Programme Specification
2023–2024

Computer Science

MSc
PGDip
PGCert
Individual modules

Important document – please read
Contents

Important information regarding the Programme Specification ........................................ 2
Programme title and qualifications.................................................................................... 3
Entrance requirements .................................................................................................... 6
Educational aims and learning outcomes of the programmes......................................... 9
Learning, teaching and assessment strategies................................................................ 11
Assessment methods .................................................................................................... 12
Student support and guidance........................................................................................ 13
Quality evaluation and enhancement............................................................................. 13
After graduation............................................................................................................. 15
Appendix A – Structure of the qualifications ................................................................. 16
Appendix B – Module descriptions................................................................................. 18
Important information regarding the Programme Specification

About this document
Last revised 31 May 2023

The Programme Specification gives a broad outline of the structure and content of the programme, the entry level qualifications, as well as the learning outcomes students will achieve as they progress. Some of the information referred to in this programme specification is included in more detail on the University of London website. Where this is the case, links to the relevant webpage are included.

Where links to external organisations are provided, the University of London is not responsible for their content and does not recommend nor necessarily agree with opinions expressed and services provided at those sites.

For queries about any of the programme information provided, whether here or on the website, registered students should use the ‘ask a question’ button in the student portal. Otherwise, the Contact Us link at the bottom of every webpage should be used.

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

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

Key revisions made
Programme specifications are revised annually. The quality committee of the member institution providing academic direction, as part of its annual review of standards, confirms the programme structure and the educational aims and learning outcomes, and advises on any development in student support. Where there are changes which may impact on continuing students, these are listed below. For all new students, the programme and general information provided in this document is correct and accurate and will be applicable for the current year.

Significant changes made to the programme specification 2023-2024
Maximum periods of registration for qualifications are now included in the General Regulations.

- For students initially registered from 2023-24 onwards, the PGDip maximum period of registration is now 4 years, reduced from 5 years.
- For students initially registered from 2023-24 onwards, the PGCert maximum period of registration is now 2 years, reduced from 5 years.
- Students will retain the period of registration initially given to them on registration if the maximum registration period for their qualification changes during their studies.

The ‘Internet access and computer specification’ section has been updated to reflect minimum computer requirements.
Programme title and qualifications

Postgraduate Degrees of the University of London may be classified. The award certificate will indicate the level of the academic performance achieved by classifying the award. The classification of the degree will be based on the ratified marks from the completed assessments.

The classification system for the MSc, Postgraduate Diploma and Postgraduate Certificate in Management is as follows:

Distinction; Merit; Pass.

Specific rules for the classification of awards are given in the Programme Regulations, under Scheme of Award.

Programme title

Computer Science

Qualifications

Master of Science in Computer Science
Postgraduate Diploma in Computer Science
Postgraduate Certificate in Computer Science

Intermediate qualifications

Students may not normally request a lower intermediate qualification if studying on a higher qualification (except as an exit qualification) or accumulate these qualifications as they progress from lower to higher qualifications.

Exit qualifications

Postgraduate Diploma in Computer Science
Postgraduate Certificate in Computer Science

An exit qualification is an intermediate qualification, as noted above, for which the student may not have registered at the outset but which may be awarded on completion of specific modules/courses (or credit accumulated) in a longer programme of study, if the student leaves the programme. Exit qualifications are awarded at the discretion of the Board of Examiners and once a student has accepted an exit qualification they will not normally be permitted to continue their study of the same programme with the University of London.

Individual modules available for study on a stand-alone basis

There is also provision for select individual modules of the programme to be studied on a stand-alone basis without being registered for a related qualification, with the exception of the Project. Only three modules (a maximum of 45 credits) may be counted as credit towards a related qualification. Neither progression nor credit is automatic.
Award titles may be abbreviated as follows:
Master of Science – MSc
Postgraduate Diploma – PGDip
Postgraduate Certificate – PGCert

Level of the programmes


The awards are placed at the following Levels of the Framework for Higher Education Qualifications (FHEQ):
- MSc Level 7
- PGDip Level 7
- PGCert Level 7

Relevant QAA subject benchmarks group

See the QAA website for information about quality assurance.

Computing

Awarding body
University of London

Registering body
University of London

Academic direction
Birkbeck, University of London

Accreditation by professional or statutory body
As part of the development of the MSc Computer Science programme, requirements set out by a number of relevant professional bodies are being reviewed.

Language of study and assessment
English

Mode of study
Web supported learning with an online tutor.

Programme structures
The programme has two registration points in the year: October and April. There are four study sessions in a year, each lasting 10 weeks for each 15-credit module. The 30-credit Project module runs over two 10-week study sessions. Study sessions begin in October, January, April and July. Each session is following by an assessment submission point.
Students have an online induction session available through the virtual learning environment (VLE) prior to the start of their study session. This includes orientation of their learning environment and guidance on the structure and learning expectations for the module.

The **MSc Computer Science** is a 180 UK credit degree programme consisting of:
- Ten compulsory modules (15 credits each): and
- One compulsory Project (30 credits)

The **PGDip Computer Science** is a 120 UK credit degree programme consisting of:
- Eight modules (15 credits each)

The **PGCert Computer Science** is a 60 UK credit degree programme consisting of:
- Four modules (15 credits each)

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

**Maximum and minimum periods of registration**

The minimum periods of registration from a student’s effective date of registration, are:

<table>
<thead>
<tr>
<th>Qualification</th>
<th>Minimum*</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSc</td>
<td>Two years</td>
</tr>
<tr>
<td>PGDip</td>
<td>One and a half years</td>
</tr>
<tr>
<td>PGCert</td>
<td>Six months</td>
</tr>
</tbody>
</table>

*The minimum period of registration is subject to module availability.

See the [General Regulations](#) for the maximum periods of registration for these qualifications.

Students will retain the period of registration initially given to them on registration if the maximum registration period for their qualification changes during their studies.

In making a decision as to how many modules to register for in a given session, it is important to take account of on-going work and/or personal commitments.

**Credit value of modules**

Further information about the credit systems used by universities in the UK and Europe is provided by the [Quality Assurance Agency](#) and the [European Credit Transfer and Accumulation System](#).

Where credits are assigned to modules of a programme, credit indicates the amount of learning carried out in terms of the notional number of study hours needed, and the specified Framework for Higher Education Qualifications in England (FHEQ) credit level indicates the depth, complexity and intellectual demand of learning involved. The details below indicate the UK credits and the European Credit Transfer and Accumulation System (ECTS) values.

The MSc Computer Science comprises a total of 180 UK credits (90 ECTS credits) at FHEQ level 7.
The PGDip Computer Science comprises a total of 120 UK credits (60 ECTS credits) at FHEQ level 7.

The PGCert Computer Science comprises a total of 60 UK credits (30 ECTS credits) at FHEQ level 7.

One UK credit equates to a notional ten hours of study.

Each 15-credit module equates to 150 hours. Over the 10 teaching weeks of a study session, students will need to dedicate around 15 hours of study per module per week.

The 30-credit Project module equates to 300 hours. Over the 20 teaching weeks of the study sessions, students will need to dedicate around 15 hours of study per week.

**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 at an institution other than the University of London. Where the prior learning covered a similar syllabus to a module/course on the University of London programme, credit will be awarded as if the student took the University of London module/course.

See the [General Regulations](#) (Section 3) and [Programme Regulations](#) for more rules relating to prior learning.

MSc and PGDip students may apply for recognition of prior learning mapped against modules to a total of 60 UK credits. Applications for recognition of prior learning for the Project will not be accepted. PGCert students may not apply for recognition of prior learning.

**Entrance requirements**

Applicants must submit an application in line with the procedures and deadlines set out on the website.

**Entrance requirements for the MSc and PGDip**

To qualify to register for the MSc or PGDip, applicants will need a bachelor’s degree which is considered at least comparable to a UK second class honours degree from an institution acceptable to the University.

**Entrance requirements for the PGCert**

If applicants do not meet the MSc and PGDip entrance requirements, successful completion of the PGCert will allow progression to the MSc or PGDip.

To qualify to register for the PGCert, applicants will need:

*Either*, a bachelor’s degree which is considered at least comparable to a UK second class honours degree from an institution acceptable to the University;

*Or*, a minimum of two years’ work experience in a relevant field. This will most commonly be in a software engineering role but each application will be considered on a case-by-case basis.

**Entrance requirements for stand-alone individual modules (if available)**

To qualify to register for a stand-alone individual module, applicants will need:
Either, a bachelor’s degree which is considered at least comparable to a UK second class honours degree from an institution acceptable to the University;

Or, a minimum of two years’ work experience in a relevant field. This will most commonly be in a software engineering role but each application will be considered on a case-by-case basis.

**English language requirements**

All applicants must satisfy the English language requirements for the programme. These are set out in detail on the programme page under [Entry Requirements](#).

Applicants who have not met any of the above must have passed, within the past three years, a test of proficiency in English language, from an organisation acceptable to the University. This standard must be reached before registering for the MSc.

Further information on English language proficiency tests can be found on our [website](#).

**Internet access and computer specification**

Students will require regular access to a portable computer with an internet connection to use the University of London’s online resources and systems.

Students must be able to download and install software to their Windows or MacOS device to include secure examination browsers for online assessment purposes (if offered on their programme of study). Depending on the security settings for each assessment, students may be required to have full administrator rights on their computer to install and run the software needed to take part in the assessment. Full administration rights are likely to apply to a computer that they own but not to one provided by their employer, for example.

The portable computer must have at least the following minimum specification:

- Windows: 10 and 11 on 64-bit platforms
- Mac OS X 10.15 and higher (for some assessments macOS Big Sur (version 11) as a minimum)
- CPUs newer than 2011 (Intel Sandy Bridge or newer)
- OpenGL 2.0 graphics driver
- Local storage for the recording of proctored examinations (75MB per hour)
- Web camera & microphone (internal or external)
- A broadband internet connection capable of streaming and a minimum of 0.15Mbps upload speed
- Minimum device requirements are subject to change and older operating systems may become obsolete over time.

and the following applications installed:

- a word processor that accepts Microsoft Word formats (.doc and .docx)
- a PDF reader (e.g. Adobe)
- Microsoft Excel which can run macros
- a presentation program that supports Microsoft PowerPoint
Additional hardware capabilities and software for specific modules will be required, such as:

- Video and audio recording.
- Programmer’s text editor such as Atom or Visual Studio and an integrated development environment software (IDE) such as IntelliJ.
- Additional software: including software development kits for example Anaconda for python and the Java SDK, databases such as Mysql, web server such as Apache Tomcat and network analysis tools such as Wireshark.
- Access to hardware: Android or iOS device; microcontrollers such as Arduino; and additional networking elements such as routers and protocol service endpoints.

Finally, students will be required to create and use free accounts on several online services including but not limited to github, Amazon Web Services Educate, Google Cloud Engine and Google Collaborate and Oracle Academy.

**Please note:** Full mobile access is not available for all programmes. Proctored assessments will not work on any smartphone, tablet, Chromebook, Linux Operating Systems or other mobile device of any kind.

**Students with specific access requirements**

The University of London welcomes applications from disabled students and/or those who have access requirements. The University will make every effort to provide reasonable adjustments to enable those with a disability, learning difficulty or access requirements to have the same opportunity as all other students to successfully complete their studies.

The University is committed to managing the application procedure and the programme itself to ensure that services are accessible for all students and that an inclusive environment is created. Students with a disability, or others who may need access arrangements to assist in taking examinations, should complete the relevant section of the application form, or contact the [Inclusive Practice Manager](#). A separate room or other arrangements may be considered.

Requests are considered by a University panel, whose purpose is to ensure that students with disabilities and/or specific access requirements are neither advantaged nor disadvantaged by such arrangements when compared with other students. These considerations remain separate from the academic selection processes.

For further information, see [Inclusive Practice Policy](#)

**Sources of funding and scholarships**

Information about potential sources of funding and scholarships is updated annually and where available is included in the prospectus web pages.

For further information see the [website](#).
Educational aims and learning outcomes of the programmes

Programme aims

The MSc Computer Science provides an intensive course in computing for graduates of other subjects.

As well as gaining a broad knowledge of the subject, students acquire practical skills and have the opportunity to investigate certain areas of current research more deeply. For students who are new to the subject, it provides a foundation for a career in IT; for those already working in IT, it provides an opportunity to broaden their knowledge and update their skills while obtaining a formal qualification.

Holders of the MSc will have demonstrated:

- A systematic understanding and a critical awareness, much of it at the forefront of the discipline
- A comprehensive and practical understanding of applicable techniques
- Originality in the application of knowledge
- The ability to evaluate current research and methodologies
- The independent learning ability required for continuing professional development.

Learning outcomes (MSc Computer Science)

A student is expected to:

A. Knowledge and understanding:

- Demonstrate a knowledge of:
  - core programming principles including variables, statements, data structures, and functions.
  - mathematical and algorithmic foundations of computing and data management
  - the concepts related to object-oriented and functional programming
  - information and network security, and data protection policies and regulations
  - the legal, ethical, and social implications of information system design decisions
  - database architecture, design, storage, and management
  - software engineering, design, testing, and deployment of small and large scale software applications
  - computer architecture and operating systems including hardware components, and the querying of data stored temporary in-memory or long-term through data storage and management
  - local network and distributed systems architecture, and their associated communication protocols
  - machine learning architectures, methods, and techniques
  - information modelling in computing and their application to different types of data and processes
o commercial, economic, and social implications of computing applications and techniques

o the fundamental models of computation, computer arithmetic and logic

o Information System and Information Technology career paths

B Cognitive skills

- Demonstrate the ability to:

  o apply algorithmic thinking to solve a specific task and to convert this into executable program code

  o identify and correct both logical and syntactic bugs in a software program for a specific programming language

  o design, implement, and deploy principles of database and information systems

  o demonstrate self-direction, creativity, and originality in tackling and solving computational problems, including at scale

  o apply reasoning through abstract concepts and demonstrate abstraction skills in applying computational methods to solve problems

  o plan, execute, and complete a substantial project involving independent study over several months

  o act autonomously in planning, solving, and implementing computational tasks at a professional level

  o conduct a critical analysis of professional articles, and research papers

  o communicate design decisions, approaches, and solutions with clarity to both specialist and non-specialist audiences

  o identify the social, legal, ethical, and organisational implications of the use of computers

  o compare and evaluate a range of computational methods and techniques to analyse data and information systems and identify the benefits and short-comings of each approach with confidence

  o recognise the efficiency and resource requirements of computational techniques, including the organisational and environmental factors

C Practical and professional skills

- Demonstrate the ability to:

  o implement computer code in a programming language including Python, Java, and C#

  o implement a computer program in an object-oriented programming language to perform a calculation from a software specification.

  o implement the SQL data-manipulation language to create, query, and manipulate data

  o create and document a design using UML approaches and software tools

  o use a coherent information system development process to solve a problem
o gain experience of applying network and cloud computing technologies
o implement appropriate algorithms and data structures in sequential and parallel environments
o apply allocation techniques in software design and implementation to manage computational resources effectively
o compare different approaches and make informed decisions on computational model and system architecture selection to solve a problem
o manage learning and development, including time management and organisational skills.
o act autonomously in planning and implementing tasks at a professional level
o discuss and present findings of research, analysis, and evaluation
o develop an Information System/Information Technology project plan and explain the risks inherent in developed plans

Learning outcomes (PGDip Computer Science)

Students who are granted the PGDip will be expected to have passed eight 15-credit modules from a choice of ten (120 credits total). As such, students obtaining this qualification should have gained sound understanding of the learning outcomes listed above for the MSc as relevant for the modules chosen.

Learning outcomes (PGCert Computer Science)

Students who are granted the PGCert will be expected to have passed four 15-credit modules (60 credits total) from a choice of ten. As such, students obtaining this qualification should have gained sound understanding of the learning outcomes listed above for the MSc as relevant for the modules chosen.

This specification provides a concise summary of the main features of the programme and the learning outcomes that a typical student might reasonably be expected to achieve and demonstrate if they take full advantage of the learning opportunities provided. More detailed information on the specific learning outcomes, content and the learning, teaching and assessment methods of each module can be found in the module syllabuses.

Learning, teaching and assessment strategies

The core principles of the Standard Academic Model, which incorporates the learning, teaching and assessment strategy for this programme, are outlined below.

Principle 1: Access and opportunity

Our programmes are designed to maximise access and opportunity for learners. An online induction will ensure that students are prepared for study and are familiar with the learning environment and sources of support during their student journey.

Principle 2: Student learning

Our programmes are designed to provide students a rich, vibrant, rigorous and engaging learning experience.
Principle 3: Student support

Our programmes are designed to provide students with the support they need to succeed and fulfil their potential.

Principle 4: Programme and module structure

Our programmes follow a consistent structure, designed to drive progression and retention, and offer flexibility within an efficient module release schedule.

Principle 5: Design for learning

Our programmes are a co-creation between various teams who each contribute their professional expertise, and educational design for successful learning is at the heart of the programme development process.

Principle 6: Unbundling, portability

Wherever possible, our modules are designed as discrete sequences of learning for maximum use and portability.

Principle 7: Assessment

Our assessment is rigorous, robust, secure, transparent and aligned to the learning outcomes.

Principle 8: Equality, Diversity and Inclusion

Inclusion is at the heart of our practice in teaching, learning and assessment.

Assessment methods

Each module is run over a 10-week block, with the exception of the Project which is run over two 10-week blocks.

The following modules are assessed by two assessments: one online test of a suitable type for the specific module, for example, multiple choice questions (MCQs) or auto-graded problem-solving task using programming (25%) and an end of term coursework assessment/online examination (75%):

- Applied Machine Learning
- Cloud Computing
- Object-Oriented Programming
- Principles of Programming
- Software Design and Programming

The following modules are assessed by an end of term coursework assessment/online examination (100%):

- Computer Systems
- Data Management
- Fundamentals of Computing
- Information Security
- Information Systems
An examination is defined as an element of assessment that takes place in a controlled environment. All examinations are scheduled using an online delivery method. Students will be given details of how the modules on their programme are assessed, the specific environment or location that is permitted and the time allowed for the assessment. Students must ensure that their device is kept up to date and complies with University Computer Requirements.

The Project is assessed by two assessments: a Research Proposal (30%) and a Project Report (70%).

Where a module has more than one element of assessment, the grade awarded for each module will be based on both elements of assessment. Students will be required to pass the module with a weighted average of 50%.

Repeating or resitting a failed module may be done once the module results have been confirmed by the Board of Examiners.

Coursework is submitted in the VLE by prescribed deadlines.

Student support and guidance

Key features of the support for students include:

- **Student Portal**: for accessing student induction, study skills support, careers and employability resources, student wellbeing advice.

- Student induction resources.

- **Student Guide**: This provides information which is common to all students and gives guidance on a range of issues from the start of a student’s relationship with the University of London through to their graduation.

- VLE containing: self-assessment and student planner tools; comprehensive learning materials; e-resources/e-library; student forums and progress monitoring tools

- Online student advisor and online tutor

- **Programme Regulations**.

- **The Online Library**: This provides a range of full-text, multidisciplinary databases where journal articles, book reviews and reports can be found.

- A University of London email account and web area for personal information management.

Quality evaluation and enhancement

The University of London delivers the majority of its flexible and distance learning programmes through a collaboration between the University of London Worldwide and federation members of the University of London. However, some of the flexible and distance learning programmes draw solely on academic input from the University of London, and are delivered without academic lead by a federation member. The policies, partnerships (where applicable) and quality assurance mechanisms applicable for the programmes are defined in the following key documents: The Quality Framework, the **Quality Assurance Schedules, Guidelines for Examinations, General Regulations** and, for each programme, **programme specific regulations**.
Awards standards

All University of London qualifications have to align with the Frameworks for Higher Education Qualifications of UK Degree-Awarding Bodies to assure appropriate standards for each qualification. In addition, every programme that is developed by a federation member of the University of London (or a consortium with representation by more than one federation member) will be developed to the same standard as would be applied within the institution concerned. Proportionate and robust approval procedures, including external scrutiny and student engagement are in place for all programmes. Learning materials are written and all assessments are set and marked by academic staff who are required to apply the University’s academic standards.

Review and evaluation mechanisms

Some of the key mechanisms in place to assure the standards of all University of London qualifications and the quality of the student experience, include:

- Annual programme reports: produced for all programmes in order to review and enhance the provision and to plan ahead;
- Independent external examiners: submit reports every year to confirm that a programme has been assessed properly and meets the appropriate academic standards;
- Annual student information statistics: prepared for all systematic reporting within the University of London;
- Periodic programme reviews: carried out every 4-6 years to review how a programme has developed over time and to make sure that it remains current and up-to-date.

Improvements are made as necessary to ensure that systems remain effective and rigorous.

Student feedback and engagement

The principal channel for collecting feedback from students is the Student Experience Survey. Carried out every year, this collects feedback from the student body on a range of topics relating to the student lifecycle. The results are analysed externally and then considered in a number of different ways, including by the programme team, principal committees and the senior leadership team. Details of any resulting actions taken are published on the Virtual Learning Environment and the Student Portal.

Additionally, on completion of their programme of study students will be invited to take a survey that seeks to measure what they have gained from their studies.

There are also opportunities for students to get involved in governance. An undergraduate and postgraduate student member is appointed by the University to the majority of committees through an annual appointment round. Some programmes also recruit student members at the programme level. Students are frequently invited to take part in quality review processes such as Periodic Programme Reviews, Programme approval, Thematic Reviews, MOOC review panels and ad hoc focus groups. Opportunities such as these are advertised through social media and on the website. More information can be found on the website.

Students can also apply to join the Student Voice Group, which meets four times a year to consider initiatives for enhancing student experience. Notes from these meetings are published on the Student Portal.
After graduation

Further study
Successful completion of the programme may serve as preparation for students who wish to go on to take further study in the subject area. Enquiries about further study opportunities should be directed to the University of London Student Advice Centre 'ask a question' button in the student portal.

Graduate employment routes
At registration, students are asked to complete a small number of questions relating to where they are in their career planning. This data is used to assist with the implementation of employability strategies and interventions.

This programme is targeted at career minded individuals who wish to develop their professional skills with master's level understanding and skills. Possible career paths include different software engineering roles ranging from testing and quality assurance to software architecture, devops and full-stack engineering. This degree can also be useful in becoming a systems analyst, engineering manager, or other role that requires an in-depth understanding of software design and practice in a range of sectors or organisations.

The Alumni Community
Upon finishing a course of study, graduates automatically become part of the University of London alumni community, a diverse global network of more than one million graduates in over 180 countries, providing lifelong links to the University and to each other.

Alumni are encouraged to keep in touch after they graduate and to become active members of the alumni community; in return they receive a number of benefits and services, including an extensive programme of events and engagement opportunities.

More information is available on the alumni webpage.

Follow the alumni community on social media: Facebook, Instagram, LinkedIn
Appendix A – Structure of the qualifications

**MSc Computer Science**

For the qualification of MSc Computer Science, students 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, students 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, students 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,
Programme Specification 2023-2024 Computer Science (MSc, PGDip, PGCert)

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 programme.
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
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 qualification 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
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.
- 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