus at an appropriate level to a module on the University of London programme, credit will be awarded as if you took the University of London module/course.

3.1

If you are registered for the MSc or PGDip, you may apply for recognition of prior learning mapped against modules to a total of 60 UK credits.

3.2

Applications for recognition of prior learning for the Project will not be accepted.

3.3

If you are registered for the PGCert, you may not apply for recognition of prior learning.

4 Assessment for the programme

See General Regulations for Rules for taking written assessments.

4.1

Summary table of assessment

Module	Applied Machine Learning Cloud Computing Object-Oriented Programming Principles of Programming Software Design and Programming		Computer Systems Data Management Fundamentals of Computing Information Security Information Systems	Project	
Element weighting	25%	75%	100%	30%	70%
Item of assessment	One online test, for example, MCQs or auto-graded problem-solving task	One end of term coursework / online examination	One end of term coursework / online examination	Research Proposal	Project Report

Passing assessments

4.2

The pass mark for each module is 50%. Where there is more than one element of assessment for a module, you do not need to pass each element of assessment, although you do need to obtain an overall weighted mark of 50% in each module.

4.3

For a module with two elements of assessment, if you do not submit the first element of assessment, but do submit the second element of assessment, you will receive a mark of zero for the first element and this will count as an attempt. Your module mark will still be based on the overall weighted mark.

Invalid attempts

4.4

For a module with two elements of assessment, if you submit the first assessment but do not submit the second assessment, this will not count as an attempt at the module and there will be no academic penalty.

4.5

For a module with two elements of assessment, if you submit neither element of assessment, this will not count as an attempt at the module and there will be no academic penalty.

4.6

For a module with one element of assessment, if you do not submit the final assessment, this will not count as an attempt at the module and there will be no academic penalty.

4.7

For the Project, if you do not submit the Research Proposal your attempt at the module will not be valid and you will not be permitted to submit the final assessment. This will not count as an attempt at the module and there will be no academic penalty.

4.8

If you have not made a valid attempt at the module (see regulations 4.4 to 4.7), you will need to re-register and make a new attempt at the module. You will be required to pay the **full module fee**. If there are two elements of assessment, all assessment elements will need to be attempted.

See General Regulations for Rules for taking written assessments

Mitigating circumstances

4.9

For 15 credit modules where there is more than one element of assessment, mitigating circumstances will only be accepted for the second, higher weighted element of assessment.

4.10

For the Project, mitigating circumstances will be accepted for either element of assessment.

Penalty for exceeding the word count of coursework elements

4.11

For coursework elements, you should not exceed the word limit by more than 10%. If the word count is between 10% to 20% above the word limit, the coursework will receive a five mark penalty. If the word count exceeds the word limit by more than 20% you will receive a mark of zero for your work.

Late submission of coursework elements

4.12

You must keep to the deadlines given on the VLE. Coursework elements that are submitted after the deadline will not be marked and the attempt will be considered invalid.

See regulations 4.4 to 4.8 for more information on invalid attempts.

5 Number of attempts permitted at an assessment element

5.1

The maximum number of attempts permitted for any element of assessment is two.

5.2

You will fail the assessment if your overall weighted mark for the module is below 50%.

5.3

You must make a second attempt at the assessment for a module you have failed, provided that you have not exceeded the maximum number of attempts at the assessment/s. If there are two elements of assessment for the module, all assessment elements will need to be attempted.

5.4

If you pass the module overall with a mark of 50% or above, you will not be permitted to make a second attempt at any assessment element.

Resitting the assessment of a failed module

If you resit the assessment for a module, you will have to pay a fee when you re-register for the module to resit the assessment. The fee payable is outlined in the fee schedule.

You will not be allocated a tutor group but will have access to the learning materials on the VLE and will be required to resubmit your summative assessment.

5.5

If you fail the assessment for a module held in the October session or the January session, your resit opportunity will be in the July session of the same academic year.

5.6

If you fail the assessment for a module held in the April session or the July session, your resit opportunity will be in the January session of the following academic year.

5.7

If you do not make a second attempt at a failed module at the first opportunity, you will be required to repeat the module in full. **You will be required to pay the full module fee.**

Repeating a failed module

If you repeat a module, you will have to pay the full module fee when you re-register for the module. When you repeat a failed module you will be allocated a tutor group, you will have access to the learning materials on the VLE and you will be required to resubmit your summative assessment.

5.8

You may choose when you repeat a failed module. You do not have to take the assessment at the next available study session.

6 Progression within the programme

See Section 4: Assessment for the programme for method of assessment.

6.1

You must commence study of the *Principles of Programming* module before, or along with, any other modules. You must have passed 60 credits before you register for the Project. All other modules can be attempted in any order. However, it is recommended that you complete some modules before others as detailed in the advice below.

It is *strongly recommended* that you attempt the assessment of the *Principles of Programming* module before commencing study of the following modules:

Object-Oriented Programming;

Software Design and Programming;

Applied Machine Learning;

Data Management.

It is *strongly recommended* that you attempt the assessment of the *Object-Oriented Programming* module before commencing study of the *Software Design and Programming* module.

Module selection

6.2

In any one study session, you may register for a maximum of 45 credits in a combination of new, failed and resumed modules, of which a maximum of 30 credits may be made up of new modules. A new module is a module you have not registered for previously or for which a previous attempt was invalid.

In a session where you are registered for the Project, this will count as 15 credits per session.

6.3

There are two exam boards a year, following the January and July sessions. You will receive provisional results following the October and April sessions. These results will be formally ratified by the next available exam board. Provisional results should be used for the basis of progression.

Progression between qualifications within the programme

6.4

If you do not already meet the entrance requirements for the MSc or PGDip, successful completion of the PGCert will allow you to progress to the MSc or PGDip. Final results ratified at the exam board will be used as the basis for progression.

6.5

If you are registered on a lower qualification and wish to transfer your registration to a higher qualification, you should notify us before you enter for final assessments for the lower qualification.

As the entrance requirements for the PGDip and MSc are the same, you do not need to successfully complete the PGDip to transfer to the MSc. However, transfer of registration cannot take place whilst a study session is live and before results for this session are ratified by the exam board.

Transfer from standalone individual modules

6.6

You may take up to three modules (45 credits total) on a stand-alone basis (if available) without being registered for the PGCert, PGDip or MSc. If you apply to progress to the PGCert, PGDip or MSc and this is approved, you may be credited with any individual modules successfully completed.

6.7

A mark awarded for completion of an individual module may not be used to replace any mark for a degree, diploma or certificate already awarded.

6.8

If you are registered on standalone individual modules and you wish to transfer your registration to the PGCert, PGDip or MSc, you must meet the entrance requirements for the PGCert, PGDip or MSc.

6.9

Only three modules (a maximum of 45 credits) may be counted as credit towards the MSc, PGDip or PGCert.

If you request to transfer from a standalone individual module to the MSc, PGDip or PGCert and are currently undertaking the study for these modules, transfer of registration cannot take place whilst a study session is live and before results for this session are ratified by the exam board.

7 Schemes of award

If your last assessments take place in the October or January sessions, the date of award will be 1 May in the year of the last assessments that contribute to the award.

If your last assessments take place in the April or July sessions, the date of award will be 1 November in the year of the last assessments that contribute to the award.

Marking criteria

See Appendix C for the Assessment Criteria.

7.1

All assessments will be marked according to the published Assessment Criteria.

Mark scheme

7.2

The following mark scheme is used for the MSc, PGDip and PGCert:

Mark range	Outcome
70% and over	Distinction

Mark range	Outcome
60% – 69%	Merit
50% – 59%	Pass
0% – 49%	Fail

7.3

To calculate the final grade for the qualification, the marks for modules are weighted equally, with the exception of the Project which is double weighted.

7.4

To be granted a qualification with Merit, your mean average mark for the 15 credit modules must be between 60% and 69%; your mark for the Project (if applicable) must be 60% or above.

7.5

To be granted a qualification with Distinction, your mean average mark for the 15 credit modules must be 70% or above; your mark for the Project (if applicable) must be 70% or above.

Date of award

7.6

The date of award will correspond to the year that the requirements for the award were satisfied.

Exit qualifications

7.7

If you have exhausted your permitted number of attempts at module(s) and are unable to complete the MSc or PGDip, you may be considered for an exit qualification of PGDip or PGCert (respectively). In such circumstances, you will need to have achieved the credit required for a PGDip (120 credits) or PGCert (60 credits) and have successfully completed the required modules for the qualification concerned.

Exit qualifications will be classified according to regulations 7.4 and 7.5.

7.8

If you have not completed the required modules, but you have completed the required number of credits for a PGDip (120 credits) or PGCert (60 credits), the Board of Examiners may, at its discretion, consider you for an exit qualification.

7.9

The exit qualification of PGDip or PGCert will be with effect from the year in which you satisfied the requirements for that award. Your registration will cease once the exit qualification has been granted.

Appendix A – Structure of the qualifications

MSc Computer Science

For the qualification of MSc Computer Science, you must pass

- The following ten modules (each worth 15 credits):
    - Applied Machine Learning (CSM010)
    - Cloud Computing (CSM020)
    - Computer Systems (CSM030)
    - Data Management (CSM040)
    - Fundamentals of Computing (CSM050)
    - Information Security (CSM060)
    - Information Systems (CSM070)
    - Object-Oriented Programming (CSM080)
    - Principles of Programming (CSM090)
    - Software Design and Programming (CSM100)
- One compulsory Project (CSM500) (30 credits)

PGDip Computer Science

For the qualification of PGDip Computer Science, you must pass

- Any **eight** modules (each worth 15 credits) chosen from:
    - Applied Machine Learning (CSM010)
    - Cloud Computing (CSM020)
    - Computer Systems (CSM030)
    - Data Management (CSM040)
    - Fundamentals of Computing (CSM050)
    - Information Security (CSM060)
    - Information Systems (CSM070)
    - Object-Oriented Programming (CSM080)
    - Principles of Programming (CSM090)
    - Software Design and Programming (CSM100)

PGCert Computer Science

For the qualification of PGCert Computer Science, you must pass

- Any **four** modules (each worth 15 credits) chosen from:
  - Applied Machine Learning (CSM010)
  - Cloud Computing (CSM020)
  - Computer Systems (CSM030)
  - Data Management (CSM040)
  - Fundamentals of Computing (CSM050)
  - Information Security (CSM060)
  - Information Systems (CSM070)
  - Object-Oriented Programming (CSM080)
  - Principles of Programming (CSM090)
  - Software Design and Programming (CSM100)

Appendix B – Module descriptions

Applied Machine Learning (CSM010)

Machine learning is an important topic in both academia and industry these days. There has been growing interest in the practical side of machine learning. This module focuses more on the practical techniques and methods with Python and Scikit-Learn than on the theories or statistics behind these methods. The module enables students to gain hands-on and practical skills for machine learning based analytics tasks, use appropriate Python libraries and tools to analyse data, and develop the design and programming skills that will help build intelligent artefacts. The module helps students assess the performance of machine learning models and develop a deeper understanding of several real-life topics in applied machine learning, in order to develop the practical skills necessary to pursue research in applied machine learning.

Topics covered:

- Introduction to Python for machine learning
- Preparing data
- Feature selection for machine learning
- Resampling
- Feature evaluation
- Rule-based algorithms: decision tree and random forest
- Regression-based algorithms: logistic regression and neural networks
- Large-scale machine learning using TensorFlow
- Real-life case studies: financial forecasting
- Real-life case studies: computer vision

Cloud Computing (CSM020)

There is an emerging computing paradigm called cloud computing wherein IT-related functions (e.g., storage or database, applications) are provided “as a service” to end-users under a usage-based payment scheme, due to economies of scale and advancements in virtualisation technology. In such a cloud computing model, end-users can hire virtualised services on the fly based on fluctuating requirements (workload pattern, quality of service expectations etc.), and, in doing so, they avoid worry about infrastructure details such as where these resources are hosted or how they are managed. Furthermore, developers with innovative ideas for new Internet services no longer require large capital outlays in hardware to deploy their service or human expense to operate it: instead of buying and maintaining machines, they can just rent computing cycles that are needed. This offers great benefit to IT companies by freeing them from the low-level task of setting up basic hardware and software infrastructures, and thus enabling more focus on innovation and creating business value for their services. It is widely believed that cloud computing is the most significant and disruptive transformation the IT industry has ever undergone.

An information system or software application “in the cloud” typically consists of a front-end and a back-end. The front-end is usually in the form of a light-weight Web or mobile-phone program that can be almost as interactive and functional as traditional desktop software. The back-end normally involves heavy-weight batch jobs that harness the power of tens, hundreds, or even thousands of machines to crunch massive amounts of data. The module introduces the concepts of distributed computing systems, scalable infrastructures as well as development and configuration of complex large-scale applications and systems. Students learn how to develop and deploy modern applications on cloud platforms, such as in Google Cloud Platform and Amazon EC2.

Cloud Computing introduces a variety of modern tools and technologies including the use of virtual machines and containers, the configuration of distributed systems, deployment and understanding of operations of NoSQL systems, development of RESTFul services with Python, understanding of DevOps practices and infrastructure as a code and use of big data processing systems such as Hadoop MapReduce and Apache Spark.

Topics covered:

- Cloud computing technology
- Cloud services and Virtualization
- Web services, REST and authorization protocols
- Using Python frameworks to develop APIs with Django
- Distributed and parallel systems with Python
- Cloud data storage systems and NoSQL systems
- DevOps and Container systems
- Service-Oriented Architectures
- Distributed systems configuration
- Introduction to Big data and Hadoop-Map Reduce framework

Computer Systems (CSM030)

This module takes a programmer's view of how computer systems execute programs, store information, and communicate. Its purpose is to enable students to become more effective programmers with the ability to deal with performance, portability and robustness challenges using an in-depth understanding of hardware and operating system capabilities and their constraints. The material covered in this module provides the foundation for students to delve deeper into key elements of the software engineering toolkit and offers a variety of methods and techniques to employ in solving real-world problems.

The main aim of the module is to learn how computers work: the basics of computer architecture and organization, and the role and mechanism of operating systems. To learn the basics of computer architecture and organisation, and the role and mechanism of operating systems. Specifically, students are introduced to three key aspects of computer systems: storage, processing and transmission of information.

Topics covered:

- Introduction: history of computers, main parts of computers, fundamental concepts of computing.
- Processors: main components of a CPU, the fetch-execute cycle, instruction sets.
- Processes and threads: implementation, scheduling, inter-process communication, synchronisation.
- Deadlocks: detection and recovery, avoidance, prevention.
- Memory storage: cache, internal and external memory.

- Memory management: allocation and protection, virtual memory with paging and segmentation.
- Input-Output systems: bus architecture, interrupt mechanism, OMA, device drivers, disk scheduling algorithms.
- File systems: access and allocation methods.
- Protection and Security.
- Multiple processor systems.

Data Management (CSM040)

This module covers the principles and application of data and knowledge management technologies and languages including SQL. Students study the use of these in leading commercial database management systems as well as emerging approaches to data management. The module also examines the technologies underlying modern data management systems. It studies advanced aspects of query processing, transaction management, distributed data management, and recent developments in web data, big data and alternative database architectures. Data and knowledge management represent a core technology in the theory and application of Computer Science. Students study the advanced aspects of databases and recent advances in data management technologies in three major directions: performance, distribution of data, and heterogeneity of data.

Topics covered:

- Database management software: origins and objectives.
- The relational model: algebraic and logical foundations.
- SQL: data manipulation, host language support for SQL.
- Transaction management, recovery, concurrency.
- Relational database theory: dependencies, normal forms.
- DBMS architectures and implementations.
- DBMS storage and indexing.
- Query optimization.
- Enhanced database capabilities: procedural extensions to SQL, database triggers, deductive databases.
- Non-relational DBMS, object databases, NoSQL databases.

Fundamentals of Computing (CSM050)

Discrete mathematics, mathematical logic, and the related fundamental areas of data structures and algorithms lie at the heart of any modern study of Computer Science. Any understanding of how computers operate and how to use them effectively and efficiently, in terms of either their hardware or software, inevitably involves numerous mathematical concepts. This module introduces and develops mathematical notions, data structures and algorithms that are used in various areas of Computer Science, in particular those required for other modules of the proqramme.

The module aims to: introduce the notation, terminology, concepts and techniques underpinning the discipline of Computer Science; promote the importance of formal notations as the necessary means of ensuring clarity, precision, and absence of ambiguity; provide an introduction to the concepts and manipulation of the basic finite structures as these arise in Computer Science, e.g., computer arithmetic, strings, graphs, sets, digital circuits, lists, binary trees; introduce basic models of computation, such as finite automata and Turing machines; give students an understanding of the fundamentals of data structures and file organisation: their representation, algorithms for their operation, and the relative merits of different structures and methods; and introduce the design and analysis of algorithms, and their efficiency/complexity.

Topics covered:

- Numbers: integer, rational, and real. Numeral systems
- Arithmetic for computers
- Digital logic (combinational circuits)
- Elements of set and graph theories
- Finite state machines (automata) and regular languages
- Turing machines
- Data structures: representations and operations
- Lists, trees, forests, binary trees
- Tree traversal and other operations; binary search trees
- Sorting and searching

Information Security (CSM060)

Information security is about protecting information (and information systems) against unauthorised access and tampering. Avoiding security breaches has a high priority for organisations storing and handling confidential data. Large amounts of confidential information are stored in today's information systems. Breaches of security can have dire consequences for the organisations running these systems, therefore it is very important to manage the risks associated with security-related issues. In recent years the topic of information security has played a bigger and bigger role. Consequently, everyone involved in the management of information systems should have at least some basic knowledge of it.

The main aim of this module is to provide broad coverage of the field of information security. This course covers the technical as well as the management side of security information systems. Despite being an essential part of security, technical methods such as cryptography are not enough to guarantee a high level of security. They have to be embedded into a wider context in order to make them more effective. Users of technology have to understand the underlying principles and follow certain policies to avoid security breaches. This module introduces the fundamental approaches to security engineering and includes a detailed look at some important applications.

Topics covered:

- Overview of Information Security
- Security Policies
- Social Engineering
- Basic Cryptography
- Identity Management
- Access Control Mechanisms
- Assurance and Trust

- Network Intruders and Intrusion Detection
- Firewalls and Malicious Software
- Economics of Information Security

Information Systems (CSM070)

The module describes approaches, processes, methodologies and techniques commonly used for large-scale information systems development. It covers the systems development life cycle (SDLC), including project initiation, analysis, design and implementation, addressing key aspects and techniques at each stage. Project methodologies are described, with an emphasis on the Scrum methodology. The module also incorporates insights into professional and legal issues associated with EIS development.

The primary aim of the module is to describe enterprise information systems (EIS) and to set out the considerations and approaches used to implement (deploy) these systems in the business enterprise. This covers predominantly the Systems Development Life Cycle (SDLC) and the various methodologies used to formalise it, including waterfall and agile approaches, with particular emphasis on the Scrum method. In the course of this module students are introduced to a range of topics relevant to EIS deployment and the SDLC, including object-orientation, the Unified Process and Universal Modelling Language (UML), enterprise architecture and technical architecture.

Alongside describing the SDLC, students will be introduced to practical aspects associated with a career as an IS professional, and social and organisational aspects of enterprise computing. This will include topics such as Intellectual Property, Digital Surveillance, Data Privacy and Ethical issues in computing.

Topics covered:

- Introduction to Enterprise Information Systems (EIS)
- SDLC, IS project methodologies and the Unified Process
- Unified Process – Planning & Analysis
- Scrum I – Process, Roles, Activities & Ceremonies
- Scrum II – Artefacts & Concepts
- Enterprise Architecture & Technical Architecture
- EIS Implementation and Operation
- GDPR, Freedom of Information & Intellectual Property Rights
- Contracts & Business Planning
- Computer Misuse, Digital Surveillance and Ethical Issues in Computing

Object-Oriented Programming (CSM080)

The module further develops the core software engineering skills and knowledge following on from the Principles of Programming module as a key ingredient for students pursuing a qualificiation in Computer Science. In particular, this module discusses issues specifically related to developing programs for large programming projects and for modern computer hardware architectures.

This module covers object-oriented programming, including the use of subclasses, modules, and library classes to create well-organised programs. The module enhances student’s understanding of making appropriate choices on the selection of algorithms, their implementation together with the required data structures (e.g. arrays, lists, trees, graphs, depth- and breadth-first search algorithms). The module enables students to develop programs for modern multi-core architectures utilising functional programming constructs.

Topics covered:

- Transition to Object-Oriented, including types, encapsulation, inheritance, polymorphism, and message passing
- Static typing and reference types
- Further recursion and memoisation
- Local I/O
- Generics
- Style rules
- Concurrency
- Further test-driven development through unit testing and JUnit
- Lambdas and Streams
- Programming in teams

Principles of Programming (CSM090)

This module introduces programming concepts and techniques, as well as elementary software development principles. Both for absolute beginners and for those with prior programming experience, the module introduces the fundamentals of programming, including: variables and assignment, primitive and complex types, methods, control structures, collections, iteration, recursion, as well as classes and objects in object-oriented programming. The module also introduces basic software development issues such as design, testing, debugging. The module provides the student with a comprehensive grounding in programming and familiarises students with a modern programming language such as Python.

Topics covered:

- Core imperative programming ideas: sequence, iteration, assignment, and variables
- Data types
- Collection Data Structures: Arrays, List, Sets, Dictionaries
- Functions
- Version Control
- Automated testing and test-driven development (TDD)
- Functional programming features of Python
- Dynamic data structures: Linked lists, Queues, and Stacks
- Recursion
- Exception handling

Software Design and Programming (CSM100)

The main aim of the module is to provide students with the necessary skills to design software in an object-oriented way according to high quality standards. This module provides students with the necessary skills for developing software using object-oriented and functional programming paradigms. This ranges from learning object-oriented concepts, designing object-oriented software using a proven methodology and tools, to learning how to program in an object-oriented and functional style. The module provides a detailed examination of Software Design Patterns and the emerging functional features of current day object-oriented programming languages.

Topics covered:

- The object model and how it is realised in various object-oriented languages (e.g., Kotlin, JavaScript, C#, Java, Scala, Swift, ...)
- Further development of the ideas of inheritance, polymorphism, and abstraction
- Language features nested classes, closures, higher-order functions, meta-objects, etc.
- The functional paradigm. Abstract data types, polymorphic types, static typing and type inference. Recursion and induction. List processing. Higher-order functions. Eager and lazy evaluation. Imperative features. Signatures, structures, functors, type classes, monads
- An introduction to Test Driven Design (TDD) and Behavioural Driven Design (BDD)
- The use of an Integrated Development Environment (IDE) for software development: e.g., editing, debugging, compilation, etc.
- Modularity, versioning, packaging, and managing the build process
- Design Patterns and Anti-Patterns and their application to software design
- The SOLID (Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion) approach to object-oriented programming and design
- Code refactoring and analysis

Project (CSM500)

To consolidate the learning achieved during the ten modules of the MSc programme, students will undertake a work-based project. The project will draw on elements of learning from different parts of the programme and demonstrate students' insight into, and understanding of, software engineering and computer science in the context of their addressing the needs of a client organisation.

Examples of projects include:

- An electronic commerce website for a family business following digitisation
- A mind-mapping native application tool for a major operating system such as Linux, Windows or Mac OS designed for the specific requirements of a school
- A fundraising app for a charity for one of the major mobile OSs such as Android and iOS
- A data analytics tool for the analysis of mobility patterns collected from a geo-location service for a local delivery business
- The back-end service and REST API providing access to a legacy knowledge base operated by a client organisation

The module aims to: provide an academic structuring for a client-facing research project; support students in negotiating the complexity of undertaking such a project, using appropriate methodology; develop students' ability to function as independent software engineering professionals; enable students to collect, record and analyse software requirements in the context of a problem in a specifically-focused application domain; develop students' ability to design a software solution and fully document their design; demonstrate the ability of students to carry out all the phases of the software engineering process; and support students in the recognition of structural factors and uncertainties and the ability to work effectively with these.

The module offers students the opportunity to: develop a systematic understanding and critical awareness of an agreed problem in the area of data science; plan and execute a major piece of programming work appropriate to the MSc programme, critically present existing approaches in the problem area, position their own approach within the area and evaluate their contribution; and gain experience in communicating complex ideas/concepts and approaches/techniques to others by writing a comprehensive, self-contained report.

The main part of this module will be completed by the student on their own. There is a small taught component of the module in which students are acquainted with:

- How to formulate the objectives and aims of an MSc project.
- How to write a project proposal.
- How to organise and plan the project.
- How to undertake a literature review.
- How to write a project report.

During self-study, students cover the following topics:

- Identifying a client and their needs
- Preparing a project proposal
- Writing a critical literature review
- Document client requirements
- Designing a software solution and documenting the design using a formal methodology
- Selecting an appropriate software methodology process that can deliver this solution
- Identifying data protection and ethical issues
- Providing a comprehensive implementation of the software design developed
- Testing the developed software using an appropriate methodology and establishing that it is fit for purpose
- Organizing and presenting a report on the work undertaken

Appendix C – Assessment criteria

Coursework and Project Assessment Criteria

This is an indicative description of expectations at each grade level. Overall grades will comprise qualitative and quantitative elements. The setting of questions, tasks and requirements and the accompanying marking scheme should take account of the criteria below.

Mark band	Assessment criteria
80%-100% High Distinction	Marks in this range indicate an exceptionally high level of scholarship and outstanding performance in terms of all of the dimensions outlined. While work at this level exhibits scrupulous completion of the requirements of the assignment, it will also exhibit a high degree of initiative, high quality of analysis, academic sophistication, comprehension and critical assessment, making a novel contribution to computer science studies.
70%-79% Distinction	Marks in this range indicate high levels of scholarship, and high performance in terms of all of the dimensions outlined. Comprehensively argued writing of interest and originality which is also well organized and presented exhibiting a sound, critical and analytical grasp of the relevant literature(s) and drawing on an extensive range of relevant academic sources. The work will display an excellent understanding of underlying theory as well as employing appropriate analytical techniques, resulting in an argument of interest and significance.
60%-69% Merit	Work that demonstrates a good command of the subject and relevant literature(s) as well as a sound grasp of critical issues, with evidence of independent thought and a high standard of argument as well as good presentation. Work towards the bottom of this range may have occasional weaknesses and flaws but will nevertheless show a generally high level of competence. Work towards the top of this range will be highly competent on all dimensions.
50%-59% Pass	Marks in this range indicate general capability, but with moderate levels of weaknesses on one or more dimensions indicated above. Work in this range may contain inaccuracies, the arguments may lack clarity or rigour, or there may be a lack of critical understanding. It will however be coherently structured and presented, showing a sound command of the subject, some awareness of critical debate, and the ability to construct a generally coherent argument.
40%-49% Fail	Marks in this range do not quite meet the minimum standards for a pass, with considerable levels of weaknesses on one or more dimensions. Work in this range may suffer from flawed arguments, weak structure and presentation, an inadequate command of course materials, or a serious failure to reflect on those materials. It will however demonstrate a basic understanding of computer science studies and show evidence of reasonable attention to the course materials.
30%-39% Low Fail	Marks in this range display major levels of weaknesses on two or more dimensions. The work may be reliant on a minimal range of reading and reflection with poor attention to detail. Work in this range may be characterised by assertions lacking supporting evidence or argument, or by seriously flawed understanding of key concepts.
0%-29% Very Low Fail	Marks in this range indicate general incompetence, with highly serious levels of weaknesses on two or more dimensions. Work in this range will either fail to present any real argument or opinion, or fail to engage at all with the topic in question. Work may quote heavily from a small number of sources, but fail to integrate them and provide little or no narrative to explain their relevance.