Swansea University
Prifysgol Abertawe

FACULTY OF SCIENCE AND ENGINEERING

UNDERGRADUATE STUDENT HANDBOOK

MSC (FHEQ LEVEL 7)

COMPUTER SCIENCE

DEGREE PROGRAMMES

SUBJECT SPECIFIC
PART TWO OF TWO
MODULE AND COURSE STRUCTURE
2023-24

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 23-24 academic year begins on 25 September 2023

Full term dates can be found here

DATES OF 23-24 TERMS

25 September 2023 – 15 December 2023

8 January 2024 – 22 March 2024

15 April 2024 – 07 June 2024

SEMESTER 1

25 September 2023 – 29 January 2024

SEMESTER 2

29 January 2024 – 07 June 2024

SUMMER

10 June 2024 – 20 September 2024

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.

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.

At Swansea University and in the Faculty of Science and 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, technical and administrative staff, administrators - I'm sure you will find many friendly helping hands ready to assist you. And make the most of living and working alongside your fellow students.

During your time with us, please learn, create, collaborate, and most of all – enjoy yourself!

Professor David Smith
Pro-Vice-Chancellor and Executive Dean
Faculty of Science and Engineering

Faculty of Science and Engineering	
Pro-Vice-Chancellor and Executive Dean	Professor David Smith
Director of Faculty Operations	Mrs Ruth Bunting
Associate Dean – Student Learning and Experience (SLE)	Professor Laura Roberts
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	Postgraduate Taught – Dr Bertie Müller
Coordinators	Advanced Computer Science – Dr Anton Setzer Advanced Software Technology – Dr Anton Setzer Computer Science (MSc) – Dr Oliver Kullmann Cyber Security – Dr Pardeep Kumar Data Science – Dr Alma Rahat

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 8.30am-4pm.

The **Student Support Team** provides dedicated and professional support to all students in 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 (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 other resources:

https://myuni.swansea.ac.uk/fse/

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 23-24 handbooks to ensure that you have access to the most up-to-date versions.
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/essentialinfo-taught-students/your-programme-explained/

MSc (FHEQ Level 7) 2023/24
Computer Science (MSc)
MSc Computer Science

Coordinator: Dr O Kullmann

Compulsory Modules

Semester 1 Modules	Semester 2 Modules
CSCM41 **Introduction to Programming in Java** **15 Credits** **Dr T Doan**	**CSCM10** **Computer Science Project Research Methods** **15 Credits** **Dr MJ Roach**
CSCM53 **Computer System Concepts** **15 Credits** **Dr S Yang**	**CSCM12** **Software Concepts and Efficiency** **15 Credits** **Dr C Pradic**
CSCM59 **Relational and Object-Oriented Database Systems** **15 Credits** **Dr W Macinnes**	**CSCM94** **Software Engineering Principles** **15 Credits** **Dr T Owen**
Dissertation	
CS-M20 **MSc Project** **60 Credits** **Dr U Berger** **CORE**	
Total 180 Credits	

Optional Modules

Choose exactly 15 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 one module.

CSCM08	Information Security Management	Dr B Muller	TB1	15
CSCM13	Critical Systems	Dr AG Setzer	TB1	15
CSCM68	Embedded System Design	Dr H Nguyen/Prof SA Shaikh	TB1	15
CSCM85	Modelling and Verification Techniques	Dr U Berger	TB1	15

And

Choose exactly 15 credits

Select one module

CSCM37	Data Visualisation	Dr B Mora/Dr JF Maestre Avila	TB2	15
CSCM39	Human Computer Interaction	Dr MI Ahmad	TB2	15
CSCM64	Software Testing	Dr E Neumann	TB2	15
CSCM68B	Embedded System Design	Dr B Chaparro Rico	TB2	15
CSCM75	Logic in Computer Science	Dr U Berger	TB2	15
CSCM79	Hardware and Devices	Dr DR Sahoo	TB2	15

CS-M20 MSc Project

Credits: 60 Session: 2023/24 September-June

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 of the whole cohort

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.

CSCM08 Information Security Management

Credits: 15 Session: 2023/24 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr B Muller

Format: 30 hours lectures and seminars

Delivery Method: On-campus 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:	Assignment 1 (30%) Examination (70%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Assignment 1: Short report and group presentation/video The exam will be a regular closed-book 2h exam.
Moderation approach to main assessment: Moderation by sampling of the cohort
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: 2023/24 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 lectures

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 (40%) Report (50%) Class Test 1 - Coursework (10%)
Resit Assessment:	Coursework reassessment instrument (100%)

Assessment Description: Initial Project Report - This report will be in the form of a technical academic report and consist of key contents: Project definition and evidence of understanding the challenges of the project.

Project Presentation - Presentation on the project at the Student Conference.

Moderation approach to main assessment: Universal Double Blind Marking of the whole cohort

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 a document addressing unsatisfactory aspects of initial submission(s) in the form of a project specification report or a project reflection report.

Additional Notes:

Only available to students on MSc Computer Science, MSc Advanced Computer Science, MSc Advanced Software Technology.

CSCM12 Software Concepts and Efficiency

Credits: 15 Session: 2023/24 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr C Pradic

Format: 20 hours of lectures and 10 hours of lab sessions

Delivery Method: On campus lectures.

Module Aims: This module provides a solid introduction to algorithm design, complexity analysis and data structure for efficient algorithm development. The module will involve students developing and evaluating their own algorithms. Java will be used as the main programming language.

Module Content: Introduction to the concept of algorithm and program efficiency.
Sorting and searching algorithms.
Top-down and bottom-up approach to algorithm development.
Recursion.
Complexity analysis, comparison and implementation of algorithms and various data structures.
The concept of data abstraction with particular reference to the object-oriented paradigm.
The abstract specification, implementation and complexity of various standard data types, which include array, linked list, stack, queue, tree, heap, graph and hashtable.
Importances of algorithms and data structures, and their applications (e.g. database - indexing, security - hashing and message digest).
The object-oriented language Java will be used for practical implementations.

Intended Learning Outcomes: Students will be able to:
• Develop algorithms to solve programming problems
• Determine the appropriate data structures and data management techniques in a variety of software applications, including distributed applications, and critically evaluate the relative merits of alternatives.
• Formally specify a range of abstract data types for use with the object-oriented and procedural paradigms.
• Implement a range of data structures and data management techniques in different ways, utilising the object-oriented paradigm.
• Determine the efficiency and complexity implications of the different approaches.
• Appreciate the idea of analysing an algorithm to determine its efficiency.
• Implement and analyse standard sorting and searching algorithms.
Students will be aware of how memory is managed and how this can affect program performance.

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

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
Coursework 1 - Algorithms and complexity analysis
Coursework 2 - Practical programming exercise with a focus on data structures

Moderation approach to main assessment: Moderation by sampling of the cohort

Assessment Feedback: Outline solutions provided along with group and individual analytical feedback for courseworks. Oral immediate feedback will be provided shortly after student's oral presentations.
Examination feedback summarising strengths and weaknesses of the class.

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

Additional Notes: Only available to students doing a Computer Science MSc.

CSCM13 Critical Systems

Credits: 15 Session: 2023/24 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: Moderation by sampling of the cohort

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: Available to visiting and exchange students.

CSCM37 Data Visualisation

Credits: 15 Session: 2023/24 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr B Mora, Dr JF Maestre Avila

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: Moderation by sampling of the cohort

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.

CSCM39 Human Computer Interaction

Credits: 15 Session: 2023/24 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 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: Coursework 1 – Paper Critique (Report) 30%
Coursework 2 – Short Paper and Prototype (70%)

Moderation approach to main assessment: Moderation by sampling of the cohort

Assessment Feedback: Individual written feedback.

Failure Redemption: Resubmit coursework as appropriate.

Additional Notes: Available to visiting and exchange students

CSCM41 Introduction to Programming in Java

Credits: 15 Session: 2023/24 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr T Doan

Format: 10 hours lecture, 20 hours lectures in lab environment, 10 hours lab.

Delivery Method: On-campus lectures and lab sessions.

Module Aims: This intensive course provides a solid introduction to the Java programming language and development process.

Module Content: Introduction to Programming. Introduction to Java.
Basic structures of programming. Functions (static methods).
The Object-Oriented paradigm: objects, classes and methods.
Various I/O methods, and input/output via command-line.
Basic correctness for programming (precision, finding and correcting errors).
Aspects of exception handling and defensive programming.

Intended Learning Outcomes: Students will be able to:
Use variables, expressions and control structures, together with elementary input/output facilities.
Understand and properly use functions, static as well as non-static ones (methods).
Understand the foundations of classes, with methods and constructors, complete the foundations of basic programming.
Understand the fundamental concepts of correct and precise programming.
Design and implement working Java programs in a modular maintainable style to the standard required in a Masters degree.
Read code written by others, and to identify errors and bugs and to correct them.

Assessment:	Examination 1 (50%) Coursework 1 (20%) Coursework 2 (20%) Laboratory work (10%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
Coursework 1 and 2 - practical assignments.

Moderation approach to main assessment: Moderation by sampling of the cohort

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 or resubmit coursework as appropriate.

Additional Notes: Only available to Masters students who are not majoring in Computer Science.

CSCM53 Computer System Concepts

Credits: 15 Session: 2023/24 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr S Yang

Format: 30 Lectures

Delivery Method: On-campus and lab sessions.

Module Aims: This module gives an overview of some of the main principles underlying computers and computing from both a theoretical and an applied point of view. It includes a brief history of computers and software, an introduction to the representation of data and the basic components of a computer, the basic features of operating systems, file systems, computer networks, the world wide web, and some basic issues of computer security. A brief discussion on the use of formal methods in specifying computer systems is also given.

Module Content: Brief history of computers and software
Binary values and number systems
Data representation
Logic, gates and circuits
Computing components
Low level programming
Operating system concepts
File systems and directories
Computer networks
The world wide web
Overview of computer security
Introduction to formal methods

Intended Learning Outcomes: After completing this module students will have demonstrated:

- an understanding of how software and hardware interact in a computer system.
- an understanding of the principles behind modern computer architecture, operating systems, file systems, networks and the world wide web.
- an understanding of the critical issues in computer security.
- an ability to apply the fundamental mathematical tools of Computer Science and be able to apply them to writing formal specifications and documentation.

Assessment:	Examination 1 (80%) In class test (non-invigilated) (20%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Examination: duration 2hrs.
Weekly in-class Canvas quizzes.

Moderation approach to main assessment: Moderation by sampling of the cohort

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

Failure Redemption: Resit examination

Additional Notes:

Only available as part of the MSc Computer Science and MSc Applied Data Science degree programmes.

CSCM59 Relational and Object-Oriented Database Systems

Credits: 15 Session: 2023/24 September-January

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr W Macinnes

Format: 20 lectures, 2 x 3 hours lab class,
4 hours problem sessions

Delivery Method: Primarily on campus

Module Aims: This module gives an appreciation of of the complexity of real-world databases. It considers some of the problems that can occor in multi-user, multi-transactions situations. It discusses relational and object-oriented databases and covers their design and implementation. Distributed databases and databases linked to the web will also be discussed, as will data warehousing and data mining. Students will gain practical experience in designing and implementing a database.

Module Content: A review of the nature of data and databases and an overview of database management and
database system architecture.
Data models: relational databases, object databases.
Relational databases: the structure of the relational model, integrity constraints, relational algebra and calculus, normalisation.
Transaction management, data security and recovery, optimisation, distributed databases, concurrency control.
Object-oriented databases, type inheritance, active databases, temporal databases, logic-based databases.
Data warehouses and data mining, data visualisation.
Web technology and databases.

Intended Learning Outcomes: After completing this module students will have demonstrated:
- a comprehensive understanding of the principles of relational databases and object-oriented databases.
- the ability design and implement databases according to best practice.
- the ability to normalise a database and explain why it is necessary.
- the ability to explain the additional problems associated with distributed databases and the need to ensure integrity and how to control concurrency.
- the ability to explain how data warehouses operate and the principles of data mining.
- how to link databases to web applications.

Assessment:	Examination 1 (70%) Coursework 1 (20%) Laboratory work (10%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs. All questions should be attempted.
Practical assignment. Students will be required to build and interrogate a database according to specifications.
Laboratory sessions. Students will demonstrate key skills.

Moderation approach to main assessment: Moderation by sampling of the cohort

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 assignment as appropriate

Additional Notes: Available to visiting and exchange students

CSCM64 Software Testing

Credits: 15 Session: 2023/24 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr E Neumann

Format: 20 hours lectures; 10 hours practicals.

Delivery Method: On-campus 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 format Computer Science exam, 2hrs.
Two practical programming/testing exercises.
Reflective report.
Weekly small in-class tests.

Moderation approach to main assessment: Moderation by sampling of the cohort

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.

CSCM68 Embedded System Design

Credits: 15 Session: 2023/24 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 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: Moderation by sampling of the cohort

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.)

CSCM68B Embedded System Design

Credits: 15 Session: 2023/24 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr B Chaparro Rico

Format: 12 lectures + 18 lab sessions

Delivery Method: On-campus 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:	Examination 1 (50%) Coursework 1 (10%) Coursework 2 (10%) Coursework 3 (30%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Standard Computer Science format unseen examination, duration 2hrs.
Coursework 1: Online Canvas Assignment
Coursework 2: Online Canvas Assignment
Coursework 3: Online Canvas Quiz

Moderation approach to main assessment: Moderation by sampling of the cohort

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.)

CSCM75 Logic in Computer Science

Credits: 15 Session: 2023/24 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%) Coursework 2 (10%) Laboratory work (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: Moderation by sampling of the cohort

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

CSCM79 Hardware and Devices

Credits: 15 Session: 2023/24 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 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
- Internet of Things and Edge Computing
- Interfacing with the real-world using sensors and actuators with Phidgets/Raspberry-Pi/Arduino/ESP32 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%)
	Assignment 1 (10%)
	Assignment 2 (25%)
	Report (10%)
	Laboratory work (5%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Examination 1: Standard Computer Science format unseen examination, duration 2hrs.
Assignment 1: Mid-term demonstration of group project - during 6th week.
Assignment 2: End-term demonstration of group project - during 11th week.
Report: Report of group project - during 12th week.
Laboratory work: Five practical programming assignments - 2nd, 3rd, 4th, 7th and 8th weeks.

Moderation approach to main assessment: Moderation by sampling of the cohort

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: Students enrolled in this module must be able to program using object-oriented, event-driven programming to accommodate the areas of embedded systems, mobile, or web (e.g., Python, Java, JavaScript, Android, etc).

This module can only accommodate a limited number of students in the lab and the enrollment is on a first-come-first-enrolled basis. Therefore, this module is only available on selected pathways and is not available for visiting/exchange students.

CSCM85 Modelling and Verification Techniques

Credits: 15 Session: 2023/24 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: Moderation by sampling of the cohort

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

CSCM94 Software Engineering Principles

Credits: 15 Session: 2023/24 January-June

Pre-requisite Modules:

Co-requisite Modules:

Lecturer(s): Dr T Owen

Format: 15 hours lectures and 15 hours problems sessions

Delivery Method: On-campus lectures and lab sessions.

Module Aims: Students will be introduced to the principles of software development and the main professional issues associated with its practice. They will also develop a significant piece of software in teams.

Module Content: Introduction to the principles of professional software development.
Software process models (Waterfall etc).
Object-oriented software design (UML) and implementation
Review of legal, ethical, social, preliminary security, and professional issues applicable to the computer industry.
Software project management, planning, and risks
Software Testing.
Team work and time management.

Intended Learning Outcomes: Students will have:-
• appreciation of the different approaches to software development;
• become fully aware of the major professional issues associated with software development;
• worked together on a significant software project;
• become fully familiar with many aspects of managing software projects;
• knowledge of quality assurance and testing methods for software systems.

Assessment:	Examination 1 (50%)
	Coursework 1 (20%)
	Coursework 2 (30%)
Resit Assessment:	Examination (Resit instrument) (100%)

Assessment Description: Individual Coursework 1 - Design and partial implementation assignment
Group Work - Coursework - Design and implementation of a significant piece of software
Two hour examination.

Moderation approach to main assessment: Moderation by sampling of the cohort

Assessment Feedback: Written feedback for courseworks is provided electronically. Collective exam feedback is provided electronically as well

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

Additional Notes: Normally only available to students on MSc Computer Science.