



**De Montfort University,
Faculty of Computing
Engineering Media and
Niels Brock, Copenhagen
Business College**

**BSc (Hons)
Computer Science**

G40051

**Curriculum
2023/24**

The aim of the undergraduate degree, Computer Science, is to provide the students with modern approaches to software systems development. A mixture of both theory and practice is covered with an emphasis on "why" as well as "what". Students are encouraged to develop critical thinking and problem-solving skills. A practical element gives students the experience needed to develop software systems using modern languages and environments. An example of this is the final year project which gives students the opportunity to apply all stages of a software development method to produce a software system, guided by a project tutor. As well as all the technical aspects of the course, students learn about the structure of organisations, gaining an insight into the commercial context in which IT systems are commissioned and deployed. Students are also introduced to computing ethics and law.

The first year's syllabus focusses on basic concepts of Computer Science like Computer architecture, Operating systems, Computer networks, Programming, Security etc. The second year's syllabus focuses on more advanced concepts like Object-oriented software design and programming, Data structures and algorithms, Concurrent and parallel algorithms, Agile team project development etc. The third year is a specialisation year covering concepts like Big data and machine learning, Software design methods (including commercial, professional, ethical and legal considerations) and with a possibility to select an elective course covering "up-to-date" concepts like Functional Programming, Web Development, Fuzzy Logic and Data protection. The three years are organised in 4 courses each year, each running in seven-week blocks.

Upon completion of the undergraduate degree, students should be able to:

- Apply the theoretical knowledge and practical experience they have learnt to each phase of the software lifecycle.
- Be able to describe the role of computer technology within business organisations.
- Evaluate and recommend appropriate computer systems.
- Recommend an appropriate implementation strategy.
- Demonstrate knowledge and understanding of areas in computing at the forefront of the discipline.
- Appreciate the relative merits and limitations of different computing environments, paradigms and methodologies.

The subject specific aims are listed in the following module descriptions.

In addition, participants will also gain a number of cognitive skills. These include:

- Capacity for appreciation of the complexity of Computer Science
Critical evaluation and reflection of theoretical and practical issues
- Professional considerations related to both developing systems and implementing these in businesses
- Capacity for independent and self-managed learning
- Ability to draw reasoned conclusions
- Communication skills, be they in writing, as oral presentations etc.

The programme structure is set out in detail below.

Level 4 (Study Year 1)

Core modules:

Module code	Module title	Credit Value
CTEC1701	Database Design and Implementation	30
CTEC1702	Fundamental Concepts of Computer Science	30
CTEC1703	Computer Programming	30
CTEC1704	Operating Systems and Networks	30

Supplementary (but still mandatory) module:

Module code	Module title	Credit Value
NB001	Mandatory Academic Workshop	0

Please note that all level 4 modules must be passed before you may proceed to level 5

Level 5 (Study Year 2)

Core modules:

Module code	Module title	Credit Value
CTEC2710	Object Oriented Design and Development	30
CTEC2711	Data Structures and Algorithms	30
CTEC2712	Web Application Development	30
CTEC2713	Agile Development Team Project	30

Please note that all level 5 modules must be passed before you may proceed to level 6

Level 6 (Study Year 3)

Core modules:

Module code	Module title	Credit Value
CTEC3701	Software Development: Methods and Standards	30
CTEC3702	Big Data and Machine Learning	30
CTEC3451	Development Project	30

Please note that the modules CTEC3701 and CTEC3701 must be passed before you hand in the development project (CTEC3451)

Plus select one of the following:

Module code	Module title	Credit Value
CTEC3704	Functional Programming	30
CTEC3705	Advanced Web Development	30
IMAT3722	Fuzzy Logic and Inference Systems	30
IMAT3711	Privacy and Data Protection	30

The modules are described in detail below

Level 4 (Study Year 1) Module Descriptions

NB001: Mandatory Academic Workshop

The Academic Workshop is a non-credit but highly rewarding module. The Academic Workshop module is an opportunity for the student to acquire and develop academic knowledge skills that will support your entire study.

The primary objective of the academic workshop module is to reinforce students' practical assignment writing and presentation skills. Therefore, there will be an emphasis on the different assessment methods students will meet during the study, especially Case Study and Portfolio, since there are specific requirements for these but also Essay where students will have to choose a question and build scientific writing (including proper referencing) around that.

In so doing the module enables learners to think about their academic writing and presentation style at an early stage of the process. The generic support provided by the module is additional to the assignment specific instructions students will get from their module tutors.

CTEC1701: Database Design and Implementation

Structured data, held in relational databases, accessed via SQL, supports the information storage requirements of many companies, organisations, and on-line businesses. In this module the student will learn the fundamentals of how to design the structure of data within a relational database, how to interact with data within the database, and how to protect the data within the database. Topics include: The relational model, top-down modelling of business requirements, ER model, keys, relationships, traps, normalisation, SQL mapping schema to implementation via DDL, querying data using DML, integrity, transactions, access control and security, introduction to big data, semi-structured and unstructured data. Relevant elements of the Data Protection Act 2018 (DPA 2018) and the General Data Protection Regulations 2016 (GDPR) will be introduced, developing knowledge and understanding of the various 'rights and freedoms' that they provide. This will build to develop knowledge of a Data Protection Impact Assessment (DPIA), which is now a legal requirement whenever a proposed data processing need is perceived as 'risky' such as, inter alia, mass surveillance, use of sensitive data and automated decision making.

CTEC1702: Fundamental Concepts of Computer Science

This module introduces students to fundamental concepts in computer science in relevant areas of mathematics (including propositional logic, set notation, etc); software modelling; the software lifecycle; requirements capture; user interface design; and the foundations of ethical thinking. These topics can then be applied and further developed as students progress throughout the course. The module starts by introducing mathematical structures that provide a basis for computer science including logic, set theory, probability, and statistics. More specifically, logic sets, tuples, relations, functions, probability, hypothesis generation and testing, and matrices are covered. After this, students are taught about fundamental principles in areas of the software life cycle, including requirements capture and user interface design. Developing an understanding of requirements analysis will include how to formulate functional requirements, the various kinds of non-functional requirements, and usability requirements, as well as how to carry out interviews and questionnaires and formulate appropriate questions. Discussions of the software lifecycle include the range of different activities involved in a software development project and some different ways in which these activities can be sequenced, as well as the central importance of designing around user needs and capabilities and integrating user consultation and user testing into the

development process. Students then move onto developing their understanding in topics surrounding the foundations of ethical thinking. There is an introduction to ethical theories and concepts related to information systems, information security, software engineering, computer science and digital forensics. This includes examining the key issues affecting the development of future ICTs (privacy, security, etc). The material is complemented by understanding the responsibilities of the computer professional, including professional codes of conduct and codes of ethics (e.g. IEEE, ACM, BCS).

CTEC1703: Computer Programming

Computer programming requires the analysis of a problem, the production of requirements, and their translation into a design that can be executed on a computer. The design phase in particular requires the identification and combination of appropriate programming abstractions. This module introduces the skills required to develop a computer program to solve a given problem and does so from the perspective of designing trustworthy software with an emphasis on sound coding principles and unit testing. Outline content: - Practical program design using the control and data abstractions in a contemporary programming language. - Highlighting techniques and approaches from different computer programming paradigms. - The importance of good programming practice and the relevance of coding standards. - The role of problem analysis and program specification across different computer programming paradigms. - The use of functions in computer programming design and the production of unit tests. - The role of testing in the software development process. During the module students should be made aware of important principles of developing trusted software including, e.g., naming conventions, initialisation of structures and variables, variable scope and lifetime, validation of input. Students should also be made aware of the consequences of poor programming style and technique (i.e. poor maintainability, poor security and vulnerability to attack). Later in the module students will be exposed to the importance and benefit of using functions within their program design and will then utilise a key principle of trustworthy software by using an industry standard unit testing framework. Students will also consider different approaches to identifying and solving a problem in the context of a contemporary programming language.

CTEC1704: Operating Systems and Networks

This module is designed to provide a foundation in computer architecture, operating systems, and computer networks. Outline content:

1. Theoretical foundations: a. Number systems, integer and real number representation b. Finite state automata; Introduction to data encryption
2. Computer hardware: CPU components and operation, instruction sets, Computer architectures, Memory
3. Systems software: Operating system fundamentals, processes, file systems, memory management, Shell scripting, Authentication and authorisation.
4. Computer networks: a. Network architectures, data communication system fundamentals, Transmission schemes and technologies, error detection and management, Network components, LAN protocols, Internet protocols.
5. Security issues: a. Information security: confidentiality, integrity and availability (CIA), Network vulnerability and security, threats and attacks b. Operating system vulnerability and security, threats and attacks, Architectural vulnerability and security

Level 5 (Study Year 2) Module Descriptions

CTEC2710: Object Oriented Design and Development

This module focuses on Object-Oriented (OO) library and application development. Library development

will enable students to design, implement, and test medium scale software systems using an object-oriented approach. Meanwhile, application development will use extensive library packages provided by the Java SDK so that students are comfortable in navigating and making use of a variety of domains such as Collections, Input/Output and Graphical User Interfaces. The design notation used is the Unified Modelling Language (UML) and the implementation language is Java. It is essentially a programming module, with the emphasis on implementing OO designs and producing reusable libraries. Students enrolling on this module will have a foundation in programming gained from the study of a level 4 programming module. This module introduces UML and the Java language. Fundamental OO concepts and design issues are discussed. Emphasis is placed on the design and structure of software, and the OO development process. UML is used to document designs, and the concept of software design patterns are introduced. Contemporary areas of the API will be used to showcase how OO applications can benefit from more recent functional additions such as lambda expressions and stream pipelines. Students will be required to build graphical user interfaces and consider associated features such as layout policies, observable data models, and binding events to properties. The use of advanced areas of the API also allows a variety of design patterns such as composite, strategy and decorator to be discussed and deployed. Software quality, reliability, and maintainability are integral to the development of software, and are integrated into the delivery by considering different approaches to solving common problems in application design. In particular, encapsulation, decomposition, and decoupling are viewed within the context of the model-view-controller (MVC) architecture. These design issues will commonly link to advised techniques of delivery as outlined in the PAS 754:2014 software trustworthiness specification. By the end of the module students should have become more independent learners with the ability to adapt their existing knowledge and learn additional software libraries and features.

CTEC2711: Data Structures and Algorithms

This module introduces a variety of data structures and algorithms for both sequential and parallel execution. Classical data structures will be introduced (including stacks, queues, lists, trees, and hash tables) and algorithms for searching and sorting. The performance characteristics of these data structures and algorithms will be explained. Specific coding issues will also be considered such as modularity, genericity, quality, assignment, mutable and immutable objects. Later in the module students will be introduced to concurrent program design in the context of multi-core architectures and distributed applications where appropriate formal notation will be used for specification

CTEC2712: Web Application Development

This module provides a thorough grounding in the rapidly evolving area of web technologies. With equal focus on user interface design on the 'client-side' or 'front-end' and on security and persistence in 'server-side' or 'back-end' scripting. The module covers crucial design principles, information architecture and usability factors as well as standards compliance, accessibility, authentication/authorisation and security. This exciting field has been driven by advances in web standards, in particular HTML5 (introduced in 2008 and accepted as a 'living standard' in May 2019), modular CSS (~2012) and the ECMAScript standard (ECMAScript2015 in particular) which together define the modern web platform. The web standards process enables the platform to evolve extremely quickly, and new features are released with increasing regularity. Modern web applications typically make heavy use of server-side scripting using development languages such as PHP. This pragmatic language is used to great effect by some web developers and with catastrophic naivety by others. Web application solutions are typically implemented by teams of developers. The various roles typically undertaken will be discussed with students.

CTEC2713: Agile Development Team Project

This module is an opportunity for students to engage in a constrained work-place simulation based on agile software development. Students working in teams of 3 to 5 will initially identify a system of sufficient size to be distributed equally among all members. Work allocation will be monitored under the guidance of their tutor/supervisor. For example, each team member might take individual ownership of the development of 2-3 classes from initial inception to completion providing CRUD functionality. In the case of a large system this may mean that some aspects of the system are never built to completion. By the end of the module each team will be expected to develop an integrated software component based on their individual work. This would typically be an administrative dashboard allowing for maintenance of system data. No specific language is named for the module; however, it would be wise within a single team to select a family of languages/development environments aligned with the prior experience of members based on their taught programme, ideally this should also align with the team's tutor/supervisor. Projects will need to be carefully sourced to match this range of skills. In house projects will also be available. The assessment will be designed to encourage collaboration, peer learning, and formative feed-forward assessment. The module will include supporting materials introducing concepts and practice relating to agile development however beyond that there is no formal taught content as this module aims to consolidate learning and skills from prior and concurrent study. It is expected that students will use an appropriate set of tools for collaboration for example git- hub. The ethical component will focus on software design and development and risk factors associated with the management of these projects e.g. the pitfalls and problems; why do software development projects fail? There will also be a focus on examining ethical approaches for software development (co-creation, collaboration, stakeholder engagement, etc). Additionally, data collection technologies and approaches, processing and re-use, storage, security, and data minimisation will be considered in the context of why protection of personal and business data is an ethical issue.

Level 6 (Study Year 3) Module Descriptions

CTEC3701: Software Development: Methods and Standards

This module immerses the students in the methodological, regulation environment in which software systems are developed, This is achieved by exploring for types of application development: module, AI, robotic process automation and games systems. This application classes provide the basis for exploring methodological approaches, with a particular emphasis on current agile approaches, in particular Scrum and DevOps. The different needs and emphasis of different applicant classes are considered. Within these context the methodological evolution required in scaling, in embedding in security by design and in integration are considered. The module incorporates the context of standards with particular reference to standards in security (ISO27001) and risk management (ISO27005). Practical exercises include an engagement with current agile project management tools (e.g. Jira and Clickup). Students develop an understanding of a range of agile and traditional methodologies and consider the debates, ambiguity and uncertainty in their application. The module considers legal and ethical aspects of software development. The ethical approach addresses responsible development and embedding sustainability through a considering of green IT issues. Ethics studies include intellectual property (who owns your data?), copyright, patents and trademarks, and the ethics of ownership (NFTs etc). The digital divide is covered (e.g. economic, social, political, cultural, etc) understanding the issues and solutions (including SDG's). Plus, topics surrounding responsible research and innovation (RRI) and how it can begin to address the ethical issues of technological development. The law component of the module focusses on three key areas – computer misuse, professional negligence, and intellectual property law. Relevant elements of the

Computer Misuse Act 1990 (CMA 1990), the civil tort of negligence (and how it applies to professionals) and both the criminal law and civil law aspects of Intellectual Property will be introduced, developing knowledge and understanding of the various obligations.

CTEC3702: Big Data and Machine Learning

The module will focus on machine learning and its application to Big Data in a "taster-like" fashion. That is, ML will be applied to solve analytics problems using appropriate tools e.g., Apache Spark that avail ML libraries. As this is done ML algorithms will be introduced and then applied. The focus is therefore not so much on the technical details of the algorithms - rather, the ability to implement them and use them within analytics. The module covers supervised and unsupervised learning techniques with a specific application to data mining. Selected classification, regression, and clustering approaches will be examined. Algorithm evaluation and evaluation metrics will be explored, and machine learning frameworks and tools introduced. The module also covers Big Data, Big Data analytics, Machine learning frameworks e.g., Apache Spark, machine learning libraries e.g., Spark Machine Learning Library (MLlib), and Hadoop Distributed File System. Additionally, the module considers ethics in relation to AI, big data, and surveillance.

CTEC3451: Development Project

The project provides students with the opportunity to carry out a significant piece of work that reflects the aims and outcomes of their specific programme. It provides students with the opportunity to demonstrate practical and analytical skills present in their programme of study; to work innovatively and creatively; to synthesise information, ideas, and practices to provide a quality solution, together with an evaluation of that solution. The project should meet some real need in a wider context.

Students will demonstrate an ability to self-manage a significant piece of work, and will undertake a self-evaluation of the process. Students will be expected to demonstrate appropriate and proactive project management, and written/verbal presentation skills throughout the period of the project. As well as analysing, designing, delivering and appraising a product of suitable quality, they will be expected to undertake research, analysis, design, implementation, verification, evaluation and reporting pertinent to the project.

Indicative Content: The range of projects will be wide. Projects are obtained from a variety of sources including: internal academic proposals, external organisation suggestions, and a number from students themselves. The deliverables will include:

1. A main report which will include:

- elucidation of the problem and the objectives of the project
- an in-depth investigation of the context and literature, and where appropriate, other similar products
- where appropriate, a clear description of the stages of the life cycle undertaken
- where appropriate, a description of how verification and validation were applied at these stages
- where appropriate, a description of the use of tools to support the development process
- a critical appraisal of the project, indicating the rationale for any design / implementation decisions, lessons learnt during the course of the project, and evaluation (with hindsight) of the project outcome and the process of its production (including a review of the plan and any deviations from it)
- a description of any research hypothesis
- references

2. A set of appendices that are referred to within the main report, and which contain the substantive work on the project, including product deliverables, such as requirements and design specifications and other project documents (project contract, informed consent, ethics review form etc.).

3. A product demonstration shortly after the submission of the main report.

CTEC3704: Functional Programming

Functional Programming (FP) is a mature software development paradigm that is for teaching, research, and industrial software development. Throughout the 2010s, and subsequently, the use of FP has grown significantly in industry, and most mainstream programming languages have adapted to include support for fundamental FP concepts. One reason for the growth of FP is that the paradigm makes it easier to develop code for concurrent execution on multiple-core machines. The removal of shared mutable state reduces the dependence on traditional locks and improves scalability; and the replacement of strict, sequential processing by computations that can be distributed automatically across multiple processes, allows for greater levels of optimisation. (Pure) functional programming is based upon firm mathematical foundations. The use of referentially transparent functions and immutable data makes it possible for techniques such as equational reasoning to be applied to program fragments that facilitate their translation, automatically, into more efficient formulations. A familiarity with the foundations, and some of the techniques that can be used to reason formally about functional programming constructs, will help the programmer to apply these constructs effectively. Programming without the use of mutable state can present a challenge to a programmer who is already familiar with traditional, non-functional programming techniques. However, the benefits that accrue from this style of software development are worthy of serious investigation by any contemporary software engineer. This module provides you with the fundamental concepts of FP and looks at how these have been provided within a modern programming language. You will gain practical experience, using a modern programming language to solve a practical problem using FP techniques. The core principles are transferrable between functional programming languages.

CTEC3705: Advanced Web Development

The module builds on the outcomes of the level 5 module – Web Application Development. Many modern computer services are now accessed via the ubiquitous web-browser, and users have come to expect instant and secure access to information on a wide range of platforms. Underpinning these web systems is usually a web application, providing a channel to data stored in databases. However, increasingly the website has also become a point of entry for unauthorized access to stored data. This is often the result of poor web application design and/or implementation. The module considers how a web application may be designed and implemented in such a way as to reduce the likelihood of unauthorised access to information. This also requires an understanding of the more common forms of browser-based attacks and the coding techniques that can be used to defend against these. The module aims to further develop key concepts and techniques for designing, evaluating and implementing interactive web applications. Designing user interfaces that users can understand immediately and learn easily, enabling them to carry out tasks smoothly and efficiently without excessive effort or stress, is a crucial part of software development. Failures of design can cause technically successful systems to fail in practical use. User interface development frequently eats a large chunk of the development budget, and large projects employ many user interface design and user experience specialists - and systems analysts and technical developers need to be able to talk to them - while many non-specialist software developers find themselves needing to tackle interface design problems. Designing successful interactive systems involves understanding and applying the key principles of designing usable systems, but also understanding the characteristics of the user populations, the nature of their tasks and environments, and which of the many different aspects of usability are important for this system and this task, and considering trade-offs between different aspects of usability. But successful interactive system design goes beyond usability: it involves considering the user experience as a whole including how graphic design and system behaviour influences user emotions, and

how the interactive system itself is integrated into the user/customer/client's experience of the entire organization. Design is only one half of the coin - testing and evaluating prototypes of interactive systems is a critical part of building systems without major usability problems and achieving a good user experience. The module therefore provides a thorough grounding in the rapidly evolving area of full-stack web development, incorporating front-end web technologies, back-end server-side scripting, and data persistence techniques. The module also considers how information can be accessed and presented from remote sources via web-service protocols.

IMAT3722: Fuzzy Logic and Inference Systems

Fuzzy logic is a mathematical model for handling uncertainty, it is able to provide a means in order to successfully inference from abstract and subjective notions. Fuzzy logic adopts the perspective that the world and humanistic understanding are inherently vague and not precise. Concepts like that of; hot; cold; near; far; and other forms of expressive language where precise values are not given, are extremely difficult to model when universal understanding of such concepts are non-existent. What is beautiful to some, may not be beautiful to others; concepts can have different meanings to different people. Fuzzy logic and fuzzy theory provide the tools in order to fuzzify abstract notions so that they can be modelled and inferred in a humanist manner, such that they can be understood by a larger population. The utilisation of fuzzy components ultimately allows for the creation of a fuzzy inference system, a system based on the thought and decision making processes adopted by human cognition. The use of fuzzy sets; a fuzzy inference engine; and knowledge base, creates for an incredibly powerful tool. Fuzzy inference systems are extremely versatile and can be deployed on many different domains and have been utilised by industry in many different sectors. This module will present the core and fundamental concepts of fuzzy logic, from theory to application. The understanding developed will allow for a fuzzy perspective to be adopted, understood and appreciated. The ability to create specialised fuzzy inference systems will be achieved and so too will the ability to articulate on thought processes needed to create such systems. A comprehensive understanding of fuzzy logic, theory and application will also be covered. The module will also investigate the literature on fuzzy and its areas of application to further instil the applicability of a fuzzy approach and the ethical implications of modelling subjective perception based uncertainty.

IMAT3711: Privacy and Data Protection

It is envisaged that this module will be of interest to students taking a significant number of business, computing or engineering modules. Students that have a background appreciation of philosophy, politics and or sociology would be expected to be able to contribute to and benefit from the module. There continues to be a growth of databases holding personal and other sensitive information in multiple formats including text, pictures and sound. The scale of data collected, its type and the scale and speed of data exchange have all changed with the advent of ICT. Whilst the potential to breach privacy continues to increase organisations are subjected to a considerable amount of legislation governing privacy and data protection. This module examines the balance between maintaining business effectiveness, legal compliance and professional practice in the field of IT/IS. The module will- 1. address the legal, social and technological aspects of privacy and data protection, 2. consider privacy enabling technologies and privacy invasive technologies 3. identify and evaluate the role of the computer professional in providing privacy and data protection. Care will be taken in ensuring perspectives from different cultures and countries are highlighted and considered in the light of global information systems.

Assessment Methods

Assessment Methods 2023/2024 BSc in Computer Science

Modules are assessed in many different ways as outlined in the DMU module guides for each particular module, but here are some of the most common methods of assessment:

- Essay** a written assignment based on a set question (or choice of questions) with a word limit.
- Case Study** an analysis of a current issue/case surrounding the associated topics delivered on the module.
- Report** a structured assignment using headings and sub-headings used to look at a particular problem or issue and make recommendations within a word limit. This could be an individual piece of work or group work.
- Portfolio** a written piece of work where students are asked to reflect on their development and experience and what they have learned from it.
- Phase Test** a shorter test (usually multi-choice or short answers) which takes place under exam conditions – typically online.
- Practical** an individually distinct work where the student demonstrates ability to do practical work related to the course subject, e.g., writing working program code in a programming course
- Presentation** this can be in groups or done individually and usually takes place in a classroom or lecture theatre using visual aids such as PowerPoint.

Emphasis is on formative feedback when it comes to coursework to enable the student to critically reflect on own work and integrate feedback in future assignments.

Code	Module Title	Aim	Assessment
CTEC1701	Database Design and Implementation	Structured data, held in relational databases, accessed via SQL, supports the information storage requirements of many companies, organisations, and on-line businesses. In this module the student will learn the fundamentals of how to design the structure of data within a relational database, how to interact with data within the database, and how to protect the data within the database.	Phase test 1 (40%) Phase test 2 (40%) Report 1000 words (20%)
CTEC1702	Fundamental Concepts of Computer Science	This module introduces students to fundamental concepts in computer science in relevant areas of mathematics (including propositional logic, set notation, etc); software modelling; the software lifecycle; requirements capture; user interface design; and the foundations of ethical thinking. These topics can then be applied and further developed throughout the course.	Phase test 1 (50%) Portfolio (30%) Case study (20%)
CTEC1703	Computer Programming	Computer programming requires the analysis of a problem, the production of requirements, and their translation into a design that can be executed on a	Practical (40%) Phase test 1 (30%) Phase test 2 (30%)

		computer. This module introduces the skills required to develop a computer program to solve a given problem and does so from the perspective of designing trustworthy software with an emphasis on sound coding principles and unit testing.	
CTEC1704	Operating Systems and Networks	This module is designed to provide a foundation in computer architecture, operating systems, and computer networks. Covering theoretical foundations, computer hardware, systems software, computer networks and security issues.	Phase test 1 (30%) Phase test 2 (30%) Phase test 3 (40%)
CTEC2710	Object Oriented Design and Development	This module focuses on Object-Oriented (OO) library and application development. Library development will enable students to design, implement, and test medium scale software systems using an object-oriented approach. Meanwhile, application development will use extensive library packages provided by the Java SDK so that students are comfortable in navigating and making use of a variety of domains such as Collections, Input/Output and Graphical User Interfaces.	Portfolio (50%) Programming specification (50%)
CTEC2711	Data Structures and Algorithms	This module introduces a variety of data structures and algorithms for both sequential and parallel execution. Classical data structures will be introduced (including stacks, queues, lists, trees, and hash tables) and algorithms for searching and sorting. The performance characteristics of these data structures and algorithms will be explained. Specific coding issues will also be considered such as modularity, genericity, quality, assignment, mutable and immutable objects.	Phase test 1 (25%) Phase test 2 (25%) Practical (50%)
CTEC2712	Web Application Development	This module provides a thorough grounding in the rapidly evolving area of web technologies. With equal focus on user interface design on the 'client-side' or 'front-end' and on security and persistence in 'server-side' or 'back-end' scripting. The module covers crucial design principles, information architecture and usability factors as well as standards compliance, accessibility, authentication/authorisation and security.	Phase test 1 (40%) Web Project (60%)
CTEC2713	Agile Development Team Project	This module is an opportunity for students to engage in a constrained work-place simulation based on agile software development. Students working in teams of 3 to 5 will initially identify a system of sufficient size to be distributed equally among all members. Each team member might take individual ownership of the development of 2-3 classes from initial inception to completion providing CRUD functionality.	Portfolio (80%) Case study 1000 words (20%)
CTEC3701	Software Development: Methods and Standards	This module immerses the students in the methodological, regulation environment in which software systems are developed, by exploring types of application development: module, AI, robotic process automation and games systems. With emphasis on current agile approaches, in particular Scrum and	Structured Applications Proposal (60%) Case Study Analysis, 1000 words (20%) Report, 1000 words (20%)

		DevOps, students develop understanding of a range of agile and traditional methodologies and engage with agile project management tools (e.g. Jira and Clickup). The module incorporates the context of standards with particular reference to standards in security (ISO27001) and risk management (ISO27005).	NB Der mangler 40%
CTEC3702	Big Data and Machine Learning	The module will focus on machine learning (ML) and its application to Big Data in a “taster-like” fashion. That is, ML will be applied to solve analytics problems using appropriate tools e.g., Apache Spark that avail ML libraries. As this is done ML algorithms will be introduced and then applied. The focus is therefore not so much on the technical details of the algorithms but rather the ability to implement them and use them within analytics. The module covers supervised and unsupervised learning techniques with a specific application to data mining.	Phase Test (30%) Problem Specification (50%) Case Study Analysis, 1000 words (20%)
CTEC3451	Development Project	This project provides students with the opportunity to demonstrate practical and analytical skills present in their programme of study; to work innovatively and creatively; to synthesise information, ideas, and practices to provide a quality solution, together with an evaluation of that solution.	Initial Submission (25%) Final Submission (75%)
CTEC3704	Functional Programming	This module provides you with the fundamental concepts of FP and looks at how these have been provided within a modern programming language. You will gain practical experience, using a modern programming language to solve a practical problem using FP techniques. The core principles are transferrable between functional programming languages.	Phase test (20%) Theory paper (30%) Practical (50%)
CTEC3705	Advanced Web Development	The module aims to further develop key concepts and techniques for designing, evaluating and implementing interactive web applications. Providing a thorough grounding in the rapidly evolving area of full-stack web development, incorporating front-end web technologies, back-end server-side scripting, and data persistence techniques. The module also considers how information can be accessed and presented from remote sources via web-service protocols.	Phase test (40%) Web project (60%)
IMAT3722	Fuzzy Logic and Inference Systems	Fuzzy logic is a mathematical model for handling uncertainty, it is able to provide a means in order to successfully inference from abstract and subjective notions. Fuzzy logic adopts the perspective that the world and humanistic understanding are inherently vague and not precise. This module will present the core and fundamental concepts of fuzzy logic, from theory to application. The ability to create specialised fuzzy inference systems will be achieved and so too will	System and Technical Report, 2500 words (100%)

		the ability to articulate on thought processes needed to create such systems.	
IMAT3711	Privacy and Data Protection	There continues to be a growth of databases holding personal and other sensitive information in multiple formats including text, pictures and sound. The scale of data collected, its type and the scale and speed of data exchange have all changed with the advent of ICT. Whilst the potential to breach privacy continues to increase organisations are subjected to a considerable amount of legislation governing privacy and data protection. This module examines the balance between maintaining business effectiveness, legal compliance and professional practice in the field of IT/IS.	Essay, 2500 words (50%) Group presentation (50%)