Swansea University
Prifysgol Abertawe

FACULTY OF SCIENCE AND ENGINEERING

POSTGRADUATE STUDENT HANDBOOK

MSC (FHEQ LEVEL 7)

ADVANCED SOFTWARE TECHNOLOGY

DEGREE PROGRAMMES

SUBJECT SPECIFIC
PART TWO OF TWO
MODULE AND COURSE STRUCTURE
2022-23

Welcome to the Faculty of Science and Engineering!

Whether you are a new or a returning student, we could not be happier to be on this journey with you.

This has been a challenging period for everyone. The COVID-19 pandemic has prompted a huge change in society as well as how we deliver our programmes at Swansea University and the way in which you study, research, learn and collaborate. We have been working hard to make sure you will have or continue to having an excellent experience with us.

We have further developed some exciting new approaches that I know you will enjoy, both on campus and online, and we cannot wait to share these with you.

At Swansea University and in the Faculty of Science & Engineering, we believe in working in partnership with students. We work hard to break down barriers and value the contribution of everyone. Our goal is an inclusive community where everyone is respected, and everyone's contributions are valued. Always feel free to talk to academic staff, administrators, and your fellow students - I'm sure you will find many friendly helping hands ready to assist you.

We all know this period of change will continue and we will need to adapt and innovate to continue to be supportive and successful. At Swansea we are committed to making sure our students are fully involved in and informed about our response to challenges.

In the meantime, learn, create, collaborate, and most of all – enjoy yourself!

Professor Johann (Hans) Sienz
Interim Pro-Vice Chancellor/Interim Executive Dean
Faculty of Science and Engineering

Faculty of Science and Engineering	
Interim Pro-Vice Chancellor/Interim Executive Dean	Professor Johann Sienz
Head of Operations	Mrs Ruth Bunting
Associate Dean – Student Learning and Experience (SLE)	Professor Paul Holland
School of Mathematics and Computer Science **Head of School: Professor Elaine Crooks**	
School Education Lead	Dr Neal Harman
Head of Computer Science	Professor Xianghua Xie
Computer Science Programme Director	Undergraduate: Dr Liam O'Reilly MSc: Dr Bertie Müller
Year Coordinators	MSc Cyber Security Course Coordinator: Dr Pardeep Kumar MSc Computer Science Course Coordinator: Dr Oliver Kullmann MSc Advanced Software Technology Course Coordinator: Dr Anton Setzer MSc Advanced Computer Science Course Coordinator: Dr Anton Setzer MSc Data Science Course Coordinator: Dr Alma Rahat

DISCLAIMER

The Faculty of Science and Engineering has made all reasonable efforts to ensure that the information contained within this publication is accurate and up-to-date when published but can accept no responsibility for any errors or omissions.

The Faculty of Science and Engineering reserves the right to revise, alter or discontinue degree programmes or modules and to amend regulations and procedures at any time, but every effort will be made to notify interested parties.

It should be noted that not every module listed in this handbook may be available every year, and changes may be made to the details of the modules. You are advised to contact the Faculty of Science and Engineering directly if you require further information.

The 22-23 academic year begins on 26 September 2022

Full term dates can be found here

DATES OF 22-23 TERMS

26 September 2022 – 16 December 2022

9 January 2023 – 31 March 2023

24 April 2023 – 09 June 2023

SEMESTER 1

26 September 2022 – 27 January 2023

SEMESTER 2

30 January 2023 – 09 June 2023

SUMMER

12 June 2023 – 22 September 2023

IMPORTANT

Swansea University and the Faculty of Science of Engineering takes any form of **academic misconduct** very seriously. In order to maintain academic integrity and ensure that the quality of an Award from Swansea University is not diminished, it is important to ensure that all students are judged on their ability. No student should have an unfair advantage over another as a result of academic misconduct - whether this is in the form of **Plagiarism**, **Collusion** or **Commissioning**.

It is important that you are aware of the **guidelines** governing Academic Misconduct within the University/Faculty of Science and Engineering and the possible implications. The Faculty of Science and Engineering will not take intent into consideration and in relation to an allegation of academic misconduct - there can be no defence that the offence was committed unintentionally or accidentally.

Please ensure that you read the University webpages covering the topic – procedural guidance here and further information here. You should also read the Faculty Part One handbook fully, in particular the pages that concern Academic Misconduct/Academic Integrity. You should also refer to the Faculty of Science and Engineering proof-reading policy and this can be found on the Community HUB on Canvas, under Course Documents.

STUDENT SUPPORT

The Faculty of Science and Engineering has two **Reception** areas - Engineering Central (Bay Campus) and Wallace 223c (Singleton Park Campus).

Standard Reception opening hours are Monday-Friday 9am-5pm.

The **Student Support Team** provides dedicated and professional support to all studentsin the Faculty of Science and Engineering. Should you require assistance, have any questions, be unsure what to do or are experiencing difficulties with your studies or in your personal life, our team can offer direct help and advice, plus signpost you to further sources of support within the University. There are lots of ways to get information and contact the team:

Email: studentsupport-scienceengineering@swansea.ac.uk (Monday–Friday, 9am–5pm)

Call: +44 (0) 1792 295514 and 01792 6062522 (Monday-Friday, 10am–12pm, 2–4pm).

Zoom: By appointment. Students can email, and if appropriate we will share a link to our Zoom calendar for students to select a date/time to meet.

The current student **webpages** also contain useful information and links to otherresources:

https://myuni.swansea.ac.uk/fse/coe-student-info/

READING LISTS

Reading lists for each module are available on the course Canvas page and are also accessible via http://ifindreading.swan.ac.uk/. We've removed reading lists from the 22-23 handbooks to ensure that you have access to the most up-to-date versions. Access to print material in the library may be limited due to CV-19; your reading lists will link to on-line material whenever possible. We do not expect you to purchase textbooks, unless it is a specified key text for the course.

THE DIFFERENCE BETWEEN COMPULSORY AND CORE MODULES

Compulsory modules must be **pursued** by a student.

Core modules must not only be **pursued**, but also **passed** before a student can proceed to the next level of study or qualify for an award. Failures in core modules must be redeemed.

Further information can be found under "Modular Terminology" on the following link - https://myuni.swansea.ac.uk/academic-life/academic-regulations/taught-guidance/essential-info-taught-students/your-programme-explained/

MSc (FHEQ Level 7) 2022/23
Advanced Software Technology
MSc Advanced Software Technology

Coordinator: Dr AG Setzer

Compulsory Modules

Semester 1 Modules	Semester 2 Modules
CSCM01 **Software Engineering Project Planning and Management** **15 Credits** **Ms L Powell**	**CSCM10** **Computer Science Project Research Methods** **15 Credits** **Dr MJ Roach**
	CSCM64 **Software Testing** **15 Credits** **Dr E Neumann**
CSCM04 **Software Team Project** **15 Credits** **Dr T Owen**	
Dissertation	
CS-M20 **MSc Project** **60 Credits** **Dr U Berger**	
Total 180 Credits	

Optional Modules

Choose a maximum of 30 credits

Graduates from our BSc programmes in Computer Science at Swansea University are usually not allowed to take modules of which they have already taken the level 3 version. The department aims to offer sufficient modules to allow a balanced choice of optional modules. In case of queries regarding the required modules for your scheme, please contact the course coordinator for the respective scheme.

Select two modules.

CSCM08	Information Security Management	Dr B Muller	TB1	15
CSCM13	Critical Systems	Dr AG Setzer	TB1	15
CSCM27	Human-Centred Visual Analytics	Dr DW Archambault	TB1	15
CSCM68	Embedded System Design	Dr H Nguyen/Prof SA Shaikh	TB1	15
CSCM72	Optimisation	Dr AAM Rahat	TB1	15
CSCM85	Modelling and Verification Techniques	Dr U Berger	TB1	15
CSCM98	Operating Systems and Architectures	Dr B Mora	TB1	15

And

Choose a maximum of 15 credits

You may choose to make no selection in this section.

CSCM18	IT-Security: Cryptography and Network Security	Dr P Kumar/Dr P James	TB1	15
CSCM45	Big Data and Machine Learning	Dr Z Li	TB1	15
CSCM48	Web Application Development	Dr SP Walton	TB1	15

And

Choose a maximum of 30 credits

Select two modules.

CSCM29	Blockchain, Cryptocurrencies and Smart Contracts	Dr AG Setzer	TB2	15
CSCM35	Big Data and Data Mining	Dr S Yang	TB2	15
CSCM37	Data Visualisation	Dr C Wacharamanotham/Dr B Mora	TB2	15
CSCM38	Advanced Topics: Artificial Intelligence and Cyber Security	Prof SA Shaikh	TB2	15
CSCM39	Human Computer Interaction	Dr MI Ahmad	TB2	15
CSCM75	Logic in Computer Science	Dr U Berger	TB2	15
CSCM77	Computer Vision and Deep Learning	Prof X Xie	TB2	15
CSCM79	Hardware and Devices	Dr DR Sahoo	TB2	15

And

Choose a maximum of 15 credits
You may choose to make no selection in this section.

CSCM45J	Big Data and Machine Learning	Dr S Sharifzadeh	TB2	15
CSCM48B	Web Application Development	Dr A Buzdalova	TB2	15

CS-M20 MSc Project

Credits: 60 Session: 2022/23 Academic Year

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr U Berger

Format: Individual project supervision

Delivery Method: Individual project supervision

Module Aims: This module will provide students with the opportunity of exploring a particular topic in computer science in some considerable depth. It is only open to students studying MSc Computer Science, MSc Advanced Computer Science, MSc Advanced Software Technology, MSc in High Performance and Scientific Computing, and MSc Data Science.

Module Content: The student will carry out independent project under the guidance of their supervisor.
The dissertation may include the
following topics:

- Discussion of the subject area and its history;
- A literature survey;
- Formulation of scientific questions and the answers to them;
- Theoretical background;
- Description of the approach taken;
- Discussion of issues arising in the undertaking of the project;
- Evaluation of results;
- Progress and achievements of the project;
- Suggestions for further work.

Intended Learning Outcomes: Students will be able
to undertake independent research into appropriate areas of Computer Science;
plan and undertake a significant independent piece of project work;
critically evaluate their work in the context of current work in related areas.

Assessment: Project (100%)

Assessment Description: Project dissertation. The maximum word count for a Swansea University MSc is defined in the online Academic Guide:
http://www.swan.ac.uk/registry/academicguide/

Moderation approach to main assessment: Universal double-blind marking

Assessment Feedback: Students will receive guidance from their academic supervisor during individual supervision meetings. The minimum frequency of these is defined in University regulations; though it is expected that in practice they will be more frequent. Formal notification of the result of the MSc dissertation will be sent to the student via usual University processes. The student will receive individual feedback on their dissertation from their supervisor.

Failure Redemption: Resubmit dissertation in accordance with University regulations.

Additional Notes: Only available to students pursuing an MSc degree in Computer Science.

CSCM01 Software Engineering Project Planning and Management

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Ms L Powell

Format: Lectures and Lab Sessions

Delivery Method: Lectures, Lab Sessions, directed reading, guided project development planning.

Module Aims: Software projects have long had a reputation for cost and time overruns - but they need not, and there are well-established, and emerging, techniques and processes to manage them well and effectively: for example, agile methodologies like SCRUM which are becoming a de-facto standard in the industry. Also, many projects have significant legal, social, ethical and professional consequences that a practitioner needs to be aware of and sensitive to. This module is about the process of successfully building complex software systems, and the implications, including on wider society, of doing so. It will also prepare students for their dissertation project by equipping them with the skills to successfully plan it, and to commence that planning process.

Module Content: Project planning and management principles:
- Timescales and dependencies
- Scoping and resources
- Risks: identifying, quantifying, managing, monitoring, and mitigating
- Team management

Requirements and Specification
- Heavyweight and lightweight models

Methodologies for developing software:
- Traditional - waterfall, prototyping, spiral
- Rapid Application Development (RAD)
- Iterative and incremental
- Agile development
- Hybrid models - Scrum (agile/incremental)

Legal, Social, Ethical, and Professional Issues

Intended Learning Outcomes: On completion of this module, students will:
*Understand and be able to participate in the legal, social, ethical and professional framework, particularly with reference to software engineering.
*Have an in-depth knowledge of a range of software development models, including agile as well as traditional, and their specific properties, advantages, and disadvantages.
*Choose between a range of software development models, with reference to the particular nature, constraints and priorities of a particular project.
*Understand the issues of project planning and management in general, including risk analysis and controls, time scale and resource planning, exception monitoring, and progress monitoring and control.
*Understand the issues of team work in software projects.
*Have experience of project planning, and choosing a methodology, for a specific project that they will undertake in the future.
*Have experience of considering the legal, social, ethical and professional implications of a specific project they will undertake in the future.

Assessment:	Report (20%)
	Group Work - Coursework (50%)
	Coursework 1 (20%)
	Laboratory work (10%)
Resit Assessment:	Coursework reassessment instrument (100%)

Assessment Description: •Individual Coursework – Requirements, Dependencies and CPA
•Individual Report – Project Proposal – to include:
- description of the project/problem to be solved (requirements);
- outline of proposed solution (specification);
- identification of LSEPI issues;
- identification of pre-project risks (not those arising from choice of tools/methodology etc.).

•Group Work – Project Plan – to include:
- Refinements/changes to proposed solution (since Project Proposal Document);
- Methodology choice;
- Technology choices;
- Project plan and dependencies (if appropriate - dependent on methodology choice);
- Basic risk analysis and management (including risk arising out of methodology and technology choices);
- Testing plan

•Laboratory Work – Weekly 1-hour lab session with tasks to be signed off in lab sessions

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Individual categorised feedback (structured by assessment criteria) on each coursework component.
Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.

Failure Redemption: Resubmission of failed component(s)

Additional Notes:

Only available to MSc Advanced Software Technology students. Updated September 2016.

CSCM04 Software Team Project

Credits: 15 Session: 2022/23 September-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr T Owen

Format: 20 hours project supervision,
10 hours lecture.

Delivery Method: Primarily on campus

Module Aims: The aim of this module is to provide students with the opportunity to apply their specialised knowledge to a realistic problem, and gain practical experience of the processes involved in the team-based production of software.

Module Content: Project planning, tools and techniques for planning.
Agile and traditional software development methodologies, including SCRUM.
Project conduct, time management, risk analysis management, and team working.
Application of legal, ethical, social and professional issues applicable to the computer industry, in the specific context of software development projects.

Students will specify, develop, test and document a substantial software system under the supervision of an academic staff member.

Significant emphasis will be placed on delivery - that is, meeting stated project goals
Somewhat less emphasis will be placed on ambitious technological solutions.

Intended Learning Outcomes: Students will gain experience of: extended team working; project planning; resource estimation; risk analysis; software development; document writing; and system and document versioning.

Students will have demonstrated their ability to apply their specialised knowledge to a realistic problem, and have gained practical experience of the processes involved in the team-based production of software.

Students will know what is required for a team project to work effectively and successfully.

1. Hold regular group meetings and maintain a diary of their progress.
2. Define and allocate tasks to individual members according to their skills.
3. Use available project management tools to organise their activities.
4. Produce a fully tested working implementation of their software.
5. Deliver appropriate documentation of a professional standard.
6. Give a presentation in which they described the way the project was managed, explained their design and demonstrated their final product.

Each individual student will be expected to have contributed fully to his or her team's activities, and will be able to:

1. Describe the processes involved in the team-based production of software.
2. Explain and justify the design of their team's finished product.
3. Give an account of their individual contribution to the team's effort.

Assessment: Coursework 1 (30%)
Coursework 2 (10%)
Coursework 3 (60%)

Resit Assessment: Coursework reassessment instrument (100%)

Assessment Description: Milestone1. Typically this contains the following deliverables:

- Team Structure, Methodology and Risks Document describing the roles of the team members, the methodology they will use and a risk analysis, including steps taken to 'design out' risks with a relatively high likelihood and serious consequences. This document should contain an explicit analysis of the legal, social, professional and ethical issues related to the project, how they relate to the deliverables and development of the project, and how, where appropriate, they appear as project risks. (approx 10 pages).
- Requirements Document describing the requirements of the system in a structured form (typically 5-10 pages but depends on project).
- Specification Document describing the specification of the behaviour (and if relevant other aspects of the project - e.g. performance) in a precise and structured form (typically 5-10 pages but depends on project).

Note that, depending on the methodology chosen, it may not be appropriate to explicitly describe the full requirements/specification of the final product in a traditional way. For example, in the case of a SCRUM-based project, it may be more appropriate to describe typical/indicative projected/forecast sprints instead.

Milestone 2. Deliverables:

- Interim Report summarising in a narrative form the progress made on the project at this point (typically 10-20 pages but depends on the project).

Milestone 3. Deliverables:

- Poster session describing the project and its status, and a demonstration of the software; User Manual (length variable).
- Design Document describing the design of the system, preferably using a (semi-)formal notation such as UML, and including a rationale for design decisions made (length variable;).
- Testing Document describing in a structured way the tests applied and their outcome (length variable).
- Narrative and Reflective Account describing the team's experiences, problems encountered and solved (or not) and the team's reflections on the successful and less successful aspects of the project, including how well the team was able to work together (typically 10-15 pages).

Moderation approach to main assessment: Universal double-blind marking

Assessment Feedback: Structured written feedback using an assessment form and grades.

Failure Redemption: Resubmit documents as appropriate.

Additional Notes:

Created March 2015. This module is only available to Level M (final year) MEng Computing students and MSc Advanced Software Technology students.

This is a group coursework module, and students will be assessed collectively. It is normally expected that all students participating will receive the same mark for each component. However, if the students or co-ordinator feel this will be unfair, a system based on each students' ranking of their own and the other team members' work will be used. It is generally expected that students will work in teams of three or four.of three or four.

Updated November 2016

CSCM08 Information Security Management

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr B Muller

Format: 30 hours lectures and seminars

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: This module will address the theory and practice of information security. In particular, it will consider where data comes from, who collects it and what they can do with it. It will further look into theories of monitoring and surveillance, digital identity, legal and regulatory frameworks, data protection, cybercrime, business resilience, disaster recovery, and security audits.

Module Content: The theory and practice of information security: where does data come from, who collects it and what can they do with it? Data as a management tool, commodity, private asset, public good and public service.

Theories of monitoring and surveillance.

Theories of digital identity with applications to trust, anonymization and privacy. Technologies: biometrics, authentication, access control.

Legal and regulatory frameworks. Information Commissioners Office. Development of data protection. General Data Protection Regulation 2018. Company security policies and practices on digital media: use of email, the web and databases whilst at work, travelling and at home. Failures of information security: internal versus external. Case studies of data breaches.

The global landscape of cybercrime. Classification of cybercrime. Hackers and mules -- social engineering, leakage, penetration, betrayal, etc. Case studies of cybercrime, especially fraud. Convergence of real and virtual crimes.

Business resilience, continuity and disaster recovery. Risk analysis. Security audits. Role of chief information security officers.

Intended Learning Outcomes: Students will be able to
- critically evaluate the personal, organisational, and legal/regulatory context in which information systems could be used, the risks of such use and the constraints that may affect how cyber security is implemented and managed,
- explain security requirements, and specify appropriate security measures,
- critically analyse the nature, role and problems of data in all aspects of modern life as well as the scope and limits of technologies and human factors in security,
- carry out risk analysis and evaluate compliance issues for data in an organisation or company,
- undertake security audits of policies, practices and technologies.

Assessment:	Group Work - Presentation (30%)
	Report (70%)
Resit Assessment:	Coursework reassessment instrument (100%)

Assessment Description: Presentation and report on an individual topic.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Formative feedback during term time. Written individual feedback on presentation and report outlining strengths and weaknesses.

Failure Redemption: Use of resit instrument as appropriate.
Additional Notes: Available to visiting and exchange students.

CSCM10 Computer Science Project Research Methods

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr MJ Roach

Format: 12 lectures and seminars plus 6 one-to-one project supervision meetings; 3 hours presentations

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: This module will introduce students to some fundamental research methodologies and good practice in research. They will undertake background research including a literature review and specify the aims of their MSc project.

Module Content: Seminars about selected scientific texts and research projects
Lectures on
• fundamental research methodologies
• good practice in research
• formulation of research questions and hypotheses
• logical reasoning
• literature research
• proper acknowledgement of sources
• principles of carrying out experimental research including ethical issues
• presentation of results
Individual guidance from project supervisors on
• identifying a research topic
• finding and reading related work
• report writing, citations and references
• using (digital) library services and search tools

Intended Learning Outcomes: Students will:-
- have gained a thorough understanding of fundamental research methodologies and good practice in research including:
• the formulation of research questions and hypotheses;
• techniques of valid and convincing argumentation;
• literature research methods;
• the proper acknowledgement of sources;
• the extraction of information from literature;
• project planning.

- be conversant with the principles of carrying out experimental research.
- have an understanding of how scientific research is conducted, reported, reasoned about and evaluated.
- be able to show their understanding of the requirements of a masters level project by writing a formal project proposal and specification which contains an outline solution to the problem and which clearly defines the scope of the MSc project, its goals, the methodology to be undertaken and the criteria of its evaluation
- have gained an in-depth knowledge in specific areas related to their project, and have critically assessed different methods to be used in their project and will have developed a detailed plan for carrying out their project.
have an understanding and appreciation of the importance of relevant legal, social, ethical and professional issues as they relate to their project.

Assessment:	Presentation (30%)
	Report (60%)
	Class Test 1 - Coursework (10%)
Resit Assessment:	Coursework reassessment instrument (100%)

Assessment Description: Report on background research in project area 15% (end February, Semester 2)
Careers Development exercise 10% (February/March, Semester 2)
Oral presentation on project aim/background research 25% (end March/April , Semester 2)
Detailed specification of proposed summer project 50% (May, Semester 2)

Moderation approach to main assessment: Universal double-blind marking

Assessment Feedback: Individual feedback will be given by markers (CSCM10 lecturers, supervisors and second markers) using marking pro-forma. The comments and marks of the detailed specification document will be discussed by the project supervisor at individual meetings.

Failure Redemption: Failure to be redeemed by submitting document addressing unsatisfactory aspects of initial submission(s).
Additional Notes: Only available to students on MSc Computer Science, MSc Advanced Computer Science, MSc Advanced Software Technology.

CSCM13 Critical Systems

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr AG Setzer

Format: 20 hours lectures, 10 hours lab.

Delivery Method: On campus lectures.

Module Aims: This module introduces techniques for developing critical systems, especially safety critical systems. Students will gain experience in applying modern tools in the development of critical software.

Module Content: Introduction and Motivation:
What are high integrity and critical systems? Legal and ethical issues. Examples of major failures of high integrity systems. Successes and how/why they worked. Standards for safety-critical software and their shortcomings.

Analysis:
The hazard analysis process. Safety analysis and the safety case. Safety issues related to, but outside software. Human factors - the role of poor interfaces in software failures.

Specification and Verification:
Languages and tools for formal specification and verification of software. Detailed demonstration of one tool and its underlying theory.

Software Production:
Issues in program language selection to minimise failure. The software engineering process in the production of high-integrity software;

Correctness:
Validation and verification - the advantages and disadvantages of testing and formal verification.

Intended Learning Outcomes: Students will be thoroughly familiar with issues surrounding safety-critical systems, including legal and ethical issues and hazard analysis. They will understand techniques for specifying and verifying high-integrity software. They will have experience in applying formal specification techniques to critical systems. They will be familiar with and have had experience in applying programming languages suitable for developing high-integrity software for critical systems.

Assessment:	Examination 1 (60%)
	Coursework 2 (20%)
	Coursework 1 (20%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
The coursework consists of
Assignment 1 - Programming tasks
Assignment 2 - Case study

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Coursework: small report on strength and weaknesses of each solution.
Examination feedback summarising strengths and weaknesses of the class.

Failure Redemption: Resit exam and/or resubmit assignments as appropriate.

Additional Notes:

Created March 2015. Available to visiting and exchange students.
No changes July 2015

CSCM18 IT-Security: Cryptography and Network Security

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr P Kumar, Dr P James

Format: 30 hours lectures and labs

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: The aim of this course is to examine theoretical and practical aspects of computer and network security.

Module Content: Security threats and their causes.
Security criteria and models.
Cryptography: including basic encryption, DES, AES, hash functions.
Access Control.
Security tools and frameworks: including IPSec, TLS, SSL, SSH and related tools.
Vulnerabilities and attacks: including port scanning, packet sniffing, SQL injection.
Security issues in wireless networks.
Security on the cloud.

Intended Learning Outcomes: Students will have the ability to identify security threats and their causes in today's computing infrastructures.
Students will be able to explain in detail and apply techniques from Crytography and Cryptanalysis.
Students will synthesize the concepts of design, defensive programming, as well as their application to
to build robust and resilient systems.
Students will be able to apply techniques to enhance the security of existing systems, and gain a critical awareness of the limits of these techniques.
Students will be able to reflect and critique on cryptographic techniques and secure systems design.

Assessment:	Examination 1 (70%)
	Coursework 1 (10%)
	Coursework 2 (10%)
	Laboratory work (10%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination (70%).
2 Courseworks and work done in a lab.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Examination feedback summarising strengths and weaknesses of the class.

Failure Redemption: Resit exam.

Additional Notes:

Available to visiting and exchange students.

CSCM27 Human-Centred Visual Analytics

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr DW Archambault

Format: 20 hours lectures,
10 hours practicals

Delivery Method: On campus lectures.

Module Aims: Visual analytics is a human-centred area of computing that studies systems to support exploration and explanation of features in data sets. The course will provide an introduction to visual analytics and supporting necessary concepts in information visualisation. Visual analytics is the combination of visualisation and data analytics, which includes machine learning, to create effective data analysis systems.

Module Content: History and goals of visual analytics.

Types of data and encodings.

Data processing and clustering.

Information visualisation techniques.

The analytics process and pipeline.

Intended Learning Outcomes: Students will be able to apply the principles of human perception to construct and evaluate visual interfaces of data;

Students will be able to apply human-centred techniques as related to visualisation (e.g. paper prototyping and human-centred design);

Students will be able to analyse and apply data processing for visual analytics purposes;

Students will be able to apply necessary concepts in information visualisation that support the analytics process;

Students will have experience with techniques in visual analytics and information visualisation;

Students will have experience in applying the right visual analytics technique to the right problem.

Assessment:	Coursework 1 (25%)
	Presentation (15%)
	Report (40%)
	Report 2 (20%)
Resit Assessment:	Coursework reassessment instrument (100%)

Assessment Description: Coursework 1: Visual Analytics Individual Assignment

Students select a data set and design a small visual analytic system around that data set. The system will have an analysis and visualisation component that will work together.

Presentation and report (December):
Students will select a visual analytics problem and study it through the term. They will present their solution and implementation in the form of a report. This assessment will be evaluated in the form of a report and implementation of the desired technique. The project will be conducted in small groups.

Report 2: Paper Summary (January):
Students will summarise a paper from the state of the art and demonstrate that they can understand research contributions in the wider field. In particular, they will state the objective of the work, how it fits into the related work, and summarise the technical content. This understanding is commensurate with their experience with the material.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Individual feedback for presentation and report for each pair.

Failure Redemption: Resubmit coursework(s) as appropriate.

Additional Notes:

Generated December 2016. Available to visiting and exchange students.

CSCM29 Blockchain, Cryptocurrencies and Smart Contracts

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr AG Setzer

Format: 30 hours including presentation and consultation hours

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: This is a module on modern blockchain technology and its major applications. It will give an overview on the technological setup of major cryptocurrencies, and introduce the blockchain as a concept for determining the order of events in a distributed database. In addition, it will discuss the implementation of smart contracts and summarise the current state of the art of security issues in cryptocurrencies, blockchain technology, and smart contracts.

Module Content: Introduction to cryptocurrencies, blockchain technology and smart contracts
History of cryptocurrencies.
From the model of a bank to the Bitcoin model.
The Bitcoin client.
Transactions, keys, addresses, wallets.
The Bitcoin network.
Mining and consensus.
An overview over other cryptocurrencies.
History and Philosophy of Ethereum.
Smart contracts.
Smart contract development using Solidity.
Security of Smart Contracts.

Intended Learning Outcomes: Students will be able to

- explain blockchain technology and critically evaluate its current and future applicability,
-- explain the theoretical concepts behind cryptocurrencies and blockchain technology, and be able to critically reflect on issues surrounding their application, -
- explain the concepts behind smart contracts, be able to apply them in a lab environment, and critically evaluate their applicability as a technology,
-- undertake a research project on a topic in the areas of cryptocurrencies, blockchain technology, and smart contracts or develop and document software related to these areas,
-- apply blockchain technology to scenarios in a lab and critically evaluate its usability potential in the real world.

Assessment:	Examination 1 (60%)
	Coursework 1 (20%)
	Coursework 2 (20%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
Coursework 1: Java-based exercise (Java programming skills required)
Coursework 2: Solidity coursework

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Examination feedback summarising strengths and weaknesses of the class.

Failure Redemption: Resit exam and/or resubmit assignments as appropriate.

Additional Notes:

Created January 2018. Available to visiting and exchange students.

CSCM35 Big Data and Data Mining

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr S Yang

Format: 20 hours lectures, 10 hours lab

Delivery Method: Primarily on campus

Module Aims: This module introduces students to the fundamental topics of data mining, including data pre-processing techniques, applied probability and statistics, data mining algorithms (incl. associate rule, classification, clustering, outlier detection and probabilistic graphical model), and big data frameworks.

Module Content: Basic knowledge in machine learning and mathematics are required, where we students are strongly encouraged to take CSCM45 and CSCM70 in the first semester. For those who have not taken those two modules, we will revisit some key concepts in the first three weeks, i.e. probability and statistics. Practical work will be done through the medium level of Python programming. You will also get to discover key scientific libraries of Python, i.e. NumPy, SciPy, Scikit-Learn, Matplotlib, Tensorflow, Hadoop, Spark. Classes in the first 3 weeks will include support time for Python programming and the use of these libraries.

Course Overview and Python Programming for Data Science
Mathematics Background: Numerical Analysis, Applied Probability and Statistics
Frequent Pattern, Association, Correlations
Naïve Bayes Classifier, Quantitative Evaluation
Decision Tree, Random Forest, AdaBoost
Maximum Likelihood Estimation, Expectation Maximisation
Clustering, DBSCAN, High Dimensional Data Analysis
Text Data Analysis, Word2Vec, Skip-Through, CBOW
Time-Series Data Analysis, Regression, Hidden Markov Model
Scalability and Efficiency Big Data Analysis

Intended Learning Outcomes: After completing this module students will:
- be able to manipulate complex, large, heterogeneous datasets, from storage to processing
- be able to extract information from complex, large, heterogeneous datasets
- be able to critically evaluate and select data mining algorithms and techniques, and be able to apply them in real world applications.

Assessment: Laboratory 1 (15%)
Laboratory 2 (15%)
Examination (70%)
Resit Assessment: Examination (Resit instrument) (100%)

Assessment Description: Lab 1 15% (February): Examination of Python programming in Data Science
Lab 2 15% (March): Examination of Data Mining and Machine Learning algorithm
Exam 70% (May/June): Standard Format Computer Science Exam

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Grades, individual and collective written feedback for coursework.

Failure Redemption: Redemption of failure via resit instrument.

Additional Notes: Available to visiting and exchange students. Basic knowledge in machine learning and mathematics are required and Computer Science students are strongly encouraged to take CSCM45 and CSCM70 in the first semester. For those who have not taken those two modules, some key concepts will be revisited in the first three weeks, i.e. probability and statistics. NOTE: students on the MSc Applied Data Science will have covered the relevant material in MA-M06 and MA-M16

CSCM37 Data Visualisation

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr C Wacharamanotham, Dr B Mora

Format: 20 hours lectures,
10 hours practicals

Delivery Method: On-campus and lab sessions.

Module Aims: Data Visualisation is concerned with the automatic or semi-automatic generation of digital images that depict data in a meaningful way(s). It is a relatively new field of computer science that is rapidly evolving and expanding. It is also very application oriented, i.e., real tools are built in order to help scientists from other disciplines.

Module Content: Introductory topics include: purposes and goals of visualisation, applications, challenges, the visualisation pipeline, sources of data: data dimensionality, data types, and grid types.

Information visualisation topics include: abstract data, hierarchical data, tree maps, cone trees, focus and context techniques, hyperbolic trees graphs and graph layouts, multi-dimensional data, scatter plots, scatter plot matrices, icons, parallel coordinates, interaction techniques, linking and brushing.

Volume visualisation topics include: slicing, surface vs. volume rendering, transfer functions, interpolation schemes, direct volume visualisation, ray casting, shear-warp factorisation, image order vs. object order algorithms, gradients, filtering, interpolation, and isosurfacing.

Flow visualisation topics include: simulation, measured, and analytical data, steady and time-dependent (unsteady) flow, direct and indirect flow visualisation, applications, hedgehog plots, vector glyphs, numerical integration schemes, streamlines, streamline placement, geometric flow visualisation techniques, line integral convolution (LIC), texture-based techniques, feature-based flow visualisation.

Intended Learning Outcomes: Students will be able to:
- identify problems that can be addressed with visualisation.
- comprehensively explain data visualisation techniques and be able to critically appraise their suitability to particular situations.
- choose, evaluate and apply visualisation techniques to effectively reveal insights into complex and potentially-incomplete data.

Assessment: Examination 1 (60%)
Coursework 1 (20%)
Coursework 2 (20%)
Resit Assessment: Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
Two practical courseworks.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Examination feedback summarising strengths and weaknesses of the class.

Failure Redemption: Resit examination and/or resubmit coursework(s) as appropriate.

Additional Notes: Available to visiting and exchange students.

CSCM38 Advanced Topics: Artificial Intelligence and Cyber Security

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Prof SA Shaikh

Format: 20 lectures, 5 x 1hr seminars, 3 x 2hr labs, 2 drop-in hours.

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: This module introduces students to the state-of-the-art methods and research topics of artificial intelligence,

cyber security, including quantum computing, data science, deep learning and reinforcement learning. The

inspiration behind these approaches will be discussed, along with their relative merits for application in cyber

security.

Module Content: 1. Advanced AI Topics: Deep Supervised Learning: Advanced Topics on CNNs, RNNs and GNNs.

2. Advanced AI Topics: Deep Unsupervised Learning: Advanced Topics on AE, Autoregression Models, Flow, GANs.

3. Data Security: Federated Learning, Differential Privacy in Learning.

4. Model Security: Adversarial Attack and Defense, Robustness Testing.

5. AI for Cybersecurity: Malicious Intrusion Detection etc

Intended Learning Outcomes: Student will be able
- to demonstrate a broad knowledge of the state-of-the-art concepts and techniques of artificial intelligence (AI) for cyber security,
- to appraise vulnerabilities and risks introduced by AI,
- to compare and contrast different analysis methods for cyber security problems,
- to carry out independent research on AI and data science topics related to cyber security.

They should further be able
to transfer the knowledge to solve cyber security problems, from a computation, artificial intelligence and data science perspective.

Assessment:	Coursework 1 (50%)
	Coursework 2 (50%)
Resit Assessment:	Coursework reassessment instrument (100%)

Assessment Description: Two coursework assignments (50% each)

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Individual written feedback and cohort feedback

Failure Redemption: Resubmit coursework as appropriate.

Additional Notes:

Updated July 2019. Available to visiting and exchange students

CSCM39 Human Computer Interaction

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr MI Ahmad

Format: 30 Lectures including presentations and consultation hours.

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: This module gives an overview on the main topics in Human Computer Interaction and helps students understand research and research processes in Human Computer Interaction. Students explore the advanced literature and research results underpinning the field of HCI. Classic papers and controversies are covered, as well as recent work from the leading figures. Students achieve a clear view of the 'cutting edge' and issues in the field and where things are happening. The module is very interactive, and students will be expected to give presentations.

Module Content: Advanced topics in Human Computer Interaction including:
- Interface design.
- Usability.
- Evaluation.
- Human factors.
- Human error.
- Cognitive science, and their role in the field.
- User models.
- User experience.
- Larger systems, ethics, design principles.

Research topics: Classic literature and personalities
Research Methods in HCI.

Intended Learning Outcomes: The ability to demonstrate detailed understanding of advanced topics in HCI including topics such as interface design, human factors, human error, cognitive science, etc, and their role in the field.

The ability to review and critically assess the literature on specific topics at the current limits of theoretical and research understanding.

The ability to analyse and present the results of a literature review both as a scientific report (paper, poster, etc) and as an oral presentation.

Assessment: Coursework 1 (30%)
Coursework 2 (70%)
Resit Assessment: Coursework reassessment instrument (100%)

Assessment Description: 3 courseworks.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Individual written feedback.

Failure Redemption: Resubmit coursework as appropriate.

Additional Notes:

Created April 2015. Available to visiting and exchange students

CSCM45 Big Data and Machine Learning

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr Z Li

Format: 20 hours lectures, 10 hours lab.

Delivery Method: On campus lectures.

Module Aims: This module will discuss in-depth some of the most widely used and state-of-the-art artificical intelligence and machine learning techniques and their applications to big data problems. The students will gain both theoretical understanding of learning and practical know-how in applying those theories to real world problems. Topics include big data concept, data mining, learning theories, supervised and unsupervised learning, and reinforcement learning.

Module Content: This module covers three parts: introduction to big data and learning, data analysis techniques, and learning concepts and methods.

Introduction to big data and data mining;
Data clustering;
Dimensionality reduction: linear techniques;
Dimensionality reduction: nonlinear techniques;
Discriminative analysis;
Learning theory, including bias and variance theory, innovation process in machine learning;
Expert systems;
Unsupervised learning;
Supervised learning, including parametric and nonparametric methods, neural network, kernels, support vector machine, randomised decision trees;
Reinforcement and adaptive control;
Example applications to bioinformatics, health informatics, and web data processing.

Intended Learning Outcomes: Upon completion of this module students will be able to:
- Describe, explain, and critique the fundamental techniques of analysing complex and heterogeneous data.
- Describe and explain machine learning techniques and their applications to big data problems.
- Discuss and contrast both conventional and state-of-the-art machine learning techniques.
- Implement and apply machine learning techniques to synthesise solutions.
- Analyse big data problems and evaluate and devise potential solutions.

Assessment: Examination 1 (60%)
Coursework 1 (20%)
Coursework 2 (20%)
Resit Assessment: Examination (Resit instrument) (100%)

Assessment Description: Standard format Computer Science exam.
Essay based practical programming assignment.
Lab work with smaller tasks to complete.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with analytical individual feedback for assignment.
Examination feedback summarising strengths and weaknesses of the class.

Failure Redemption: Redemption of failure via resit instrument.

Additional Notes: Created March 2015; updated July 2016. Available to visiting and exchange students.

CSCM45J Big Data and Machine Learning

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr S Sharifzadeh

Format: 20 hours lectures, 10 hours lab.

Delivery Method: On campus lectures.

Module Aims: This module will discuss in-depth some of the most widely used and state-of-the-art artificical intelligence and machine learning techniques and their applications to big data problems. The students will gain both theoretical understanding of learning and practical know-how in applying those theories to real world problems. Topics include big data concept, data mining, learning theories, supervised and unsupervised learning, and reinforcement learning.

Module Content: This module covers three parts: introduction to big data and learning, data analysis techniques, and learning concepts and methods.

Introduction to big data and data mining;
Data clustering;
Dimensionality reduction: linear techniques;
Dimensionality reduction: nonlinear techniques;
Discriminative analysis;
Learning theory, including bias and variance theory, innovation process in machine learning;
Expert systems;
Unsupervised learning;
Supervised learning, including parametric and nonparametric methods, neural network, kernels, support vector machine, randomised decision trees;
Reinforcement and adaptive control;
Example applications to bioinformatics, health informatics, and web data processing.

Intended Learning Outcomes: Upon completion of this module students will be able to:

- Describe, explain, and critique the fundamental techniques of analysing complex and heterogeneous data.
- Describe and explain machine learning techniques and their applications to big data problems.
- Discuss and contrast both conventional and state-of-the-art machine learning techniques.
- Implement and apply machine learning techniques to synthesise solutions.
- Analyse big data problems and evaluate and devise potential solutions.

Assessment: Examination 1 (60%)
Coursework 1 (20%)
Coursework 2 (20%)

Resit Assessment: Examination (Resit instrument) (100%)

Assessment Description: Standard format Computer Science exam.
Essay based practical programming assignment.
Lab work with smaller tasks to complete.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with analytical individual feedback for assignment.
Examination feedback summarising strengths and weaknesses of the class.

Failure Redemption: Redemption of failure via resit instrument.

Additional Notes: Available to visiting and exchange students.

CSCM48 Web Application Development

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr SP Walton

Format: 18 hours lectures, 12 hours labs

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: The module will develop the principles and technologies used for building web-based systems. Practical experience of building web systems will be gained via laboratories and coursework. Existing high programming skill and experience is essential for this module.

Module Content: The history of web application development.
HTML and CSS: Introduction and Good Practices.
Web Application Design.
MVC driven web applications
Security and identity in web applications
Web development using Javascript and AJAX

Intended Learning Outcomes: Students will have a systematic understanding of the key aspects of current web programming principles and technologies.
Students will be able to plan and deliver a web application to a deadline.
Students will be able to create web applications following methodological good practice.
Students will be able to design secure web applications and evaluate their effectiveness.
Students will be able to design web applications which provide basic analytics for system administrators.

Assessment: Coursework 1 (10%)
Coursework 2 (20%)
Coursework 3 (70%)
Resit Assessment: Coursework reassessment instrument (100%)

Assessment Description: Coursework 1 and 2 – 30% - Code Review and code submission.

An important part of working in a software engineering organisation is code reviews. In this process engineers look at each other's code to spot bugs and ensure standards are being adhered to. You will submit a source file from your project to be reviews by another student and review another student's source file. You will be assessed both on your adherence to standards with your source code and the quality of your code review.

Coursework 3 – 70% - Implementation.

You will submit the implementation of a small web application. You will be asked to evaluate this by answering a series of questions referencing your implementation. This will assess both your knowledge of the theory and ability to apply that in practice.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Feedback during Presentation of work.
Outline solutions provided along with group and individual analytical feedback for courseworks.
Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.

Failure Redemption: Resit examination and/or resubmit coursework(s) as appropriate

Additional Notes: Students taking this module must have good programming skills (i.e., be a competent programmer in any standard programming language) as this module requires significant programming ability.

Updated August 2019. Available to visiting and exchange students.

CSCM48B Web Application Development

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr A Buzdalova

Format: 18 hours lectures, 12 hours labs

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: The module will develop the principles and technologies used for building web-based systems. Practical experience of building web systems will be gained via laboratories and coursework. Existing high programming skill and experience is essential for this module.

Existing programming experience is essential for this module.

Module Content: The history of web application development.
HTML and CSS: Introduction and Good Practices.
Web Application Design.
MVC driven web applications
Security and identity in web applications
Web development using Javascript and AJAX

Intended Learning Outcomes: Students will have a systematic understanding of the key aspects of current web programming principles and technologies.
Students will be able to plan and deliver a web application to a deadline.
Students will be able to create web applications following methodological good practice.
Students will be able to design secure web applications and evaluate their effectiveness.

Assessment:	Coursework 1 (20%)
	Coursework 2 (10%)
	Coursework 3 (70%)
Resit Assessment:	Coursework reassessment instrument (100%)

Assessment Description: Coursework 1 and 2 – 30% - Code Review and code submission.

An important part of working in a software engineering organisation is code reviews. In this process engineers look at each other’s code to spot bugs and ensure standards are being adhered to. You will submit a source file from your project to be reviews by another student and review another student’s source file. You will be assessed both on your adherence to standards with your source code and the quality of your code review.

Coursework 3 – 70% - Implementation.

You will submit the implementation of a small web application. You will be asked to evaluate this by answering a series of questions referencing your implementation. This will assess both your knowledge of the theory and ability to apply that in practice.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Feedback during presentation of implementation. Individual analytical feedback for courseworks.

Failure Redemption: Resit exam and/or resubmit assignments as appropriate.

Additional Notes: Students taking this module must have good programming skills (i.e., be a competent programmer in any standard programming language) as this module requires significant programming ability.

CSCM64 Software Testing

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr E Neumann

Format: 20 hours lectures;
8 hours practicals/discussion of report, 2 hours consultation.

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: Testing is the process of systematically experimenting with an object (the SUT = System Under Test) in order to establish its quality, where quality means the degree of accordance to the intention or specification. This module will
provide an in-depth introduction to various test scenarios and enable students to gain hands-on experience by means of a number of practical exercises.

Module Content: The module provides a profound overview on industrially relevant methods in software testing and points out current research directions.
• Functional Testing: Boundary Value Testing, Equivalence Class Testing, Decision Table- Based Testing.
• Structural Testing: Path Testing, Data Flow Testing.
• Integration and System Testing: Levels of Testing, Approaches to Integration Testing.
• Object-Oriented Testing: Issues, Class Testing, Object-Oriented Integration Testing.
• Possibly selected Research Topics: e.g. Testing Hybrid Systems.

Intended Learning Outcomes: - Critical understanding of testing as a method to validate software systems;
- The ability to test systems;
- Thorough understanding the levels of testing;
- Ability to critically evaluate and select software test scenarios;
- Ability to perform problem analysis.

Assessment:	Examination (70%)
	Coursework 1 (10%)
	Coursework 2 (10%)
	Laboratory work (10%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs
Coursework 1: Development of Blackbox-Test suites following a number of different approaches
Coursework 2: Development of Whitebox-Test suites following a number of different approaches
Lab work - weekly.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks. Examination feedback summarising strengths and weaknesses of the class.
Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.

Failure Redemption: Resit exam and/or resubmit coursework as appropriate

Additional Notes:

Awareness of propositional and predicate logic will be helpful for this module. Students should have a good understanding of programming and software architecture. Though there will be no programming required in this module, a number of the discussed testing approaches are based on program analysis.

CSCM68 Embedded System Design

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr H Nguyen, Prof SA Shaikh

Format: 12 lectures + 18 lab sessions

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: Embedded systems are information processing systems embedded into enclosing products such as cars, telecommunication or fabrication equipment. They are essential for providing ubiquitous information, one of the key goals of modern information technology.

The aim of this module is to provide an overview of embedded system design, to relate the most important topics in embedded system design to each other, and to obtain an appreciation of the model based approach to embedded systems design.

The lab provides hands-on experience in the design of embedded systems.

Due to the labs' hardware requirements, the number of places available for this module is limited. Places will be allocated during the first week of teaching; the allocation criteria will be announced in the first lecture.

Module Content: The lectures discuss selected techniques in their specialisation to the design of embedded systems such as:

- Common characteristics, Requirements, Specification and Modeling
- Programming-language-level description techniques
- Hardware (Sensors, actuators, processors)
- Operating systems, middleware, scheduling
- Model driven design process
- Hardware/software partitioning and codesign
- Simulation, testing and verification techniques

The labs consist of a series of experiments that give the students hands-on experience in developing real embedded systems where they have to pay attention to constraints such as power and latency. Possible topics include examples from

- control theory
- real time systems
- discrete control
- fault tolerance
- distributed algorithms.

Intended Learning Outcomes: Students will

- fully understand system design engineering principles;
- be able to apply engineering principles for system design;
- understand the differences that embedded system development makes to the application of system design engineering principles;
- be competent in using various methods for specification/modelling, analysis, design, implementation and verification.

Assessment:	Laboratory work (20%) Examination 1 (50%) Coursework 1 (30%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
Lab exercises plus one coursework.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Examination feedback summarising strengths and weaknesses of the class.
Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.

Failure Redemption: Resit examination and/or resubmit coursework(s) as appropriate

Additional Notes: The module has a limited capacity.

Preference is given to students who have chosen modules reflecting the safe and secure systems profile (for details contact the module coordinator.)

Because there is often a waiting list for this module, in order to ensure access is as fair as possible, students will not be permitted to transfer OUT of this module after the first two weeks of teaching in TB2 (because it would be too late for someone else to transfer in to take their place.)

CSCM72 Optimisation

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr AAM Rahat

Format: 30 hours (20 lectures, 10 laboratory hours)

Delivery Method: On Campus Lectures and Labs.

Module Aims: Optimisation is at the core of many disciplines. Whether we want to improve the performance of a machine learning model,
increase the efficiency of an aircraft design, or simply reduce the costs of productions in a business operation, we must deploy computational optimisation methods for achieving the best results. In this module, we will cover mathematical and
algorithmic fundamentals of optimisation, including derivative and derivative-free approaches for both linear and non-linear
problems. We will also discuss advanced topics, such as multi-objective optimisation, handling uncertainty, principled methods when problem evaluations are computationally expensive, and performance comparison between stochastic optimisers, in the context of real-world problems.

Module Content: Lectures:
* Introduction to optimisation.
* Derivatives and related gradient descent methods.
* Bracketing methods.
* Direct methods.
* Stochastic and evolutionary methods.
* Constrained problems.
* Multi-objective optimisation and decision making.
* Model-based methods.
* Optimisation under uncertainty.
* Performance comparison for stochastic optimisers.
Labs programmatically explore optimisation problems and algorithms.

Intended Learning Outcomes: On completion of this module, students will be able to:
* Demonstrate systematic understanding of fundamental concepts of optimisation problems and algorithms.
* Analyse an unseen optimisation problem, and formulate a mathematical description.
* Propose an appropriate method to solve an optimisation problem, and justify their selection.
* Develop appropriate software for solving optimisation problems.
* Critically evaluate performance of multiple competing optimisers, and communicate analysis to specialist and nonspecialist
audiences
* Critically review a relevant topic from the literature.

Assessment:	Examination (60%)
	Coursework 1 (20%)
	Report (20%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Examination. Standard unseen 2 hour Computer Science examination.
Coursework. A practical programming assignment on solving an optimisation problem.
Report. A short critical review on a relevant topic from the literature. [1000-1200 words]
Quiz. A range of multiple choice questions. [Non assessed]

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Individual feedback on coursework and report..

Failure Redemption: 100% Examination Resit Instrument.

Additional Notes:

Available for visiting and exchange students.

CSCM75 Logic in Computer Science

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr U Berger

Format: 20 lectures,
2 x 3 hours practicals,
4 problem consultation hours

Delivery Method: On campus.

Module Aims: This module provides an in-depth introduction to logic and its applications to computer science, as a sound basis for the formal specification and verification of computer programs. Student will also learn how to use an interactive proof tool and carry out interactive proofs themselves.

Module Content: - Propositional logic (syntax, semantics, proof systems of natural deductiuon and resolution)
- Predicate logic (syntax, semantics, proof system)
- Applications of logic to program specification and verification
- Specialised logics e.g. for security protocols, reactive systems and credit card systems

Intended Learning Outcomes: Students will
have a critical understanding of the syntax, semantics and proof rules of first-order predicate logic,
be deeply familiar with other, specialised, logics in computer science (e.g. modal logic, process logic),
critically understand the importance of logic for computer science,
be able to express informal statements as formulas in predicate logic and carry out simple formal proofs.
Students will have used an interactive logic tool to carry out formal proofs of varying difficulty.

Assessment:	Examination (70%)
	Coursework 1 (10%)
	Laboratory work (10%)
	Coursework 2 (10%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs
Coursework consists of two assignments and lab work:

Coursework 1: Syntax and semantics of propositional logic.
Coursework 2: Predicate logic and automated proof search (Resolution).
Lab: Formal proofs in natural deduction using an interactive proof tool.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Examination feedback summarising strengths and weaknesses of the class.
Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.

Failure Redemption: Resit examination and/or resubmit coursework(s) as appropriate

Additional Notes:

Available to visiting students

CSCM77 Computer Vision and Deep Learning

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules: CSC345; CSCM45

Co-requisite Modules:

Lecturer(s): Prof X Xie

Format: 20 hours lectures, 10 hours practicals.

Delivery Method: Primarily on campus

Module Aims: This module introduces students to the important and modern topics and concepts of computer vision and deep learning, including image processing, feature extraction, camera calibration, stereo vision, motion and tracking, recognition, deep neural network and its application to vision problems. It teaches techniques that are used to understand and interpret the contents of images and videos and dissects state-of-the-art vision systems, such as Microsoft Kinect. Practical examples in Matlab are provided throughout the lectures.

Module Content: This course is composed of following parts: Introduction, Image Processing and Video Analysis, Neural networks and Deep Learning.

Introduction: The first part of the lectures gives an overview of Computer Vision and Deep Learning (CVDL) and a roadshow of this course. It also provides a brief revision of basic and important mathematical techniques frequently used in CVDL.

Image processing: filtering, object extraction, segmentation, texture analysis.

Video analysis: camera models and calibration, stereo vision, depth estimation, motion estimation and tracking, local features for tracking.

Neural networks and Deep Learning: feedforward neural networks, back propagation, convolutional neural network, recurrent neural network, and applications.

The module is also accompanied with practical examples in Matlab.

Intended Learning Outcomes: Students will gain an in-depth understanding of the important concepts of computer vision and deep learning, acquire a detailed knowledge of how the analysis of digital images and videos may be performed, and develop the skills necessary to program a basic computer vision system using both conventional and deep learning approaches.

Assessment: Examination 1 (60%)
Coursework 1 (20%)
Coursework 2 (20%)

Resit Assessment: Examination (Resit instrument) (100%)

Assessment Description: The coursework is designed to test the students ability to synthesise a computer vision system to solve a practical real work problem using all the techniques and methods discussed and provided in the module.

The exam is a comprehensive assessment of bookwork, understanding of the computer vision and pattern recognition methods, and the ability to propose solutions to vision problems.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Examination feedback summarising strengths and weaknesses of the class.
Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.

Failure Redemption: Resit exam or resubmit coursework as appropriate.

Additional Notes:

Updated April 2018. This module is available to students studying
MSc Data Science, MSc Advanced Computer Science, MSc Computer Science, and MEng Computing/MSci Computer Science.

Prerequisite: CSCM45/345 Big Data and Machine Learning

CSCM79 Hardware and Devices

Credits: 15 Session: 2022/23 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr DR Sahoo

Format: 10-12 hours lectures, 24 hours lab; 4-6 hours lab assessment.

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: This module encourages students to explore the advanced literature and research results underpinning the field of interaction technologies and ubiquitous user-interface development. Students are expected to achieve a clear view of the ‘cutting edge’ and issues in the field.

Module Content: - Ubiquitous computing and tangible user interfaces
- Interfacing with the real-world using sensors and actuators with Phidgets/Raspberry-Pi/Arduino/Kinect etc.
- Mobile phone sensing, e.g. orientation/location/camera/vibration
- Processing sensor data

Intended Learning Outcomes: Thorough knowledge of variety of hardware and I/O devices.
Ability to build interactive hardware interfaces and programming them.
In-depth knowledge of non-standard devices in various hardware platforms.

Assessment:	Examination 1 (50%) Coursework 1 (25%) Coursework 2 (25%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
Two practical assignments.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Examination feedback summarising strengths and weaknesses of the class.
Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.

Failure Redemption: Resit examination and/or resubmit coursework(s) as appropriate

Additional Notes:

This module can only accommodate a limited students in the lab and the enrollment is on a first-come-first-enrolled basis. The precise number of places is defined in the 'Student Capacity' field of this catalogue entry.

Students who are enrolled in this module MUST have strong programming skills specially in OO programming, and should have android programming experience.

This module is only available for level-M students i.e. MEng/MSci and students enrolled in Advanced Computer Science master programmes. It is also available for vising/exchange students with android programming skills, provided there is still space on the course.

Updated August 2020.

CSCM85 Modelling and Verification Techniques

Credits: 15 Session: 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr U Berger

Format: 20 hours lectures;
2 x 3 practicals;
4 problem consultation hours.

Delivery Method: primarily on campus

Module Aims: This module will give an overview of the landscape and the state of the art of current modelling and verification techniques. One particular tool for software verification will be studied in depth. Students will gain hands-on experience in using that tool.

Module Content: Overview of techniques for formal verification.
Interactive theorem proving, automated theorem proving and model checking.
Introduction to one specific logic for modelling and verification.
Techniques for modelling of software using verification tools.
Practical verification of software examples.

Intended Learning Outcomes: The students will have
- gained an thorough overview of the landscape and the state of the art of current modelling and verification techniques
- acquired a deep understanding of one particular verification tool and know how to translate practical and mathematical problems into its notation
- obtained hands-on experience in practical verification.

Assessment: Examination 1 (70%)
Coursework 1 (15%)
Laboratory work (15%)
Resit Assessment: Examination (Resit instrument) (100%)

Assessment Description: Standard format Computer Science exam (2 hours), and coursework:

Assignment 1: Mathematical and logical foundations of concurrent processes.
Assignment 2: Advanced modelling and verification in the process language CSP.
Lab: Modelling and verification in CSP using the process tools ProBE and FDR.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks.
Examination feedback summarising strengths and weaknesses of the class.
Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.

Failure Redemption: Resit examination and/or resubmit coursework(s) as appropriate

Additional Notes:

Available to visiting students

CSCM98 Operating Systems and Architectures

Credits: 15 **Session:** 2022/23 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr B Mora

Format: 20 hours lectures, 10 hours lab.

Delivery Method: On-campus/virtual lectures and lab sessions.

Module Aims: This module gives an overview of current and future processor architectures, operating systems and basic concurrency problems. It intends to teach most details of the developing environment that must be taken into consideration when developing efficient software

Module Content: * Operating Systems in general (Scheduler, Virtual Memory, Multi-tasking).
- Kernel calls.
- Resource management.
- Memory management.
- Paging and virtual memory.
- File Systems
- Processes and threads management

* Architectures
- Registers+ALU
- Caches, cache lines and cache levels.
- Cache trashing.
- MMU
- TLB
- RAM Latency and throughput
- SIMD units
- SIMD Programming SSE,AVX, AVX-512
- Dedicated processor instructions.

* Concurrency and issues
- Definition of core concepts including race conditions, deadlocks, starvation, critical sections.
- Standard concurrency problems and solutions
- Some standard techniques including software based locks, mutexes and semaphores, atomic instructions, barriers.

* Distributed systems
- Distributed locks.
- Distributed file systems.
- Distributed clocks and time stamping.
- Cloud computing.
- Map/reduce algorithm.

* Security aspects of OS, including:
- Principles of memory protection (virtual memory, randomised stack address, non-executable bit, hypervisor rings).
- Spectre and Meltdown attacks

Intended Learning Outcomes: Students will have a thorough understanding of:
- Current and future processor architectures.
- The role of an Operating System, especially on the multithreading and memory management aspects.
- The issues associated with parallel programming and know some standard solutions.
- How to produce better code when programming parallel architectures.
- Cloud and distributed systems.

Assessment:	Examination 1 (70%)
	Assignment 1 (30%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
Practical assignment.

Moderation approach to main assessment: Second marking as sampling or moderation

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks. Examination feedback summarising strengths and weaknesses of the class. Individual feedback on submissions from lecturer and/or demonstrators in laboratory sessions.
Failure Redemption: Resit exam and/or resubmit assignment as appropriate.
Additional Notes: Available to visiting and exchange students.