

SEVERN BUSINESS COLLEGE

EAI Bachelor of Science in Computer Science

Course Handbook

Qualification
EAI Bachelor of Science in Computer Science

Level
6

Total Learning Hours
4500

ECTS
180

Course Structure

Bachelor of Science in Computer Science			
Unit number	**Units**	**Unit level**	**ECTS**
	SEMESTER 1		
BBA601	Fundamentals of IT and Computers	5	6
BBA602	Computer and Network Technologies	5	6
BBA603	Database Management System	5	6
BBA604	Web and Mobile Application	5	6
BBA605	Principal of Computer Programming	5	6
	SEMESTER 2		
BBA606	Software Engineering	5	6
BBA607	OOPS with Java	5	6
BBA608	Management Information Systems	5	6
BBA609	Network Information Systems	5	6
BBA610	Cyber Security	5	6
	SEMESTER 3		
BBA611	Python Primer: An Introduction to Programming with Python	5	6
BBA612	Mathematics for Computing	5	6
BBA613	Unlocking Big Data: Technologies and Strategies	5	6
BBA614	Introduction to Cryptography	5	6
BBA615	Exploring the Nexus: Data Science and Artificial Intelligence	5	6
	SEMESTER 4		
BBA616	Exploring Data Protection and IT Security Measures	5	6
BBA617	Understanding Information Security Standards	5	6
BBA618	Exploring the Internet of Things (IoT)	5	6
BBA619	Machine Learning: supervised learning and unsupervised learning	5	6
BBA620	Data Structure & Algorithm	5	6
	SEMESTER 5		
BBA621	Exploring the Fundamentals of Web Security	6	6
BBA622	Mathematical Modelling	6	6
BBA623	Introduction to Quantum Computing	6	6
BBA624	System Analysis and Designing	6	6
BBA625	Overview of Block chain and its security	6	6
	SEMESTER 6		
BBA626	IT Project Management	6	6
BBA627	E-Commerce	6	6
BBA628	Knowledge Management	6	6
BBA629	Capstone Project	6	12

UNIT SPECIFICATIONS

Unit Title
Fundamentals of IT and Computers

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
The Fundamentals of IT and Computers module provides students with a comprehensive understanding of the basic concepts and principles of information technology and computer systems. This foundational module covers essential topics including computer hardware and software components, operating systems, networking concepts, data management, and cybersecurity fundamentals. Students develop practical skills in using various software applications and tools, alongside understanding the role of IT in organisational contexts. Through hands-on activities, interactive learning, and practical exercises, learners gain the knowledge and competencies necessary to effectively navigate and utilise technology in their academic and professional pursuits, preparing them for further study in information systems and technology-related fields.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Demonstrate proficiency in using common software applications, including word processing, spreadsheets, and presentation tools, to produce professional documents.
- Explain the basic components and functions of computer hardware and software, demonstrating understanding of how they interact within a system.
- Analyse different operating systems and their functionalities, including installation, configuration, and troubleshooting.
- Assess various networking concepts, including types of networks, protocols, and security measures, to evaluate their impact on organisational communication.
- Implement basic data management practices, including database creation, querying, and reporting, using appropriate software tools.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The key components of computer systems, including hardware, software, and networking elements.
- Fundamental concepts of operating systems and their role in managing computer resources.
- Principles of data management, including data types, storage, and retrieval methods.
- The importance of cybersecurity measures and best practices in protecting information systems.
- Various types of software applications used in business environments and their functionalities.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to operate common software applications (e.g., word processors, spreadsheets, and presentation software) to create, edit, and format documents effectively.
- Proficiency in configuring basic hardware components and peripherals, demonstrating hands-on skills in setting up computer systems.
- Competence in navigating operating systems efficiently, including file management, system settings, and troubleshooting basic issues.
- Ability to utilise networking tools to establish simple network connections and troubleshoot common network problems.
- Skills in executing basic data management tasks, including data entry, query formulation, and report generation using database software.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Applying critical thinking skills to solve basic IT-related problems and troubleshoot common software and hardware issues.
- Demonstrating effective digital communication skills using various software applications to collaborate on projects.
- Evaluating technology solutions for their applicability in specific organisational contexts and workflows.
- Developing proficiency in using online resources and databases for research and information retrieval.
- Working collaboratively in teams to complete IT-related tasks, showcasing effective time management and organisational skills.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Navigating and utilising various operating systems (e.g., Windows, macOS, Linux) effectively for everyday computing tasks.
- Implementing basic cybersecurity practices, including the use of antivirus software, firewalls, and secure passwords to protect digital information.
- Employing data visualisation tools to analyse and present data effectively using software applications like Excel or Google Sheets.
- Using cloud-based storage solutions (e.g., Google Drive, Dropbox) for file management and collaboration on digital projects.
- Creating and managing digital content across various platforms, ensuring proper formatting and accessibility standards are met.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including pre-recorded lectures, interactive reading materials, and live online sessions for real-time discussions and demonstrations.
Supervised Placement and Practice Hours: Guided hands-on activities and practical exercises supervised by instructors, providing opportunities to apply IT skills and receive feedback.
Self-Study Hours: Independent learning activities including review of materials, completion of online tutorials, practice exercises, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.
Pedagogy

This module is delivered through a blended online learning approach combining asynchronous and synchronous learning methods:

- Pre-Recorded Lectures: Introduction to fundamental IT concepts, computer hardware and software, operating systems, networking, data management, and cybersecurity principles, accessible at students' own pace.
- Interactive Reading Materials: Digital resources, tutorials, and guided readings supporting understanding of core IT concepts and practical applications.
- Live Online Sessions: Real-time discussions, demonstrations, and Q&A sessions providing opportunities for direct interaction with instructors and peers.
- Hands-On Activities: Practical exercises allowing students to apply skills in software applications, hardware configuration, operating system navigation, networking setup, and data management tasks.
- Collaborative Projects: Group activities encouraging teamwork using shared digital tools and platforms, developing collaborative working skills in online environments.
- Regular Assessments and Feedback: Formative and summative assessments including quizzes, assignments, and examinations, with timely feedback supporting continuous learning and improvement.

The pedagogical approach emphasises practical skill development, active learning, and application of IT fundamentals to real-world scenarios.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%
- Final Examination: 30%

Assessment Methods

Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific IT topics, demonstrating understanding of hardware and software concepts, operating systems, networking principles, data management, or cybersecurity practices. This assesses their ability to articulate technical concepts clearly and apply theoretical knowledge to practical scenarios.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key IT concepts, terminology, and principles covered throughout the module. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of fundamental IT concepts, including computer systems, operating systems, networking, data management, cybersecurity, and software applications. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

All assessments are conducted using digital learning tools such as online assessment platforms and learning management systems.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Tanenbaum, A.S. & Austin, T. (2013). *Structured Computer Organization* (6th ed.). Pearson.
- Stallings, W. (2020). *Computer Organization and Architecture: Designing for Performance* (9th ed.). Pearson.
- Morley, S. & Parker, C.S. (2021). *Understanding Computers: Today and Tomorrow* (17th ed.). Cengage Learning.
- Schmidt, B. (2022). *Introduction to Information Technology* (5th ed.). Cengage Learning.
- Severance, C. (2020). *Python for Everybody: Exploring Data in Python 3*. CreateSpace Independent Publishing Platform.

Additional Reading

- Kurose, J.F. & Ross, K.W. (2021). *Computer Networking: A Top-Down Approach* (8th ed.). Pearson.
- Rainer, R.K. & Turban, E. (2020). *Introduction to Information Systems: Supporting and Transforming Business* (6th ed.). Wiley.
- Patterson, D.A. & Hennessy, J.L. (2017). *Computer Organization and Design: The Hardware/Software Interface* (5th ed.). Morgan Kaufmann.
- Lindsay, B. (2022). *Digital Literacy: A Primer for Librarians and Educators*. Rowman & Littlefield.
- Mackenzie, A. (2021). *Introduction to Computer Science: A Problem-Solving Approach* (3rd ed.). Springer.

UNIT SPECIFICATIONS

Unit Title
Computer and Network Technologies

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview

This module provides learners with comprehensive understanding of computer networking essentials and cloud technologies, their operating principles, protocols, standards, security considerations, and practical implementations. Learners explore different hardware and software options and develop skills in configuring and installing network systems. The module examines a wide range of networking technologies including Local Area Networks (LAN) and Wide Area Networks (WAN), tracing their evolution to create large-scale networks. Protocol methodologies related to IP data networks are explored in depth, alongside cloud computing models, client-server architectures, routing and switching techniques, and network security practices. Through theoretical instruction, hands-on laboratory work, and practical exercises, students gain the knowledge and technical competencies required to design, implement, troubleshoot, and manage modern computer networks and cloud infrastructures.
Learning Outcomes

Competences

Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Summarise the historical development of cloud computing technologies, demonstrating the ability to guide discussions on their evolution and impact on modern computing practices.
- Evaluate different cloud computing models and platforms, taking responsibility for assessing their advantages and disadvantages in various organisational contexts.
- Differentiate between client and server environments, and assess the benefits of serverless computing, ensuring the ability to advise stakeholders on suitable deployment options.
- Describe network standards, protocols, and topologies, while collaborating with peers to produce comprehensive reports on their relevance in today's networking landscape.
- Configure networks using routing and switching techniques, monitoring performance to ensure adherence to established standards and best practices.
- Manage projects related to network design and implementation, demonstrating the ability to carry out tasks effectively while authorising changes as necessary to optimise performance.

Knowledge

Upon successful completion of this module, learners will have engaged with:

- The fundamentals of cloud computing and its historical development, demonstrating understanding of its evolution and impact on modern technology.
- Networking principles, including network standards, protocols, and topologies, to facilitate effective communication within computer networks.
- Hardware and software components of computer networks, including switches, routers, gateways, and firewalls, ensuring comprehensive understanding of network infrastructure.
- PC assembling skills, basic troubleshooting techniques, and operating system installation, showcasing practical competencies in hardware management.
- Routing and switching techniques, along with network security practices, while ensuring the ability to conduct router backup and restoration effectively.

Skills

Upon successful completion of this module, learners will have developed:

- Capability to support and troubleshoot computing problems effectively in a variety of scenarios.
- Proficiency in assembling PCs and installing client and server operating systems.
- Competence in configuring client systems to meet organisational policies and requirements.
- Ability to configure networks using routing and switching techniques to ensure efficient data flow.
- Skills in setting up secure network interfaces and backing up router configurations to maintain network integrity.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Supporting and troubleshooting computing problems in diverse technical environments.
- Assembling PCs and installing client and server operating systems efficiently.
- Configuring networks using routing and switching techniques to optimise network performance.

Module-Specific Digital Skills and Competences

Learners will develop digital skills and competencies including:

- Proficiency in cloud computing technologies and platforms (e.g., AWS, Azure, Google Cloud).
- Understanding of networking principles, standards, and protocols (e.g., TCP/IP, DNS, DHCP).
- Competence in configuring network hardware and software, including routers, switches, and firewalls.
- Ability to troubleshoot network issues and implement security measures to protect network infrastructure.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering theoretical concepts and principles, supplemented by demonstrations and examples.
Supervised Placement and Practice Hours: Hands-on laboratory sessions providing opportunities for learners to apply knowledge and skills in configuring and troubleshooting computer networks under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of online tutorials, practice with simulation software, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction and practical application in a virtual learning environment (VLE):

- Lectures: Introduction to theoretical concepts and principles of computer networking, cloud computing, network protocols, standards, topologies, hardware components, and security considerations.
- Hands-On Laboratory Sessions: Practical exercises providing opportunities to apply knowledge and skills in configuring, installing, and troubleshooting computer networks, routing and switching equipment, and cloud platforms.
- Demonstrations and Examples: Instructor-led demonstrations of network configuration, troubleshooting techniques, and best practices.
- Digital Learning Tools: Use of online forums for discussion and collaboration, simulation software for network modelling and testing, and virtual labs for practical exercises in safe, controlled environments.
- Interactive Activities: Collaborative projects and peer learning activities enabling learners to work together on network design and implementation challenges.

The pedagogical approach emphasises practical skill development, problem-solving, and application of networking and cloud computing concepts to real-world scenarios.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific networking or cloud computing topics, such as network design proposals, evaluation of cloud platforms, analysis of network protocols, or security considerations. This assesses their ability to articulate technical concepts, apply theoretical knowledge to practical scenarios, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key networking concepts, cloud computing principles, protocols, standards, hardware components, and troubleshooting techniques. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of computer and network technologies, including networking principles, cloud computing models, routing and switching, network security, and practical troubleshooting skills. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, simulation software, and virtual labs will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to virtual labs, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Stallings, W. (2021). *Computer Networking with Internet Protocols and Technology*. Pearson.
- Kurose, J.F. & Ross, K.W. (2021). *Computer Networking: A Top-Down Approach*. Pearson.

Additional Reading

- Mullen, J. (2022). *Networking for Beginners: A Comprehensive Guide to Networking*. CreateSpace Independent Publishing Platform.
- Cisco Press (2021). *CCNA 200-301 Official Cert Guide Library*. Cisco Press.
- Beck, S. (2023). *Networking Essentials: A Comprehensive Guide to Networking Concepts*. Wiley.

UNIT SPECIFICATIONS

Unit Title
Database Management System

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module covers the fundamental concepts of Database Management Systems (DBMS), including Structured Query Language (SQL) and PL/SQL for database creation, manipulation, and querying. Learners explore essential aspects of database design, including normalisation, data modelling using class diagrams, and the creation of data dictionaries. The module examines forms and reports design, data clustering and partitioning techniques, and the essentials of database administration including security, backup, recovery, and management of distributed databases. Through theoretical instruction, hands-on laboratory work, and practical exercises, students develop the knowledge and technical competencies required to design, implement, query, and administer relational database systems effectively in organisational contexts.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Demonstrate a thorough understanding of the advantages and components of Database Management Systems (DBMS).

- Apply the principles of class diagrams and data types to design normalised tables and create comprehensive data dictionaries.
- Execute and test complex queries using Data Definition Language (DDL), Data Manipulation Language (DML), joins, subqueries, and grouping commands.
- Develop effective forms, reports, and procedural languages for data retrieval, storage, and error handling.
- Manage distributed databases, ensuring robust database administration, security, backup, and recovery processes.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The core concepts and advantages of DBMS, including feasibility studies and integrity constraints.
- Normal forms and the process of converting class diagrams into normalised tables.
- The syntax and functions of SQL and PL/SQL for creating and manipulating databases.
- The principles of designing forms and reports, including layout and the use of graphical objects.
- The principles and components of distributed databases, including client/server architecture and object-oriented databases.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to design and implement normalised tables from class diagrams and create comprehensive data dictionaries.
- Proficiency in writing and testing complex SQL queries involving multiple tables, subqueries, joins, and group operations.
- Competence in designing and creating interactive forms and reports, integrating procedural languages for data manipulation and error handling.
- Ability to develop and manage application structures with user interface features, custom reports, and efficient data storage methods.
- Skills in performing essential database administration tasks, including development stages, security management, backup, recovery, and handling distributed databases.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- DBMS Fundamentals: Understanding advantages, components, and feasibility of database systems in various scenarios.
- Database Design: Creating class diagrams, defining data types, and applying normalisation techniques to ensure data integrity.
- SQL Skills: Mastering SQL queries (basic, computational, subqueries, joins), DDL, and DML commands for effective database management.
- Forms and Reports: Designing effective layouts, integrating graphics, managing interactive data input, and ensuring robust error handling.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Ability to write and execute SQL queries for data retrieval, manipulation, and management across relational databases.
- Skills in creating database schemas, defining tables, relationships, and applying normalisation principles.
- Competence in using DML commands (INSERT, UPDATE, DELETE) to manipulate data within databases effectively.
- Proficiency in designing and implementing forms and reports using database query results and reporting tools.
- Knowledge of implementing error-handling mechanisms and maintaining data integrity through validation rules and constraints.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**

Component	Hours
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including live lectures and interactive video sessions covering SQL queries, database design, normalisation, forms and reports, and database administration.
Supervised Placement and Practice Hours: Hands-on learning sessions using virtual labs to practise tasks such as query writing, database schema creation, form and report design, and error handling implementation under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of online tutorials, practice with SQL environments, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy
This module is delivered through a structured blended learning approach:

- Live Lectures: Introduction to fundamental DBMS concepts, database design principles, SQL and PL/SQL syntax, normalisation techniques, forms and reports design, and database administration practices.
- Interactive Video Sessions: Guided demonstrations of SQL query writing, database schema creation, data manipulation, and report generation.
- Hands-On Virtual Labs: Practical exercises enabling learners to apply knowledge in real database environments, practising query writing, form design, error handling implementation, and database administration tasks.
- Assignments and Quizzes: Regular assessments evaluating understanding of DBMS concepts, SQL proficiency, and database design principles.
- Discussion Forums and Q&A Sessions: Collaborative learning opportunities providing spaces for peer interaction, clarification of concepts, and problem-solving support.

The pedagogical approach ensures students gain both practical skills and theoretical knowledge essential for effective database management in various organisational contexts.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific database management topics, such as database design proposals, normalisation exercises, evaluation of DBMS approaches, or analysis of database security and backup strategies. This assesses their ability to articulate technical concepts, apply database design principles, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key DBMS concepts, SQL syntax and functions, normalisation principles, forms and reports design, and database administration practices. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of database management systems, including database design, SQL query writing, PL/SQL programming, forms and reports development, and database administration tasks. The examination will include a mix of multiple-choice, short-answer, SQL coding exercises, and problem-solving questions.

Digital learning tools such as online submission platforms, simulation software, and virtual database labs will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to virtual labs and database environments, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Elmasri, R. & Navathe, S.B. (2021). *Fundamentals of Database Systems*. Pearson.
- Date, C.J. (2021). *Database System Concepts*. McGraw-Hill.
- Rob, P. & Coronel, C. (2021). *Database Systems: Design, Implementation, & Management*. Cengage Learning.
- Tharam, S. & Gopalan, R. (2022). *Database Management Systems: A Practical Approach*. Springer.
- Garcia-Molina, H., Ullman, J.D. & Widom, J. (2021). *Database Systems: The Complete Book*. Prentice Hall.

Additional Reading

- Tharam, S. & Gopalan, R. (2022). *Database Management Systems: A Practical Approach*. Springer.
- Garcia-Molina, H., Ullman, J.D. & Widom, J. (2021). *Database Systems: The Complete Book*. Prentice Hall.

UNIT SPECIFICATIONS

Unit Title
Web and Mobile Application

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module provides learners with comprehensive understanding of current web and mobile application design technologies, practices, and tools. Learners explore modern development frameworks, hosting solutions, content management systems, and integrated development environments used in creating websites and mobile applications. The module covers essential topics including website hosting and management, domain name systems, communication protocols, mobile application development methodologies, server-side scripting, database integration, search engine optimisation (SEO), and regulatory and ethical considerations in multimedia content creation. Through theoretical instruction, hands-on workshops, and practical exercises, students develop the knowledge and technical competencies required to design, develop, test, deploy, and manage websites and mobile applications that meet client specifications, user requirements, and industry standards.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Explain modern web and mobile development technologies, frameworks, and hosting solutions.
- Evaluate the impact of development technologies and frameworks on design, functionality, and search engine ranking.
- Review the importance of website design on search engine results and apply search engine optimisation (SEO) techniques to improve website ranking.
- Manage the development and deployment of mobile applications within an Integrated Development Environment (IDE), ensuring alignment with functional and technical requirements.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Website hosting and management principles, including domain names, DNS services, and communication protocols.

- Mobile application development methodologies and navigation of mobile application stores such as the Apple App Store and Google Play Store.
- The interaction between browsers and servers, server technologies, and integrated database systems for expanding website capability.
- Common web development technologies, frameworks, and SEO techniques to enhance website performance.

Skills
Upon successful completion of this module, learners will have developed:
- Capability to develop user interaction schemes influenced by characteristics of good design.
- Proficiency in creating dynamic content using server-side scripting languages such as PHP and interfacing with databases.
- Competence in utilising content management systems (CMS) for website design and management.
- Ability to apply regulatory and ethical considerations in creating multimedia content for websites.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:
- Designing and developing websites and mobile applications to fulfil client and user requirements effectively.
- Evaluating tools, techniques, and languages used for website and mobile application design and development.
- Creating wireframe documents and test plans for websites and mobile applications to guide development processes.
- Analysing the results of test plans and evaluating the overall success of websites and mobile applications against defined criteria.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:
- Proficiency in using website development tools and software such as Adobe Photoshop, Illustrator, and Dreamweaver.
- Knowledge of content management systems (CMS) such as WordPress, Joomla, or Drupal for website development and management.
- Familiarity with integrated development environments (IDEs) for mobile application development, such as Android Studio or Xcode.
- Understanding of regulatory and ethical considerations in multimedia content creation for websites, including accessibility standards and data protection.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering theoretical concepts and principles of web and mobile application development.
Supervised Placement and Practice Hours: Hands-on workshops and practical exercises conducted in virtual and physical environments, providing opportunities for application of website and mobile application design techniques using digital tools and software under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of online tutorials, practice with development tools, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, programming projects, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction and practical application:

- Lectures: Introduction to theoretical concepts and principles of web and mobile application development, including development technologies, frameworks, hosting solutions, SEO techniques, mobile application methodologies, and regulatory considerations.
- Hands-On Workshops: Practical sessions enabling learners to apply development techniques using digital tools, software applications, content management systems, and integrated development environments.
- Practical Exercises: Guided activities providing opportunities to create websites and mobile applications, implement server-side scripting, integrate databases, design user interfaces, and apply SEO strategies.
- Digital Tools and Software: Use of website development tools (Adobe Photoshop, Illustrator, Dreamweaver), CMS platforms (WordPress, Joomla), IDEs (Android Studio, Xcode), and collaboration software for project development.
- Project-Based Learning: Development of web and mobile application projects from design specifications through to testing and deployment, providing authentic learning experiences.

The pedagogical approach emphasises practical skill development, creative problem-solving, and application of web and mobile development principles to real-world scenarios.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Programming/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific web or mobile application development topics, such as evaluation of development frameworks, analysis of SEO strategies, assessment of hosting solutions, or examination of mobile application development methodologies. This assesses their ability to articulate technical concepts, evaluate development approaches, and develop evidence-based recommendations.

Programming/Mini Project (30%):
Learners will develop a web or mobile application project demonstrating practical application of development skills, including design, implementation, database integration, testing, and deployment. This assesses their capability to apply technical knowledge, create functional applications, and meet specified requirements.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of web and mobile application development, including development technologies, frameworks, hosting principles, SEO techniques, mobile application methodologies, server-side scripting, database integration, and ethical considerations. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as CMS platforms, IDEs, online submission platforms, and collaboration software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, revision and resubmission of the programming project, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to development tools and software, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Welling, L. & Thomson, L. (2021). *PHP and MySQL Web Development*. Addison-Wesley.
- Duckett, J. (2022). *HTML and CSS: Design and Build Websites*. Wiley.
- Rouse, M. (2021). *JavaScript: The Definitive Guide*. O'Reilly Media.
- Meiklejohn, A. (2022). *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. O'Reilly Media.

Additional Reading

- McFarland, D.S. (2021). *HTML & CSS: Visual QuickStart Guide*. Peachpit Press.
- Nygard, M. (2021). *Release It: Design and Deploy Production-Ready Software*. Pragmatic Bookshelf.

- Szyperski, C. (2021). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.
- Haverbeke, M. (2021). *Eloquent JavaScript: A Modern Introduction to Programming*. No Starch Press.
- Murdock, M. (2023). *The Art of Readable Code: Simple and Practical Techniques for Writing Better Code*. O'Reilly Media.

UNIT SPECIFICATIONS

Unit Title
Principles of Computer Programming

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module provides learners with a foundational understanding of object-oriented programming languages and the principles of producing effective, efficient code. Learners gain perspective on software development processes and the fundamental principles of algorithms, including algorithm design, efficiency, and complexity analysis. The module covers essential concepts including abstraction, the KISS principle, open/closed entities, coupling, cohesion, and various algorithmic techniques such as brute force, greedy algorithms, and dynamic programming. Through theoretical instruction, practical programming exercises, and hands-on projects, students develop the knowledge and technical competencies required to design, implement, debug, and evaluate object-oriented software solutions that address real-world computational problems effectively and align with business and user requirements.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Critically assess how programming practices enhance operational efficiency, foster innovation, and address community challenges.
- Show proficiency in outlining key historical milestones and trends in programming, articulating their significance in shaping modern computing.
- Exhibit competence in defining algorithms and detailing common techniques, such as sorting and searching, to solve computational problems effectively.
- Evaluate the benefits of object-oriented design and explain the use of objects in object-oriented programming.
- Design, implement, and debug object-oriented software solutions, showcasing technical programming skills and problem-solving capabilities.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Foundational programming concepts such as abstraction, the KISS principle, open/closed entities, coupling, and cohesion.
- The evolution of algorithms, focusing on techniques like brute force, greedy algorithms, and dynamic programming.
- Various object-oriented programming languages and their applications in software development.
- Different software development methodologies and their effectiveness in various contexts.
- The significance of algorithm efficiency and complexity in practical programming scenarios.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to analyse computer programming principles and software development methodologies critically.
- Proficiency in developing software programs to solve problems using object-oriented programming, including compiling and debugging code.
- Competence in designing flowcharts to illustrate problem solutions and creating software solutions based on defined user requirements and system designs.
- Ability to evaluate software against business and user requirements to ensure effectiveness.
- Skills in communicating programming concepts and solutions effectively to diverse audiences.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Critically analysing the benefits and principles of computer programming, as well as evaluating different software development techniques and methodologies.
- Problem-solving through the creation of software programs to solve real-world problems using object-oriented programming principles.
- Communicating effectively by summarising and explaining programming principles, algorithms, and software development methodologies to technical and non-technical audiences.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in programming languages such as Java, JavaScript, C#, C++, Python, PHP, and Ruby on Rails.
- Competence in debugging and compiling code using Integrated Development Environments (IDEs).
- Ability to design and implement algorithms to solve complex computational problems.
- Skills in creating flowcharts to visualise algorithmic solutions and program logic.
- Familiarity with various software development tools and methodologies, including version control systems, testing frameworks, and deployment techniques.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering theoretical concepts and principles of computer programming, supplemented by practical demonstrations and examples.
Supervised Placement and Practice Hours: Tutorials and practical exercises providing opportunities for hands-on programming practice, individualised support, and application of knowledge and skills to real-world scenarios under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of programming exercises, practice with development tools, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, programming projects, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction and practical application in a virtual learning environment (VLE):

- Lectures: Introduction to theoretical concepts and principles of computer programming, object-oriented design, algorithms, software development methodologies, and programming best practices.
- Tutorials: Structured sessions providing opportunities for hands-on programming practice, individualised support, code review, and problem-solving guidance.
- Practical Exercises: Guided programming activities allowing learners to apply knowledge and skills to real-world scenarios, including algorithm implementation, software development, and debugging.
- Digital Learning Tools: Use of online forums for discussion and collaboration, video conferencing for live instruction and support, and collaborative software development platforms for code sharing and peer learning.
- Project-Based Learning: Development of programming projects demonstrating application of object-oriented principles, algorithm design, and software development practices.
- The pedagogical approach emphasises practical skill development, critical thinking, and application of programming principles to solve computational problems effectively.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%

- Programming/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific computer programming topics, such as analysis of programming principles, evaluation of software development methodologies, examination of algorithmic techniques, or assessment of object-oriented design approaches. This assesses their ability to articulate technical concepts, critically evaluate programming practices, and develop evidence-based arguments.

Programming/Mini Project (30%):
Learners will develop a programming project demonstrating practical application of object-oriented programming principles, including problem analysis, algorithm design, code implementation, compilation, debugging, and testing. This assesses their capability to apply technical knowledge, create functional software solutions, and meet specified requirements.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of computer programming principles, including object-oriented programming concepts, algorithms, software development methodologies, code efficiency, and problem-solving techniques. The examination will include a mix of multiple-choice, short-answer, code analysis, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories (e.g., GitHub), and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
- Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, revision and resubmission of the programming project, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to development tools and programming environments, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts
- Fowler, M. (2020). *Refactoring: Improving the Design of Existing Code* (2nd ed.). Addison-Wesley Professional.
- Sweigart, A. (2020). *Automate the Boring Stuff with Python* (2nd ed.). No Starch Press.
- Somasegar, S. & Silver, A. (2021). *The Complete Software Developer's Guide* (2nd ed.). Penguin Random House.
- Robson, B. & Freeman, E. (2021). *Head First Design Patterns* (2nd ed.). O'Reilly Media.
- Richter, J. (2021). *CLR via C#* (5th ed.). Microsoft Press.

Additional Reading
- Griffiths, D. & Griffiths, D. (2021). *Head First Kotlin: A Brain-Friendly Guide*. O'Reilly Media.
- Shroff, G. (2021). *The Innovation Arc: Building Enduring Businesses for a Changing World*. MIT Press.
- Gookin, D. (2020). *C Programming for Dummies* (2nd ed.). Wiley.
- McFedries, P. (2020). *Coding with JavaScript for Dummies*. Wiley.
- Cass, S. (2020). *Programming Arduino: Getting Started with Sketches* (3rd ed.). McGraw Hill.

UNIT SPECIFICATIONS

Unit Title
Software Engineering

Level
5

Learning Time Hours

150

Credit Value

6

Module Overview

This module provides learners with comprehensive understanding of software development and its evolution as an engineering discipline, focusing on the principles, methodologies, and practices required to develop, maintain, and evolve software systems of high quality. Learners explore software design using modelling languages such as Unified Modeling Language (UML) and Extensible Markup Language (XML), object-oriented programming principles and practices, software testing strategies and tools, and project management approaches. The module covers essential topics including system architecture design, structural modelling, comprehensive testing management, code quality assurance, modularity, reusability, and software documentation. Through theoretical instruction, practical programming sessions, and project-based learning, students develop the knowledge and technical competencies required to design, implement, test, and maintain robust, scalable, and maintainable software solutions that meet business requirements and industry standards.

Learning Outcomes

Competences

Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Apply UML and XML to ensure quality, code reuse, flexibility, and modularisation in software development, demonstrating expertise in structured design.
- Implement comprehensive test strategies using tools such as Bugzilla, LoadRunner, and Jira, ensuring software quality throughout the development cycle.
- Develop object-oriented programs tailored to business requirements, showcasing the ability to integrate advanced programming concepts like collections and generics.
- Ensure thorough testing and documentation of software, maintaining high standards of accuracy and functionality.

Knowledge

Upon successful completion of this module, learners will have engaged with:

- The advantages of using modelling languages, such as UML, in system design for improving system architecture and clarity.
- The process of designing and implementing UML class diagrams, demonstrating knowledge of structural modelling.
- Software testing management, covering strategic approaches and tools for ensuring software reliability.
- The principles of object-oriented programming and the ability to develop solutions that adhere to these methodologies.

Skills

Upon successful completion of this module, learners will have developed:

- Capability to design and implement UML class diagrams to model system architecture effectively.
- Proficiency in creating object-oriented programs that meet specified requirements, applying advanced concepts like inheritance and polymorphism.
- Competence in applying various software testing strategies using tools such as Bugzilla, LoadRunner, and Jira to ensure code quality and functionality.
- Ability to implement modular and reusable code structures through UML and XML, enhancing software flexibility and maintainability.
- Skills in testing and debugging object-oriented software solutions to meet business needs, ensuring functionality and reliability.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Proficiency in utilising UML and XML for software design and modelling.
- Application of diverse testing strategies and tools for comprehensive software quality assurance.
- Development of object-oriented programs meeting specific business requirements, integrating collections, generics, and best practices.
- Effective use of Bugzilla, LoadRunner, Jira, and similar tools for project management and testing.

Module-Specific Digital Skills and Competences

Learners will develop digital skills and competencies including:

- Utilising integrated development environments (IDEs) and software tools to optimise workflows in software development.
- Expertise in version control systems such as Git for effective collaboration and code management.
- Skills in employing automated testing frameworks and continuous integration tools to enhance development efficiency.
- Proficient use of cloud-based development platforms for deploying scalable software solutions.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including theoretical lectures covering essential concepts and methodologies in software engineering, providing foundation for understanding the discipline.
Supervised Placement and Practice Hours: Practical sessions allowing students to apply theoretical knowledge through hands-on activities, such as software design using UML, coding exercises, and collaborative group projects developing software solutions under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of programming exercises, practice with development and testing tools, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy
This module is delivered through a blend of theoretical instruction, practical application, and project-based learning:

- Theoretical Lectures: Introduction to essential concepts and methodologies in software engineering, including software development lifecycle, UML and XML modelling, object-oriented design principles, software testing strategies, and quality assurance practices.
- Practical Sessions: Hands-on activities allowing students to apply theoretical knowledge through software design using UML, object-oriented programming exercises, implementation of design patterns, and use of testing tools.
- Project-Based Learning: Collaborative group projects offering opportunities to develop software solutions from requirements analysis through design, implementation, testing, and documentation, reinforcing understanding of software engineering principles in real-world contexts.
- Guest Lectures: Industry professionals providing practical insights, exposing students to current trends, best practices, and challenges in software engineering.
- Digital Learning Tools: Use of integrated development environments (IDEs), version control systems (Git), testing tools (Bugzilla, LoadRunner, Jira), online forums, and video conferencing for instruction, collaboration, and code review.

The pedagogical approach emphasises practical skill development, collaborative learning, and application of software engineering principles to create high-quality software solutions.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific software engineering topics, such as analysis of software development methodologies, evaluation of testing strategies, examination of UML modelling approaches, or assessment of object-oriented design principles. This assesses their ability to articulate technical concepts, critically evaluate software engineering practices, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key software engineering concepts, UML notation, object-oriented

programming principles, testing methodologies, and software quality assurance practices. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of software engineering, including software development methodologies, UML and XML modelling, object-oriented programming, design patterns, testing strategies, quality assurance, and project management. The examination will include a mix of multiple-choice, short-answer, UML diagram analysis, code analysis, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories (e.g., GitHub), and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to development and testing tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Martin, R.C. (2020). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Pearson.
- Sommerville, I. (2021). *Software Engineering* (10th ed.). Pearson.
- Booch, G., Maksimchuk, R. & Engle, M. (2022). *Object-Oriented Analysis and Design with Applications* (4th ed.). Addison-Wesley.
- Gaddis, T. (2021). *Starting Out with Java: From Control Structures through Objects* (7th ed.). Pearson.
- Pressman, R.S. (2022). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.

Additional Reading

- Freeman, E. & Robson, E. (2021). *Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software* (2nd ed.). O'Reilly Media.
- Fowler, M. (2022). *Refactoring: Improving the Design of Existing Code* (2nd ed.). Addison-Wesley.
- Sierra, K. & Bates, B. (2022). *Head First Java* (3rd ed.). O'Reilly Media.
- Schildt, H. (2021). *Java: The Complete Reference* (12th ed.). McGraw-Hill Education.
- Horstmann, C.S. (2021). *Core Java Volume I–Fundamentals* (12th ed.). Pearson.

UNIT SPECIFICATIONS

Unit Title
Object-Oriented Programming with Java

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
Operational information systems are commonly structured and programmed using an object-oriented approach. This module emphasizes the acquisition of fundamental skills in object-oriented programming through theoretical concepts and practical application using the Java programming language. Learners explore essential object-oriented principles including classes, objects, inheritance, polymorphism, encapsulation, abstraction, constructors, exception handling, and interfaces. The module covers the Java programming language fundamentals, including data types, variables,

operators, control structures, packages, visibility modifiers, and unit testing strategies. Through theoretical instruction, practical programming exercises, hands-on sessions, and collaborative projects, students develop the knowledge and technical competencies required to design, implement, test, and maintain robust object-oriented software applications that are flexible, maintainable, and meet business requirements.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Create objects using constructors, demonstrating understanding of standard constructors, constructor overloading, and inheritance.
- Handle exceptions effectively in Java, identifying typical scenarios, understanding standard exceptions, and defining custom exceptions.
- Utilise interfaces as programming interfaces in Java, understanding typical scenarios and implementing them effectively.
- Implement polymorphism and encapsulation in Java, showcasing the ability to design flexible and maintainable code.
- Develop comprehensive unit tests for Java applications, ensuring code quality and reliability through effective testing strategies.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The basic principles and phases involved in object-oriented development processes.
- How to structure problems using classes, define attributes and methods, and establish associations between classes.
- The Java programming language, including basic class elements, attributes, methods, and the main method as the starting point of a Java program.
- Primitive data types, variables, operators, expressions, control structures, packages, and visibility modifiers in Java.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to apply the basic principles and phases involved in object-oriented development processes.
- Proficiency in structuring problems using classes by defining attributes and methods and establishing associations between classes.
- Competence in implementing Java programming language features, including basic class elements, attributes, methods, and the main method as the starting point of a Java program.
- Ability to utilise primitive data types, variables, operators, expressions, control structures, packages, and visibility modifiers in Java programming.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Enhanced proficiency in Object-Oriented Analysis and Design, including identifying classes, attributes, methods, and associations.
- Improved Java programming proficiency, encompassing effective implementation of classes, methods, and interfaces, and handling exceptions.
- Enhanced problem-solving skills in object-oriented contexts, applying inheritance, polymorphism, and interface programming.
- Developed critical thinking in software development, analysing design decisions and proposing effective solutions.
- Improved collaboration skills in software development, demonstrated through participation in group projects and discussions.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in object-oriented programming languages, specifically Java, mastering concepts such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction.
- Competence in using Integrated Development Environments (IDEs) such as Eclipse or IntelliJ IDEA to write, compile, and debug object-oriented programs efficiently.
- Skills in using version control systems like Git, enabling collaborative development, code management, and tracking changes in object-oriented projects.
- Ability in debugging and troubleshooting object-oriented programs, identifying and fixing errors to ensure software functionality and reliability.
- Understanding the importance of documenting code and writing clear, concise comments to facilitate code comprehension, maintenance, and collaboration in object-oriented development projects.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures providing theoretical foundations covering topics such as object-oriented analysis and design, Java programming fundamentals, and software development concepts.
Supervised Placement and Practice Hours: Practical exercises and hands-on programming sessions allowing students to apply knowledge, practice coding, and solve real-world problems using object-oriented methodologies under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of programming exercises, practice with Java IDEs, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, presentations or mini projects, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction, practical application, and collaborative learning:

- Lectures: Introduction to theoretical foundations of object-oriented programming, including object-oriented analysis and design principles, Java language fundamentals, constructors, inheritance, polymorphism, encapsulation, exception handling, interfaces, and unit testing.
- Practical Exercises: Guided programming activities allowing students to apply theoretical knowledge, practice coding techniques, and implement object-oriented concepts in Java.
- Hands-On Programming Sessions: Intensive coding sessions providing opportunities to develop Java applications, solve programming challenges, and experiment with object-oriented design patterns.
- Collaborative Projects: Group-based activities fostering teamwork, enabling students to work together on software development projects applying object-oriented methodologies.
- Group Discussions and Code Reviews: Interactive sessions promoting critical thinking, problem-solving, peer learning, and code quality improvement through collaborative review.
- Digital Learning Tools: Use of IDEs (Eclipse, IntelliJ IDEA), version control systems (Git), online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development, collaborative learning, and application of object-oriented programming principles to create effective software solutions.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Presentation/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific object-oriented programming topics, such as analysis of object-oriented design principles, evaluation of Java programming approaches, examination of exception handling strategies, or assessment of software development methodologies. This assesses their ability to articulate technical concepts, critically evaluate programming practices, and develop evidence-based arguments.

Presentation/Mini Project (30%):
Learners will deliver an individual presentation (10 minutes, approximately 850 words or maximum 10 slides) or develop a mini programming project demonstrating practical application of object-oriented programming principles in

Java, including class design, inheritance, polymorphism, exception handling, and interface implementation. This assesses their capability to apply technical knowledge, create functional applications, communicate technical concepts effectively, and meet specified requirements.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of object-oriented programming with Java, including object-oriented principles, Java language fundamentals, constructors, inheritance, polymorphism, encapsulation, exception handling, interfaces, and unit testing. The examination will include a mix of multiple-choice, short-answer, code analysis, and programming questions.

Digital learning tools such as online submission platforms, code repositories (e.g., GitHub), and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, delivery of a revised presentation or resubmission of the mini project, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to IDEs and development tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Heller, S. & Bock, K. (2021). *Java in Action: A Project-Based Approach to Learning Java*. Manning Publications.
- Sierra, K. & Bates, B. (2021). *Head First Java: A Brain-Friendly Guide*. O'Reilly Media.
- Miller, K. (2022). *Object-Oriented Programming in Java: A Comprehensive Guide for Beginners and Professionals*. Springer.
- Bloch, J. & Gafter, N. (2020). *Java Concurrency in Practice*. Prentice Hall.
- Morrison, B. (2022). *Beginning Java Programming: The Object-Oriented Approach*. Packt Publishing.

Additional Reading

- Mott, D. (2023). *Java Programming: A Comprehensive Introduction*. Pearson.
- Schildt, H. (2022). *Java: The Complete Reference*. McGraw-Hill.
- Koffman, E. & Wolfgang, P. (2021). *Data Structures: Abstraction and Design Using Java*. Wiley.
- McGuire, S. (2020). *Object-Oriented Design & Programming in Java: A Hands-on Approach*. Cambridge University Press.

UNIT SPECIFICATIONS

Unit Title
Management Information Systems

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module develops learners' knowledge and skills in managing information systems for organisations, examining how systems can be used to support core business functions, drive business improvement, and enable organisations to be more productive and competitive within the global marketplace. Learners explore the strategic use of IT to support knowledge management, data management, and customer service functions, and examine the impact of IT systems on business operations and organisational structures. The module covers essential topics including the layers of information systems, corporate database management systems, data management principles, transaction

processing systems, customer relationship management, business intelligence, value creation strategies, competitive advantage concepts, decision-making frameworks, and system development tools and techniques. Through theoretical instruction, practical laboratory sessions, hands-on assignments, case studies, and interactive discussions, students develop the knowledge and competencies required to analyse, design, implement, and manage information systems that align with and achieve organisational objectives.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Manage the strategic use of IT to support knowledge management, data management, and customer service functions within organisational settings.
- Analyse the impact of IT systems on business functions and organisational structures to align with and achieve organisational objectives.
- Evaluate different types of IT systems and their roles in enhancing business operations, including transaction processing, customer relationship management, and business intelligence.
- Apply quality assurance and control measures to ensure the accuracy and reliability of data for informed decision-making and value creation.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The various layers of information systems, including services, integration, security, and analytics.
- Corporate database management systems, data management principles, and the characteristics of data within organisational contexts.
- Value creation strategies, competitive advantage concepts, and decision-making frameworks such as cost and benefit analysis.
- System development tools, techniques, and strategic models like Porter's Model of Generic Strategies and Wiseman's Strategic Planning Process.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to analyse and assess business problems and opportunities through the lens of IT systems and strategies.
- Proficiency in leveraging IT systems to enhance operational efficiency, customer service, and organisational effectiveness.
- Competence in designing and implementing differentiated products or services, optimising cost structures, and targeting specific market segments.
- Ability to use IT tools and technologies for data management, analysis, and decision support, contributing to organisational growth and competitiveness.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Proficiency in analysing and interpreting complex IT-related issues in knowledge management, data management, and customer service management.
- Enhanced ability to communicate technical concepts effectively to diverse audiences within organisational settings.
- Improved capability to collaborate with team members and stakeholders to address IT challenges and achieve business objectives.
- Advanced skills in information gathering, evaluation, and synthesis for informed decision-making in IT project management.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in using various IT systems for transaction processing, customer relationship management, business intelligence, and knowledge management to meet business objectives.
- Competence in corporate database management systems and data management techniques to ensure data quality and effective data manipulation within organisations.
- Understanding of value creation strategies, cost-benefit analysis, and make-or-buy decisions in the context of IT systems, contributing to improved operational efficiency and organisational effectiveness.
- Skills in system development tools and techniques, along with knowledge of strategic planning processes, to enhance IT position and provide value-added services in various business areas.

Module Workload

Component	Hours

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including theoretical lectures covering foundational concepts such as database management systems (DBMS), data models, normalisation techniques, IT system layers, strategic frameworks, and business intelligence.
Supervised Placement and Practice Hours: Practical laboratory sessions providing opportunities for students to apply knowledge in designing and implementing database structures using tools like MySQL and Oracle, writing SQL queries, optimising database performance, and implementing security measures under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of practical assignments, case study analysis, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, case study analysis, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction, practical application, and collaborative learning:

- Theoretical Lectures: Introduction to foundational concepts of management information systems, including IT system layers, database management systems, data management principles, transaction processing, customer relationship management, business intelligence, knowledge management, value creation strategies, competitive advantage frameworks, and strategic planning models.
- Practical Laboratory Sessions: Hands-on activities enabling students to apply theoretical knowledge in designing and implementing database structures, writing SQL queries, optimising database performance, implementing security measures, and using IT systems for business operations.
- Hands-On Assignments: Practical exercises challenging students to solve real-world business problems using IT systems, analyse organisational scenarios, and develop IT-based solutions.
- Interactive Discussions: Facilitated sessions promoting collaborative learning, critical thinking, and exploration of contemporary IT management challenges and opportunities.
- Case Studies: Analysis of real-world organisational scenarios illustrating the strategic use of IT systems, data management practices, and business transformation through technology.
- Digital Learning Tools: Use of database management tools (MySQL, Oracle), online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development, strategic thinking, and application of information systems management principles to enhance organisational effectiveness.
Assessment Strategy

Assessment Weightings

- Written Assignments: 40%
- Case Study: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific management information systems topics, such as analysis of IT strategy alignment with business objectives, evaluation of database management approaches, examination of business intelligence systems, or assessment of IT-enabled competitive advantage strategies. This assesses their ability to articulate technical and strategic concepts, critically evaluate IT management practices, and develop evidence-based recommendations.

Case Study (30%):
Learners will complete a case study analysis (750–1,000 words) examining real-world organisational scenarios involving management information systems, data management challenges, IT system implementation, or strategic use of technology for business improvement. This assesses their ability to apply theoretical frameworks, analyse complex situations, and propose effective IT-based solutions.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of management information systems, including IT system layers, database management, data management principles, transaction processing systems, customer relationship management, business intelligence, strategic frameworks, value creation strategies, and system development methodologies. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, resubmission of the case study analysis, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to database management tools and laboratory facilities, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Chaffey, D. (2020). *Digital Business and E-Commerce Management*. Pearson.
- Laudon, K.C. & Laudon, J.P. (2021). *Management Information Systems: Managing the Digital Firm* (17th ed.). Pearson.
- Rainer, R.K. & Turban, E. (2021). *Introduction to Information Systems: Supporting and Transforming Business* (4th ed.). Wiley.
- Bidgoli, H. (2020). *Electronic Commerce: Principles and Practice*. Cengage Learning.
- Lankhorst, M. (2021). *Enterprise Architecture at Work: Modelling, Communication and Analysis* (4th ed.). Springer.

Additional Reading

- Alavi, M. & Leidner, D.E. (2022). Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly*, 36(2), 107–136.
- Khosrow-Pour, M. (2021). *Advanced Methodologies and Technologies in Digital Marketing and Entrepreneurship*. IGI Global.

UNIT SPECIFICATIONS

Unit Title
Network Information Systems

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module develops learners' knowledge and skills in planning, configuring, setting up, and managing various types of networks including Local Area Networks (LAN), Personal Area Networks (PAN), Metropolitan Area Networks (MAN), and Wide Area Networks (WAN). Learners explore network architectures, protocols, standards, topologies, and

security measures essential for designing and managing effective network infrastructures. The module covers essential topics including network protocols (IPv4, IPv6, FTP, HTTP, SMTP, POP3), network topologies (logical and physical), OSI and TCP/IP models, network services configuration, network management practices, traffic management, network monitoring and troubleshooting techniques, and security implementation. Through theoretical instruction, practical laboratory sessions, hands-on assignments, and virtual labs, students develop the knowledge and technical competencies required to design, implement, monitor, troubleshoot, and secure network systems that meet organisational requirements and adapt to evolving technologies.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Design various network architectures, including peer-to-peer, client-server, cloud, and virtualised systems, tailored to organisational requirements.
- Configure network services and devices in alignment with business specifications, ensuring seamless connectivity and functionality.
- Implement effective network management practices, including throttling and traffic management, to optimise performance and resource allocation.
- Monitor and troubleshoot network issues using diagnostic tools and techniques to maintain high levels of network availability and reliability.
- Evaluate and adapt network designs to accommodate evolving technologies and business needs, ensuring scalability and efficiency.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Network protocols, including IPv4, IPv6, FTP, HTTP, SMTP, and POP3, and their functions within routed and managed networks.
- Various network topologies, such as logical (e.g., Ethernet, Token Ring) and physical (e.g., star, ring, mesh), and their implications for effective network design.
- The OSI and TCP/IP models, highlighting their roles and significance in facilitating network communication.
- The differences and similarities between various network protocols and their applications in different networking scenarios.
- How different topologies influence network performance, scalability, and reliability in various organisational contexts.

Skills
Upon successful completion of this module, learners will have developed:

- Proficiency in installing, testing, and troubleshooting networks according to specifications and design plans.
- Capability to utilise network analysis tools, such as SolarWinds and Netmon, effectively for route analysis and network tomography.
- Competence in configuring networking devices, including routers, switches, firewalls, and access points, to ensure optimal network performance.
- Ability to conduct regular network performance assessments to identify and resolve issues proactively.
- Skills in implementing security measures on network devices to protect against unauthorised access and ensure data integrity.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Systematic problem-solving and critical thinking in analysing complex network issues.
- Effective communication and collaboration with team members in network design and implementation projects.
- Proficient utilisation of network analysis tools and software for troubleshooting and optimisation.
- Adaptive learning and continuous improvement in response to evolving network technologies.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Applying advanced network management software for monitoring and optimising network performance.
- Expertise in network security tools and techniques to safeguard against cyber threats and breaches.
- Competence in configuring and managing cloud-based networking solutions for scalable and flexible infrastructure deployment.
- Proficiency in implementing network automation and orchestration techniques to streamline operations and enhance efficiency.

Module Workload

Component	Hours

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including live lectures, recorded sessions, and discussion forums facilitating real-time interaction and collaborative learning experiences covering network protocols, topologies, architectures, security, and management practices.
Supervised Placement and Practice Hours: Practical laboratory sessions and virtual labs providing opportunities for students to apply theoretical knowledge in configuring networks, troubleshooting issues, and implementing security measures under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of practical assignments simulating real-world network scenarios, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, presentations, and the final examination.

Pedagogy

- This module is delivered through a blended online learning approach combining multimedia resources, interactive content, and practical application:
- Live Lectures: Real-time instruction covering theoretical concepts of network information systems, including network protocols, topologies, OSI and TCP/IP models, network architectures, security principles, and management practices.
- Recorded Sessions: On-demand video content enabling flexible learning and review of key concepts at students' own pace.
- Discussion Forums: Online collaborative spaces facilitating peer interaction, knowledge sharing, and exploration of network-related topics.
- Interactive Modules: Engaging digital content reinforcing understanding of network concepts through interactive exercises and simulations.
- Virtual Labs: Simulated network environments enabling students to configure devices, implement protocols, troubleshoot issues, and practice network management techniques in safe, controlled settings.
- Practical Assignments: Real-world scenario-based exercises requiring application of theoretical knowledge to design, configure, and troubleshoot networks.
- Online Assessments and Quizzes: Formative assessments providing ongoing feedback to track understanding and progress.
- Collaborative Tools and Virtual Communities: Digital platforms fostering supportive learning environments and enhancing educational experiences through peer collaboration.
- The pedagogical approach emphasises practical skill development, problem-solving, and application of network information systems principles to real-world scenarios accessible from anywhere with internet connectivity.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Presentations: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific network information systems topics, such as analysis of network architectures, evaluation of network protocols, examination of network security measures, or assessment of network management strategies. This assesses their ability to articulate technical concepts, critically evaluate networking practices, and develop evidence-based recommendations.

Presentations (30%):
Learners will deliver an individual presentation (10 minutes, approximately 850 words or maximum 10 slides) communicating insights on network design, protocol implementation, troubleshooting methodologies, or network security practices. This assesses their capability to communicate technical concepts effectively, present analytical findings, and engage professional audiences using digital presentation tools.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of network information systems, including network protocols, topologies, OSI and TCP/IP models, network architectures, configuration of network devices, network management practices, monitoring and troubleshooting techniques, and security implementation. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy

- Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, delivery of a revised presentation, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to virtual labs and network simulation tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Kurose, J.F. & Ross, K.W. (2021). *Computer Networking: A Top-Down Approach*. Pearson.
- Tanenbaum, A.S. & Wetherall, D.J. (2020). *Computer Networks*. Pearson.
- Wright, J. (2020). *Networking For Dummies*. Wiley.
- Forouzan, B.A. (2021). *Data Communications and Networking*. McGraw-Hill.

UNIT SPECIFICATIONS

Unit Title
Cyber Security

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module provides learners with comprehensive understanding of cyber security fundamentals, including its historical development, legal and regulatory frameworks, risk management methodologies, and its impact on individuals and organisations. Learners explore cyber security concepts, principles, and theories including confidentiality, integrity, availability, defence-in-depth, and threat landscape analysis. The module covers essential topics including cyber security protection methods for networks and end-user devices, incident management and response strategies, compliance requirements (GDPR, HIPAA, PCI DSS, ISO/IEC 27001), and techniques for identifying, assessing, prioritising, and mitigating cyber security risks. Through theoretical instruction, practical exercises, case studies, virtual labs, and group discussions, students develop the knowledge and technical competencies required to analyse cyber security threats, implement protective measures, manage security incidents, and maintain organisational cyber security posture in contemporary digital environments.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Define the term 'cyber security' comprehensively, covering its scope, objectives, and significance in modern society.
- Explain in detail the processes of identifying, assessing, prioritising, and managing cyber security risks within various organisational contexts.
- Describe and interpret the laws, regulations, and compliance requirements related to cyber security, including GDPR and the Information Security Act.
- Summarise the historical development of cyber security, tracing its evolution from early computer networks to contemporary digital ecosystems.
- Analyse and articulate the multifaceted impacts of cyber security on individuals, organisations, economies, and national security.
- Develop effective strategies and methodologies to stay updated with the latest trends, threats, and best practices in cyber security through continuous learning and community engagement.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Cyber security concepts, principles, and theories, including confidentiality, integrity, availability, risk management, and defence-in-depth.
- Risk management techniques, encompassing identification, assessment, mitigation, and monitoring of cyber security risks across various organisational domains.
- Laws, regulations, and compliance frameworks relevant to cyber security, such as GDPR, HIPAA, PCI DSS, and ISO/IEC 27001.
- The historical evolution of cyber security, including key milestones, influential events, and shifts in the threat landscape and defensive strategies.
- The diverse impacts of cyber security breaches on individuals, organisations, governments, and society, including financial, reputational, legal, and socio-political ramifications.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to analyse cyber security scenarios to identify potential threats and formulate effective risk mitigation strategies.
- Proficiency in applying relevant legal and regulatory requirements to ensure compliance in cyber security operations.
- Competence in conducting historical analyses of cyber security trends and incidents to understand the evolving threat landscape.
- Ability to assess the impact of cyber security breaches on individuals and organisations to devise proactive risk mitigation measures.
- Skills in leveraging digital tools and resources to access and disseminate up-to-date cyber security information for decision-making.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Defining and analysing cyber security concepts, principles, and methodologies within diverse organisational contexts.
- Applying legal, regulatory, and ethical frameworks to evaluate cyber security risks, threats, and compliance requirements effectively.
- Evaluating historical developments and impacts of cyber security on individuals, organisations, and society to inform strategic decision-making and risk management practices.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Utilising advanced network security protection methods such as firewalls, intrusion detection systems (IDS), and virtual private networks (VPN) to safeguard against cyber threats and unauthorised access.
- Implementing robust end-user device protection measures including antivirus software, encryption protocols, and secure configuration settings to mitigate cyber security risks and vulnerabilities.
- Managing cyber security incidents effectively within an organisation, including incident detection, containment, eradication, and recovery, to minimise impact and restore normal operations.
- Evaluating and reviewing cyber security incident responses, analysing root causes, identifying lessons learned, and implementing corrective actions for continuous improvement in cyber security posture and resilience.

Module Workload

Component	Hours
Total Contact Hours	**30**

Component	Hours
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering cyber security fundamentals, interactive sessions, and group discussions exploring theoretical concepts and real-world applications.
Supervised Placement and Practice Hours: Practical exercises, virtual labs, and hands-on sessions providing opportunities to apply theoretical knowledge using cyber security tools and techniques, including network security configurations, incident response simulations, and vulnerability assessments under supervision.
Self-Study Hours: Independent learning activities including review of materials, exploration of online resources, participation in online cyber security communities, completion of practical assignments and case studies, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, case study analysis, and the final examination.

Pedagogy
This module is delivered through an interactive, multi-faceted approach combining theoretical instruction, practical application, and collaborative learning:

- Lectures: Introduction to cyber security fundamentals, including concepts, principles, theories, risk management methodologies, legal and regulatory frameworks, historical development, and impact analysis.
- Practical Exercises: Hands-on activities enabling learners to apply theoretical concepts to configure security systems, implement protective measures, and practice incident response procedures.
- Case Studies: Analysis of real-world cyber security incidents, breaches, and organisational responses illustrating practical application of security principles and lessons learned.
- Group Discussions: Interactive sessions encouraging active engagement with content, exploration of contemporary cyber security challenges, and collaborative problem-solving.
- Guest Lectures: Industry professionals and practitioners sharing insights into current trends, best practices, and real-world cyber security challenges and solutions.
- Digital Learning Tools: Use of virtual labs for hands-on practice with cyber security tools, simulation software for incident response training, and online forums for peer collaboration and knowledge sharing.
- Continuous Learning Resources: Encouragement to explore online resources, participate in cyber security communities, attend webinars, and stay updated with latest developments in the field.

The pedagogical approach emphasises practical skill development, critical thinking, and application of cyber security principles to protect digital assets and manage security incidents effectively.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Case Study: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific cyber security topics, such as analysis of risk management approaches, evaluation of legal and regulatory compliance requirements, examination of cyber security protection methods, or assessment of incident response strategies. This assesses their ability to articulate technical and strategic concepts, critically evaluate cyber security practices, and develop evidence-based recommendations.

Case Study (30%):
Learners will complete a case study analysis (750–1,000 words) examining real-world cyber security incidents, breaches, or organisational security challenges. This assesses their ability to apply theoretical frameworks, analyse

complex security scenarios, identify root causes, evaluate responses, and propose effective solutions and improvements.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of cyber security, including fundamental concepts, risk management techniques, legal and regulatory frameworks, historical development, protection methods, incident management, and impact analysis. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, simulation software, and virtual labs will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, resubmission of the case study analysis, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to virtual labs and cyber security tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Stallings, W. & Brown, L. (2022). *Computer Security: Principles and Practice*. Pearson.
- Mason, J. (2021). *Cybersecurity for Beginners: A Comprehensive Guide to Cybersecurity*. Independently published.
- Andress, J. & Winterfeld, S. (2021). *Cyber Warfare: Techniques, Tactics, and Tools for Security Practitioners*. Syngress.
- O'Connor, D. (2021). *Cybersecurity for Dummies*. Wiley.

Additional Reading

- Kizza, J.M. (2020). *Guide to Computer Network Security*. Springer.
- Shostack, A. (2021). *Threat Modelling: Designing for Security*. Wiley.
- Raghavan, S. & Pal, A. (2022). *Cyber Security: A Practical Guide for IT Professionals*. CRC Press.

UNIT SPECIFICATIONS

Unit Title
Python Primer: An Introduction to Programming with Python

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module provides students with foundational understanding of the Python programming language, encompassing basic concepts including variables, data types, statements, control structures, functions, error handling, and logging. Emphasising Python's relevance in data science-related programming roles, the module explores essential topics including recursion, comprehensions, iterators, parameters, scope, exception management, and file handling. Students are introduced to popular data science library packages such as Pandas, NumPy, and Matplotlib, enhancing their proficiency in Python for data analysis and related fields. Through lectures, hands-on coding sessions, interactive discussions, assignments, and projects, students develop the knowledge and technical competencies required to configure Python environments, write efficient code, solve programming challenges, and apply Python packages effectively for data analysis, web development, automation, and other applications.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Demonstrate ability to set up Python environments and utilise them effectively for programming tasks.
- Show proficiency in working with various types of data and variables, utilising appropriate Python constructs.
- Exhibit ability to write Python code using various programming constructs and effectively handle errors and exceptions.
- Apply Python packages and modules for data analysis and related tasks, enhancing productivity and efficiency.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The reasons why Python is widely used across various fields, including data science.
- How to obtain and configure Python, along with familiarity with tools like Python, Jupyter, and the Python interpreter.
- Different data types such as values, quantities, coordinates, arrays, and documents.
- Key concepts like control structures, recursion, comprehensions, iterators, functions, scope, and parameters.
- Common programming errors, managing exceptions, and handling files in Python.
- Python packages and modules for data science and other applications.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to obtain and configure Python on various platforms using Python, Jupyter, and the Python interpreter.
- Proficiency in working with different types of data and variables effectively in Python programming.
- Competence in writing Python code using various programming constructs like control structures, recursion, comprehensions, iterators, functions, and parameters.
- Ability to identify and fix common mistakes, manage exceptions, and handle records effectively in Python programs.
- Skills in applying packages and modules in Python programming, managing namespaces, and utilising commonly used data science packages.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Demonstrating advanced skills in writing efficient and concise Python code for various applications, including data analysis, web development, and automation.
- Applying enhanced problem-solving abilities by using algorithmic thinking to tackle complex programming challenges and optimise code efficiency.
- Exhibiting advanced critical thinking in identifying and resolving errors, bugs, and logical inconsistencies in Python code through systematic debugging techniques.
- Adapting to learning new Python libraries, frameworks, and technologies to stay updated with the evolving landscape of programming.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Expertise in using Python libraries such as Pandas, NumPy, and Matplotlib for data manipulation, analysis, and visualisation.
- Ability to create dynamic web applications using Python frameworks like Django or Flask, covering aspects such as user authentication and database integration.
- Skills in deploying Python applications to cloud platforms, understanding cloud services, resource management, and scalability.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**

Component	Hours
Total Learning Hours	150
ECTS Credits	6

Contact Hours: Structured learning activities including lectures covering essential Python concepts such as variables, data types, control structures, functions, modules, error handling, and data science libraries.
Supervised Placement and Practice Hours: Hands-on coding sessions allowing students to apply theoretical knowledge to solve programming challenges directly, practice with Python environments, and receive guidance on code development under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of programming exercises, exploration of online resources, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, programming projects, and the final examination.

Pedagogy
This module employs a comprehensive teaching approach combining theoretical instruction, practical application, and interactive learning:

- Lectures: Introduction to essential Python programming concepts including variables, data types, control structures, functions, recursion, comprehensions, iterators, scope, parameters, error handling, exception management, file handling, packages, and modules.
- Hands-On Coding Sessions: Practical programming activities enabling students to apply theoretical knowledge directly to solve programming challenges, develop applications, and experiment with Python features.
- Interactive Discussions: Facilitated sessions encouraging student engagement, clarification of concepts, exploration of programming challenges, and collaborative problem-solving.
- Assignments: Regular programming exercises enabling students to practice Python skills, reinforce learning, and demonstrate understanding of concepts.
- Projects: Real-world problem-solving activities enabling students to showcase their proficiency in Python by developing applications, analysing data, or automating tasks.
- Online Resources: Access to supplementary materials, tutorials, documentation, and learning platforms complementing instruction and encouraging deeper exploration of Python programming concepts.
- Digital Learning Tools: Use of Python development environments, Jupyter notebooks, online submission platforms, code repositories (e.g., GitHub), and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach aims to furnish students with robust understanding of Python programming through active participation, practical exercises, and progressive skill development.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Programming/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific Python programming topics, such as analysis of programming concepts, evaluation of Python libraries, examination of error handling approaches, or assessment of code optimisation strategies. This assesses their ability to articulate technical concepts, critically evaluate programming practices, and develop evidence-based arguments.

Programming/Mini Project (30%):
Learners will develop a programming project demonstrating practical application of Python skills, including problem analysis, code implementation, use of appropriate data structures and algorithms, error handling, and application of Python packages. This assesses their capability to apply technical knowledge, create functional applications, and meet specified requirements.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of Python programming, including fundamental concepts, data types, control structures, functions, recursion, comprehensions, iterators, error handling, file handling, and use of Python packages and modules. The examination will include a mix of multiple-choice, short-answer, code analysis, and programming questions.

Digital learning tools such as online submission platforms, code repositories (e.g., GitHub), and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, revision and resubmission of the programming project, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to Python development environments and learning resources, and online support, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Matthes, E. (2023). *Python Crash Course* (3rd ed.). No Starch Press.
- Ramalho, L. (2022). *Fluent Python* (2nd ed.). O'Reilly Media.
- McKinney, W. (2022). *Python for Data Analysis* (3rd ed.). O'Reilly Media.
- Müller, A.C. & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Inc.
- Ramalho, L. (2015). *Fluent Python: Clear, Concise, and Effective Programming*. O'Reilly.

Additional Reading

- Slatkin, B. (2020). *Effective Python: 90 Specific Ways to Write Better Python* (2nd ed.). Addison-Wesley.
- Unpingco, J. (2021). *Python Programming for Data Science and Machine Learning*. Springer.

UNIT SPECIFICATIONS

Unit Title
Mathematics for Computing

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module equips students with essential mathematical foundations for analysing computational problems in computer science and related fields. It covers discrete mathematics, graph theory, set theory, logic, number theory, and linear algebra—fundamental areas underpinning algorithm design, data structures, and computational systems. Additionally, the module introduces mathematical concepts relevant to data collection, data science, and data analysis, including data cleansing, statistical analysis, and probability, with focus on applications in computer science and information technology. Through lectures, interactive tutorials, virtual labs, online discussions, and case studies, students develop the knowledge and competencies required to apply mathematical principles to solve computational problems, perform statistical and probabilistic analyses, implement mathematical modelling techniques, and ensure accuracy and reliability in data-driven decision-making within computer science and IT contexts.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Apply mathematical concepts to analyse and solve computational problems in computer science.
- Ensure accuracy and reliability in data analysis through statistical and probabilistic methods.
- Manage mathematical modelling techniques for effective problem-solving in data science and IT applications.
- Collaborate with teams to integrate mathematical principles into computational and data-driven projects.

Knowledge

Upon successful completion of this module, learners will have engaged with:

- Fundamental concepts of discrete mathematics, set theory, logic, and graph theory in computational contexts.
- The principles of number theory and linear algebra relevant to computer science applications.
- Statistical analysis, probability, and data cleansing techniques for data science.
- Mathematical methods used in algorithm design, problem-solving, and data-driven decision-making.

Skills

Upon successful completion of this module, learners will have developed:

- Capability to apply discrete mathematics, set theory, and logic to solve computational problems.
- Proficiency in utilising statistical and probabilistic methods for data analysis and decision-making.
- Competence in implementing mathematical modelling techniques in computer science and data science applications.
- Ability to analyse datasets using data cleansing and statistical techniques to ensure accuracy and reliability.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Formulating computational problems using mathematical concepts and techniques, developing mathematical representations for algorithms, data structures, and systems.
- Utilising mathematical reasoning and algorithmic methods to solve computational problems, including sorting, searching, and graph algorithms.
- Performing matrix and vector operations such as addition, multiplication, inversion, and solving linear equations, while applying these concepts to computer science and information technology.

Module-Specific Digital Skills and Competences

Learners will develop digital skills and competencies including:

- Constructing mathematical representations of algorithms, data structures, and computational systems.
- Applying mathematical reasoning to solve computational problems such as sorting, searching, and graph algorithms.
- Performing matrix and vector operations, including addition, multiplication, inversion, and solving linear equations.
- Utilising statistical and probabilistic methods for data cleansing, interpretation, and decision-making in IT applications.

Module Workload

Component	Hours
Total Contact Hours	30
Supervised Placement and Practice Hours	20
Self-Study Hours	95
Assessment Hours	5
Total Learning Hours	150
ECTS Credits	6

Contact Hours: Structured learning activities including live and recorded lectures providing theoretical foundations in discrete mathematics, graph theory, set theory, logic, number theory, linear algebra, statistics, and probability.
Supervised Placement and Practice Hours: Interactive tutorials for problem-solving and virtual labs for implementing mathematical models in computational contexts under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of mathematical exercises, self-paced study, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, programming/mini projects, and the final examination.

Pedagogy

This module is delivered through an online blended learning approach combining theoretical instruction, interactive problem-solving, and practical application:

- Live and Recorded Lectures: Introduction to theoretical foundations of mathematics for computing, including discrete mathematics, set theory, logic, graph theory, number theory, linear algebra, statistical analysis, probability, and data cleansing techniques.
- Interactive Tutorials: Structured problem-solving sessions enabling students to apply mathematical concepts to computational problems, work through examples, and receive guidance.
- Virtual Labs: Hands-on activities for implementing mathematical models, performing matrix operations, applying statistical methods, and using computational tools for mathematical analysis.
- Online Discussions: Collaborative forums promoting critical thinking, exploration of mathematical applications in computing, and peer learning.
- Case Studies: Analysis of real-world computational scenarios illustrating application of mathematical principles to algorithm design, data analysis, and problem-solving.
- Self-Paced Study: Independent learning resources enabling students to review concepts, practice problems, and progress at their own pace.
- Digital Learning Tools: Use of mathematical software, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises development of mathematical reasoning, problem-solving skills, and application of mathematical principles to computer science and data science contexts.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Programming/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific mathematical topics relevant to computing, such as analysis of discrete mathematics concepts, application of graph theory to computational problems, examination of linear algebra techniques in data science, or assessment of statistical methods for data analysis. This assesses their ability to articulate mathematical concepts, apply theoretical knowledge, and develop evidence-based solutions.

Programming/Mini Project (30%):
Learners will develop a programming project demonstrating practical application of mathematical concepts in computational contexts, such as implementing algorithms using discrete mathematics, performing data analysis using statistical methods, or solving problems using linear algebra techniques. This assesses their capability to integrate mathematical principles with programming skills and create functional solutions.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of mathematics for computing, including discrete mathematics, set theory, logic, graph theory, number theory, linear algebra, statistical analysis, probability, and data cleansing techniques. The examination will include a mix of multiple-choice, short-answer, problem-solving, and computational questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, revision and resubmission of the programming project, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to mathematical software and learning resources, and online support, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Pomde, N.P. (2024). *Discrete Mathematics for Computer Science*. [Publisher not specified].
- Fessler, J.A. & Nadakuditi, R.R. (2024). *Linear Algebra for Data Science, Machine Learning, and Signal Processing*. [Publisher not specified].
- Liben-Nowell, D. (2024). *Discrete Mathematics for Computer Science*. [Publisher not specified].

- Lewis, H. & Zax, R. (2024). *Essential Discrete Mathematics for Computer Science*. [Publisher not specified].

Additional Reading

- Prakash, A.A. (2024). *Discrete Mathematics for Computer Science*. [Publisher not specified].
- Davies, S. (2024). *Discrete Mathematics for Computer Science Made Easy*. [Publisher not specified].
- Kumar, V. (2024). *Discrete Mathematics for Computer Science*. [Publisher not specified].

UNIT SPECIFICATIONS

Unit Title

Unlocking Big Data: Technologies and Strategies

Level

5

Learning Time Hours

150

Credit Value

6

Module Overview

This module provides students with comprehensive understanding of big data analytics methods and techniques, enabling them to effectively analyse large-scale datasets and derive meaningful insights for decision-making processes. Through theoretical learning and practical applications, students explore the evolution of big data, best practices for big data analytics including data validation and feature extraction, big data methods and tools such as YARN, HDFS, and MapReduce, advanced data analytic techniques including classification and regression algorithms, and data stream analysis including architecture and modelling principles. Using tools like Hadoop, Spark, and R, students develop the knowledge and technical competencies required to implement big data analytics solutions, perform data processing and analysis, develop sophisticated analytical models, and communicate the value of big data insights to stakeholders in professional contexts.

Learning Outcomes

Competences

Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Demonstrate proficiency in employing best practices for big data analytics, including data validation and promoting its value through real-world examples.
- Show competence in utilising big data methods and tools such as YARN, HDFS, and MapReduce, demonstrating understanding of their roles in high-performance architecture.
- Exhibit skill in implementing advanced data analytic techniques, including classification and regression, and assessing their applicability to real-world problems.
- Demonstrate capability to apply concepts related to data stream analysis, including understanding the architecture and models for streaming data.

Knowledge

Upon successful completion of this module, learners will have engaged with:

- The evolution of big data and its significance in contemporary data analytics practices.
- Best practices for big data analytics, including data validation, feature extraction, and value promotion.
- Big data methods and tools such as YARN, HDFS, and MapReduce, and their roles in distributed computing.
- Advanced data analytic techniques, including classification and regression algorithms, and their theoretical foundations.

Skills

Upon successful completion of this module, learners will have developed:

- Proficiency in applying big data analytics techniques using tools and platforms such as Hadoop, Spark, and R.
- Capability to perform data validation and advocate for the significance of big data insights to stakeholders.
- Competence in crafting and deploying sophisticated analytical models tailored for classification and regression tasks.
- Ability to analyse and process data streams efficiently, showcasing comprehension of stream architecture and modelling principles.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Utilising advanced big data analytics techniques using tools like Hadoop, Spark, and R to solve complex real-world problems.
- Conducting comprehensive data validation and effectively communicating the value of big data insights to stakeholders.
- Developing and deploying sophisticated analytical models for tasks such as classification and regression, demonstrating proficiency in machine learning techniques.
- Analysing and processing data streams efficiently, showcasing deep understanding of stream architecture and modelling concepts.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in utilising advanced big data analytics tools such as Hadoop, Spark, and R for efficient data processing and analysis.
- Ability to leverage data visualisation techniques to present complex insights derived from big data analytics effectively.
- Skills in implementing machine learning algorithms and statistical models to derive actionable insights from big data sets.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering theoretical concepts of big data analytics, methods, tools, and techniques.
Supervised Placement and Practice Hours: Practical demonstrations and hands-on exercises illustrating application of big data analytics using tools like Hadoop, Spark, and R under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of practical exercises, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction and practical application:

- Lectures: Introduction to theoretical concepts of big data analytics, including evolution of big data, best practices, data validation, feature extraction, big data methods and tools (YARN, HDFS, MapReduce), advanced analytic techniques (classification, regression), and data stream analysis.
- Practical Demonstrations: Instructor-led demonstrations illustrating application of big data analytics techniques using Hadoop, Spark, R, and other tools.
- Hands-On Exercises: Practical activities enabling students to apply knowledge using big data tools and platforms, ensuring comprehensive understanding of subject matter through direct experience.
- Digital Learning Tools: Use of big data platforms, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development and application of big data analytics techniques to real-world scenarios.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%

- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific big data analytics topics, such as analysis of big data methods, evaluation of analytical techniques, examination of data stream architectures, or assessment of best practices in big data analytics. This assesses their ability to articulate technical concepts, critically evaluate big data approaches, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key big data concepts, tools, techniques, and analytical methods. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of big data analytics, including evolution of big data, best practices, big data tools and methods, advanced analytic techniques, and data stream analysis. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to big data tools and platforms, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts
- Marz, N. & Warren, J. (2020). *Big Data: Principles and Best Practices of Scalable Real-Time Data Systems*. [Publisher not specified].
- Provost, F. & Fawcett, T. (2022). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. [Publisher not specified].
- Karau, H., Konwinski, A., Wendell, P. & Zaharia, M. (2021). *Learning Spark: Lightning-Fast Data Analytics*. [Publisher not specified].
- White, T. (2021). *Hadoop: The Definitive Guide* (4th ed.). [Publisher not specified].

Additional Reading
- Mahesh, A.S.S.P.T.E.B.K. (2023). *Fundamentals of Big Data Analytics*. [Publisher not specified]. This textbook covers essential concepts, methodologies, and tools for big data analytics, including practical examples and case studies.
- Walkowiak, S. (2022). *Big Data Analytics with R: A Practical Guide to Big Data Analysis*. [Publisher not specified]. A practical guide focusing on using R for big data analytics, covering techniques and tools for data analysis and visualisation.

UNIT SPECIFICATIONS

Unit Title
Introduction to Cryptography
Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module offers both theoretical understanding and practical application of cryptography and network security techniques essential for protecting digital information and communications. Learners explore fundamental and advanced cryptographic concepts including conventional encryption, asymmetric and symmetric cryptology, digital signatures, certificates, key exchange, key management, authentication, network access control, cloud computing security, and electronic mail security. Special focus is given to emerging technologies such as Bitcoin and blockchain, including block verification and cryptocurrency security requirements. Through theoretical lectures, practical demonstrations, hands-on exercises using Python programming, and interactive discussions, students develop the knowledge and technical competencies required to apply cryptographic techniques, implement security measures, analyse security threats, and understand the principles and applications of blockchain technology in contemporary digital environments.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Deal with security goals and various types of security attacks in digital environments by applying appropriate preventive and responsive measures.
- Recognise applied cryptography in daily life and the rationale behind their usage.
- Demonstrate proficiency in setting up a mobsp server for projects and implementing hash functions in Python programs.
- Apply digital signatures, their security properties, and the ability to generate digital signatures using Python.
- Deal with blockchain technology, including block verification, Bitcoin, and the security requirements of cryptocurrencies.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Security goals and various types of security attacks in the context of information system protection.
- Practical applications of cryptography and their significance in securing digital communications and data.
- Mobsp server setup, hash functions, digital signatures, and their implementation in Python.
- Blockchain technology, including block verification, Bitcoin, and cryptocurrency security requirements.

Skills
Upon successful completion of this module, learners will have developed:

- Proficiency in setting up mobsp servers and implementing hash functions and digital signatures in Python.
- Capability to generate and verify digital signatures using Python programming.
- Competence in analysing encryption modes and evaluating their effectiveness for different security contexts.
- Ability to apply cryptographic concepts in securing communication and data in practical scenarios.
- Skills in solving problems related to number theory and cryptographic algorithms through programming exercises.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Developing the ability to critically evaluate security goals, cryptographic techniques, and blockchain technologies, enabling them to make informed decisions in complex digital environments.
- Demonstrating proficiency in solving cryptographic problems, analysing security threats, and proposing effective solutions to mitigate risks.
- Improving their communication skills by articulating complex cryptographic concepts, security principles, and blockchain mechanisms to diverse audiences.
- Acquiring technical skills in setting up servers, implementing cryptographic algorithms, and programming Python scripts for hash functions, digital signatures, and blockchain verification.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in using cryptographic software tools for encryption, digital signatures, and blockchain operations.
- Expertise in Python programming, extending to implementing cryptographic algorithms and incorporating blockchain functionalities.
- Ability to navigate and utilise blockchain platforms for transaction validation and analysis.
- Competence in analysing cryptographic data using specialised software.
- Proficiency in implementing digital security protocols for data protection.

Module Workload

Component	Hours

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including theoretical lectures providing groundwork covering encryption algorithms, cryptographic protocols, security principles, blockchain technology, and cryptocurrency security.
Supervised Placement and Practice Hours: Practical demonstrations illustrating cryptographic concepts in action and hands-on exercises allowing students to apply cryptographic techniques, implement algorithms in Python, and work with blockchain technologies under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of programming exercises, exploration of cryptographic tools, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy
This module is delivered through a blend of theoretical instruction, practical application, and interactive learning:

- Theoretical Lectures: Introduction to cryptography fundamentals, including conventional encryption, symmetric and asymmetric cryptology, digital signatures, certificates, key exchange and management, authentication, network access control, cloud computing security, electronic mail security, and advanced crypto primitives.
- Practical Demonstrations: Instructor-led demonstrations illustrating cryptographic concepts, encryption techniques, digital signature generation and verification, and blockchain operations in action.
- Hands-On Exercises: Programming activities enabling students to implement cryptographic techniques using Python, including hash functions, digital signatures, encryption algorithms, and blockchain verification scripts.
- Interactive Discussions: Facilitated sessions promoting deeper exploration and critical analysis of cryptographic concepts, security principles, and their practical implications in contemporary digital environments.
- Emerging Technology Focus: Special emphasis on Bitcoin and blockchain technology, including block verification, cryptocurrency security, and practical applications.
- Digital Learning Tools: Use of cryptographic software, Python development environments, blockchain platforms, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development, critical thinking, and application of cryptographic principles to secure digital communications and data.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific cryptography topics, such as analysis of encryption algorithms, evaluation of digital signature schemes, examination of blockchain security mechanisms, or assessment of cryptographic applications in securing communications. This assesses their ability to articulate technical concepts, critically evaluate cryptographic techniques, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key cryptographic concepts, encryption methods, digital

signatures, blockchain technology, security principles, and practical applications. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of cryptography, including encryption algorithms, symmetric and asymmetric cryptology, digital signatures, key management, authentication, network security, blockchain technology, and cryptocurrency security. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to cryptographic tools and Python development environments, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Stallings, W. (2021). *Cryptography and Network Security: Principles and Practice*. [Publisher not specified].
- Katz, J. & Lindell, Y. (2020). *Introduction to Modern Cryptography: Principles and Protocols*. [Publisher not specified].
- Drescher, D. (2022). *Blockchain Basics: A Non-Technical Introduction in 25 Steps*. [Publisher not specified].

Additional Reading

- Antonopoulos, A.M. (2021). *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. [Publisher not specified].
- McKenzie, C.W. (2023). *Python Cryptography Toolkit*. [Publisher not specified]. A hands-on guide focusing on implementing cryptographic algorithms in Python, including practical examples and exercises.
- Mitnick, K.D. (2021). *The Art of Deception: Controlling the Human Element of Security*. [Publisher not specified].

UNIT SPECIFICATIONS

Unit Title
Exploring the Nexus: Data Science and Artificial Intelligence

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module provides students with comprehensive understanding of the intersection between Data Science and Artificial Intelligence, exploring how these complementary fields combine to solve complex problems across various domains. Learners explore foundational concepts of Data Science including the data science process, data collection strategies, data preprocessing techniques, data mining, and model development, alongside Artificial Intelligence techniques including production systems, searching algorithms, game playing strategies, and machine learning algorithms (supervised, unsupervised, and reinforcement learning). The module covers practical applications across various domains, security concerns in data science, regression and classification techniques, model evaluation metrics, cross-validation, and strategies to avoid overfitting and underfitting. Through theoretical insights, practical applications using tools like Scikit-Learn, Keras, and TensorFlow, workshops, and real-world collaborative projects, students

develop the knowledge, skills, and competencies necessary to leverage data-driven approaches and AI techniques effectively in professional contexts.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Apply foundational concepts of Data Science and Artificial Intelligence to carry out tasks and solve problems within practical or professional contexts.
- Demonstrate proficiency in identifying and navigating through different stages of a Data Science project.
- Show ability to recognise applications of Data Science across various domains and address associated security concerns.
- Exhibit skill in implementing Data Science processes including data collection, preprocessing, and model development.
- Demonstrate competence in applying AI techniques such as searching algorithms, game playing strategies, and machine learning algorithms.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The steps involved in the Data Science process and appropriate data collection strategies in relevant contexts.
- AI techniques, production systems, and search algorithms.
- Model development techniques, including regression, evaluation metrics, and model selection.
- Data mining and machine learning concepts such as supervised, unsupervised, and reinforcement learning.

Skills
Upon successful completion of this module, learners will have developed:

- Proficiency in data preprocessing techniques, including data cleaning, integration, and transformation.
- Capability to implement searching algorithms such as breadth-first search and depth-first search.
- Competence in developing and evaluating machine learning models using visualisation techniques.
- Ability to apply regression and classification techniques for prediction and decision-making.
- Skills in evaluating models using out-of-sample evaluation metrics, cross-validation, and strategies to avoid overfitting and underfitting issues.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Implementing advanced data preprocessing techniques for complex datasets.
- Developing and deploying complex machine learning models in real-world scenarios.
- Effectively communicating insights derived from data analysis to diverse audiences.
- Evaluating and optimising AI algorithms for performance and efficiency.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in utilising advanced data visualisation tools for conveying complex insights effectively.
- Capability in cloud computing platforms for scalable data storage and processing.
- Expertise in leveraging big data technologies for handling large-scale datasets efficiently.
- Skills in implementing automated data pipelines for streamlined data processing workflows.

Module Workload

Component	Hours
Total Contact Hours	30
Supervised Placement and Practice Hours	20
Self-Study Hours	95
Assessment Hours	5
Total Learning Hours	150
ECTS Credits	6

Contact Hours: Structured learning activities including lectures offering theoretical insights into the synergy between Data Science and Artificial Intelligence, covering data science processes, AI techniques, machine learning concepts, and practical applications.
Supervised Placement and Practice Hours: Workshops providing hands-on experience with relevant tools and techniques, and collaborative projects encouraging application of knowledge in real-world scenarios under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of programming exercises, exploration of data science and AI tools, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, programming projects, and the final examination.

Pedagogy
This module employs a multifaceted teaching approach integrating theoretical instruction, practical application, and collaborative learning:

- Lectures: Introduction to theoretical foundations of Data Science and Artificial Intelligence, including data science processes, data collection and preprocessing, AI techniques, production systems, searching algorithms, machine learning concepts (supervised, unsupervised, reinforcement learning), model development, evaluation metrics, and security considerations.
- Workshops: Hands-on sessions providing practical experience with data science and AI tools including Python libraries (Scikit-Learn, Keras, TensorFlow), data preprocessing techniques, implementation of searching algorithms, development of machine learning models, and data visualisation.
- Collaborative Projects: Real-world problem-solving activities encouraging students to apply knowledge in practical scenarios, fostering deeper understanding and skill development through teamwork and project-based learning.
- Practical Applications: Exploration of Data Science and AI applications across various domains including business, healthcare, finance, and technology.
- Digital Learning Tools: Use of Python development environments, machine learning frameworks, data visualisation tools, cloud computing platforms, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development, critical thinking, and application of data science and AI principles to solve complex problems in professional contexts.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Programming/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific Data Science and AI topics, such as analysis of data science processes, evaluation of machine learning algorithms, examination of AI techniques, or assessment of ethical and security considerations in data science applications. This assesses their ability to articulate technical concepts, critically evaluate approaches, and develop evidence-based recommendations.

Programming/Mini Project (30%):
Learners will develop a programming project demonstrating practical application of Data Science and AI techniques, including data collection and preprocessing, implementation of machine learning models, application of AI algorithms, model evaluation, and communication of insights. This assesses their capability to apply technical knowledge, create functional solutions using appropriate tools, and meet specified requirements.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of Data Science and Artificial Intelligence, including data science processes, data preprocessing techniques, AI techniques, searching algorithms, machine learning concepts, model development and evaluation, regression and classification techniques, and strategies to avoid overfitting and underfitting. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, revision and resubmission of the programming project, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to data science and AI tools and platforms, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Provost, F. & Fawcett, T. (2022). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. [Publisher not specified].
- Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. [Publisher not specified].
- Goodfellow, I., Bengio, Y. & Courville, A. (2021). *Deep Learning*. [Publisher not specified].
- Igual, L. & Seguí, S. (2021). *Introduction to Data Science: A Python Approach to Concepts, Techniques, and Applications*. [Publisher not specified].

Additional Reading

- Bishop, C.M. (2020). *Pattern Recognition and Machine Learning*. [Publisher not specified].
- Ng, A. (2022). *Machine Learning Yearning: Technical Strategy for AI Engineers, In the Era of Deep Learning*. [Publisher not specified].

UNIT SPECIFICATIONS

Unit Title
Exploring Data Protection and IT Security Measures

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
The 'Exploring Data Protection and IT Security Measures' module provides students with solid understanding of essential IT security concepts, familiarising them with foundational terminology, typical application domains, and various procedures and techniques widely used in the field. Learners explore security threats including network attacks, viruses, worms, and denial of service attacks, alongside security policies and models including institutional regulations, privacy policies, and access control models such as Role-Based Access Control (RBAC). The module covers cryptographic techniques including symmetric ciphers, public key systems, digital signatures, and hashing methods, security measures for hardware, operating systems, programs, applications, and databases, secure system design methodologies, and security throughout the software lifecycle. Through lectures, practical exercises, case studies, hands-on labs in simulated environments, and interactive discussions, students develop the knowledge and technical competencies required to identify and mitigate security vulnerabilities, implement protective measures, conduct security assessments, and ensure ongoing protection in diverse digital environments.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Identify and defend against network attacks by implementing secure layers and configuring SSL/TLS, VPNs, and firewalls.
- Address security issues in distributed systems, including web services and cloud computing, while mitigating cross-site scripting attacks.
- Follow secure system design methodologies to develop secure software and systems.
- Utilise design patterns and formal methods to enhance security in software development.
- Evaluate security measures throughout the software lifecycle to ensure ongoing protection.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Security threats, including the structures, growth, and vulnerabilities of Internet and Intranet systems, such as viruses, worms, and denial of service attacks.
- Security policies and models, including institutional regulations, privacy policies, and access control models like Role-Based Access Control (RBAC).
- Cryptographic techniques, including symmetric ciphers, public key systems, digital signatures, and hashing methods.
- The implications of security vulnerabilities and the importance of implementing protective measures.
- Best practices for maintaining security in digital environments, including risk assessment and incident response strategies.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to implement security measures in hardware and operating systems, focusing on process and memory protection, authentication, and securing Unix, Linux, and Windows systems.
- Proficiency in securing programs and applications by addressing language-specific vulnerabilities, such as buffer overflows, and implementing security measures in Java and .NET environments.
- Competence in applying database security principles, utilising authorisation systems to prevent SQL injection attacks and secure data in both relational and NoSQL databases.
- Ability to conduct security assessments and penetration testing to identify and mitigate vulnerabilities in software and systems.
- Skills in developing and enforcing security policies and procedures for managing access control and ensuring data protection.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Enhanced ability to analyse complex security issues and devise effective solutions.
- Advanced skills in identifying and mitigating security vulnerabilities in diverse systems.
- Improved capacity to convey security concepts to technical and non-technical audiences.
- Strengthened teamwork skills through group projects and discussions.
- Increased ability to adjust security measures in response to emerging threats and technologies.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in essential cybersecurity tools for threat detection and incident response.
- Ability to configure and manage network security components like firewalls and intrusion detection systems.
- Understanding and application of secure coding principles to develop resilient software.
- Skills in developing and executing incident response plans to address security breaches effectively.
- Knowledge of key cybersecurity regulations and standards to ensure organisational compliance.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures providing theoretical foundations covering cybersecurity threats, encryption techniques, security policies, access control models, secure system design, and security throughout the software lifecycle.
Supervised Placement and Practice Hours: Practical exercises and hands-on labs offering opportunities to apply knowledge in simulated environments, reinforcing concepts through practical application including network security configuration, penetration testing, and security assessment under supervision.

Self-Study Hours: Independent learning activities including review of materials, completion of practical exercises, exploration of cybersecurity tools, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy
This module employs a blended approach combining theoretical instruction, practical application, and experiential learning:

- Lectures: Introduction to theoretical foundations of IT security, including cybersecurity threats (network attacks, viruses, worms, denial of service attacks), security policies and models, cryptographic techniques, access control mechanisms, secure system design methodologies, and software lifecycle security.
- Practical Exercises: Hands-on activities enabling students to implement security measures, configure network security components (SSL/TLS, VPNs, firewalls), and apply cryptographic techniques.
- Hands-On Labs: Simulated environment activities providing opportunities to practice security configurations, conduct penetration testing, identify vulnerabilities, and implement protective measures in realistic scenarios.
- Case Studies: Analysis of real-world cybersecurity challenges and solutions illustrating security vulnerabilities, attack scenarios, and effective mitigation strategies, fostering critical thinking and problem-solving skills.
- Guest Lectures: Industry experts providing insights into current trends, best practices, and emerging challenges in cybersecurity.
- Interactive Discussions: Facilitated sessions encouraging exploration of security concepts, contemporary challenges, and collaborative problem-solving.
- Regular Assessments: Quizzes and assignments used to gauge understanding and track progress throughout the module.
- Digital Learning Tools: Use of cybersecurity tools, virtual labs, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development, critical thinking, and application of IT security principles to protect digital systems and data.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific IT security topics, such as analysis of security threats, evaluation of cryptographic techniques, examination of access control models, or assessment of secure system design methodologies. This assesses their ability to articulate technical concepts, critically evaluate security approaches, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key IT security concepts, security threats, cryptographic techniques, security policies, access control models, secure coding practices, and incident response strategies. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of data protection and IT security measures, including security threats, security policies and models, cryptographic techniques, network security, system and application security, database security, secure system design, and software lifecycle security. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.

- Support: Additional academic support, including individual tutorials, access to virtual labs and cybersecurity tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Stallings, W. & Brown, L. (2021). *Computer Security: Principles and Practice*. Pearson.
- Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley.
- Sullivan, B. & Liu, V. (2020). *Web Application Security: A Beginner's Guide*. McGraw-Hill Education.

Additional Reading

- Stallings, W. & Brown, L. (2021). *Computer Security: Principles and Practice*. Pearson.
- Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley.
- Sullivan, B. & Liu, V. (2020). *Web Application Security: A Beginner's Guide*. McGraw-Hill Education.
- Bertino, E. & Sandhu, R. (2021). *Database Security: Concepts, Approaches, and Applications*. Springer.
- Stuttard, D. & Pinto, M. (2021). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. Wiley.

UNIT SPECIFICATIONS

Unit Title
Understanding Information Security Standards

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module equips students with advanced knowledge and skills in cybersecurity, focusing on information security standards, threat analysis, risk management, cryptography, and network security. Through a blend of theoretical exploration and practical application, students delve into topics including different types of attacks and vulnerabilities, security goals and priorities, security services and mechanisms, conventional and modern cryptographic techniques (substitution ciphers, block ciphers, DES, AES, RSA), authentication methods, digital signatures, key management protocols, programme security issues (buffer overflow, virus threats), and network security threats and controls. By the end of the module, students will be proficient in implementing security measures, conducting risk assessments, configuring security technologies such as firewalls and intrusion detection systems, securing communication channels through encryption, and safeguarding digital assets against evolving cyber threats in alignment with information security standards and best practices.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Deal with different types of attacks and vulnerabilities in information security, ensuring appropriate mitigation strategies are considered.
- Demonstrate ability to identify and prioritise security goals for protecting information assets.
- Show competence in recognising and implementing various security services and mechanisms.
- Exhibit skill in analysing security risks and developing strategies to mitigate them effectively.
- Demonstrate capability to assess and enhance the overall security posture of an organisation or system.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Conventional cryptographic techniques, including substitution ciphers and block ciphers.
- Symmetric and asymmetric cryptographic algorithms such as DES, AES, and RSA.
- Authentication methods, digital signatures, and key management protocols.
- Common programme security issues like buffer overflow and virus threats.
- Common network security threats and the use of controls and technologies such as encryption and firewalls.

Skills

Upon successful completion of this module, learners will have developed:

- Capability to implement cryptographic techniques for authentication and data protection, ensuring confidentiality, integrity, and availability of information.
- Proficiency in identifying and mitigating non-malicious program errors and security vulnerabilities through comprehensive code reviews and testing methodologies.
- Competence in designing and implementing robust network security architectures and controls, focusing on defence in depth and layered security approaches.
- Ability to configure and manage security technologies such as firewalls and intrusion detection systems (IDS) to protect networks from unauthorised access and attacks.
- Skills in securing communication channels through encryption, implementing secure email protocols like PGP and S/MIME to safeguard sensitive information in transit.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Analysing and evaluating various types of cyber threats and vulnerabilities.
- Designing and implementing effective security measures to protect information assets.
- Conducting risk assessments and developing strategies for risk mitigation.
- Communicating security concepts and recommendations to stakeholders effectively.
- Applying best practices and standards in information security management and governance.

Module-Specific Digital Skills and Competences

Learners will develop digital skills and competencies including:

- Proficiency in utilising cybersecurity tools and technologies for threat detection and prevention.
- Competence in analysing security logs and data to identify potential security incidents.
- Ability to implement encryption techniques and secure communication protocols.
- Skills in configuring and managing network security devices such as firewalls and intrusion detection systems.
- Understanding of ethical hacking techniques and penetration testing methodologies for assessing system security.

Module Workload

Component	Hours
Total Contact Hours	30
Supervised Placement and Practice Hours	20
Self-Study Hours	95
Assessment Hours	5
Total Learning Hours	150
ECTS Credits	6

Contact Hours: Structured learning activities including lectures covering theoretical concepts and foundational principles of information security, including threat analysis, risk management, cryptographic techniques, authentication methods, programme security, and network security.
Supervised Placement and Practice Hours: Hands-on labs providing practical experience in implementing security measures, using cybersecurity tools, configuring security technologies, and conducting security assessments under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of practical exercises, exploration of cybersecurity tools and standards, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, case study analysis, and the final examination.

Pedagogy

This module is delivered through a combination of theoretical instruction, practical application, and experiential learning:

- Lectures: Introduction to theoretical concepts and foundational principles of information security, including information security standards, attacks and vulnerabilities, security goals and services, cryptographic techniques (conventional and modern), authentication methods, digital signatures, key management, programme security issues, and network security threats and controls.

- Hands-On Labs: Practical sessions providing experience in implementing security measures, using cybersecurity tools for threat detection and prevention, configuring firewalls and intrusion detection systems, implementing encryption protocols, and securing communication channels.
- Case Studies: Analysis of real-world security scenarios allowing students to apply knowledge to practical situations, examine security challenges, and develop effective solutions.
- Interactive Discussions: Facilitated sessions engaging students in critical thinking about security challenges, emerging threats, industry standards, and best practices in information security management.
- Digital Learning Tools: Use of cybersecurity tools, virtual labs, security testing platforms, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development, critical analysis, and application of information security standards to protect digital assets and enhance organisational security posture.

Assessment Strategy
Assessment Weightings
- Written Assignments: 40%
- Case Study: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific information security topics, such as analysis of cryptographic algorithms, evaluation of network security architectures, examination of risk management frameworks, or assessment of information security standards and compliance requirements. This assesses their ability to articulate technical concepts, critically evaluate security approaches, and develop evidence-based recommendations.

Case Study (30%):
Learners will complete a case study analysis (750–1,000 words) examining real-world information security scenarios, including security breaches, threat mitigation strategies, implementation of security controls, or organisational security posture assessments. This assesses their ability to apply theoretical frameworks, analyse complex security situations, identify vulnerabilities, and propose effective security solutions.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of information security standards, including attacks and vulnerabilities, security goals and services, cryptographic techniques, authentication methods, digital signatures, key management, programme security issues, network security threats and controls, and information security management best practices. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, resubmission of the case study analysis, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to hands-on labs and cybersecurity tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts
- Stallings, W. (2020). *Cryptography and Network Security: Principles and Practice*. Pearson.
- Schneier, B. (2020). *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley.
- Stallings, W. (2021). *Network Security Essentials: Applications and Standards*. Pearson.

Additional Reading
- Stuttard, D. & Pinto, M. (2021). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. Wiley.
- McMullen, D.H. (2021). *Secure Email: A Practical Guide to Using PGP and S/MIME*. O'Reilly Media.

UNIT SPECIFICATIONS

Unit Title
Exploring the Internet of Things (IoT)

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module provides students with comprehensive understanding of the technical and theoretical foundations of the Internet of Things (IoT) and its diverse applications across various domains. Students explore the structure of IoT systems, technology standards, communication protocols, network topologies, and the significant impact of IoT on the economy, society, consumers, industry, and the world of work. The module covers essential topics including the evolution of the Internet and IoT innovations, data protection principles and security compliance in IoT systems, communication standards and protocols, data exchange mechanisms, data storage and processing techniques (including linked data, RDF(S), NoSQL, MapReduce, and complex event processing), software engineering principles for IoT, and architectural patterns and platforms used in IoT system design and development. Through interactive lectures, hands-on laboratory sessions, case studies, and group discussions, students develop the knowledge and technical competencies required to design, implement, and deploy secure and scalable IoT solutions that address real-world challenges in contemporary networked environments.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Apply basic concepts of the Internet of Things (IoT) to real-world contexts, demonstrating awareness of its purpose and motivation.
- Evaluate the social and economic significance of IoT innovations for consumers and industry, and advise stakeholders on their practical implications.
- Apply data protection principles and ensure security compliance in the design and implementation of IoT systems.
- Select and implement appropriate communication standards, network topologies, and protocols relevant to IoT environments.
- Identify and analyse suitable data storage and processing techniques in networked environments to support IoT applications.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The evolution of the Internet and key innovations driven by IoT that impact consumers, industry, and the world of work.
- Network topologies, protocols, and technologies used in IoT communication systems.
- Data storage and processing techniques including linked data, RDF(S), and complex event processing.
- Software engineering principles and architectural patterns and platforms used in IoT system design and development.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to apply communication standards and protocols to design and implement Internet of Things (IoT) systems in real-world contexts.
- Proficiency in analysing and processing networked data using semantic reasoners and complex event processing techniques.
- Competence in operating and analysing large-scale data clusters using NoSQL and MapReduce within IoT environments.
- Ability to design and develop software for distributed and embedded systems specific to the IoT domain.
- Skills in applying architectural patterns and styles to develop robust and scalable IoT solutions across diverse industry domains.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Integrating IoT concepts and technologies into real-world applications across various domains.
- Analysing and evaluating the social, economic, and ethical implications of IoT innovations.
- Designing and implementing IoT systems with a focus on data protection and security measures.
- Collaborating effectively in multidisciplinary teams to develop and deploy IoT solutions.
- Demonstrating critical thinking and problem-solving skills in addressing complex challenges within the IoT ecosystem.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in utilising IoT technologies and platforms for data collection, analysis, and visualisation.
- Ability to implement communication protocols and standards to establish seamless connectivity in IoT systems.
- Skills in designing and deploying secure IoT architectures to protect data integrity and privacy.
- Competence in leveraging cloud computing and edge computing technologies for scalable and efficient IoT deployments.
- Proficiency in programming languages and tools relevant to IoT development.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including interactive lectures providing theoretical foundations and conceptual understanding of IoT fundamentals, applications, communication systems, data processing techniques, and system architectures.
Supervised Placement and Practice Hours: Hands-on laboratory sessions allowing students to gain practical experience in implementing IoT systems and solutions using various hardware and software tools under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of practical exercises, exploration of IoT technologies and platforms, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy
This module is delivered through a multifaceted approach combining theoretical instruction, practical application, and collaborative learning:

- Interactive Lectures: Introduction to theoretical foundations of the Internet of Things, including IoT fundamentals, evolution of the Internet, IoT innovations and their impact, network topologies and protocols, communication standards, data storage and processing techniques, software engineering principles, and architectural patterns for IoT systems.
- Hands-On Laboratory Sessions: Practical activities enabling students to implement IoT systems and solutions, work with IoT hardware and software tools, configure communication protocols, develop embedded systems applications, and deploy IoT architectures.
- Case Studies: Analysis of real-world IoT implementations examining practical applications across various industries, security and privacy challenges, data management approaches, and lessons learned from successful and unsuccessful deployments.
- Group Discussions: Collaborative sessions facilitating critical thinking and problem-solving by exploring IoT challenges, ethical considerations, social and economic implications, and emerging trends in the IoT ecosystem.
- Digital Learning Tools: Use of IoT development platforms, hardware kits, simulation tools, cloud computing services, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises acquisition of both theoretical knowledge and practical skills essential for success in the field of IoT.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific IoT topics, such as analysis of IoT architectures, evaluation of communication protocols, examination of data processing techniques, or assessment of security and privacy considerations in IoT systems. This assesses their ability to articulate technical concepts, critically evaluate IoT approaches, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key IoT concepts, communication standards, network topologies, data storage and processing techniques, software engineering principles, and security considerations. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of the Internet of Things, including IoT fundamentals, evolution and innovations, communication systems, network protocols, data processing techniques, software engineering principles, architectural patterns, security and privacy considerations, and social and economic implications. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to laboratory facilities and IoT development tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Ziegler, S., Radócz, R., Rodriguez, A.Q. & Matheu Garcia, S.N. (Eds.) (2024). *Springer Handbook of Internet of Things*. Springer.
- Chiara, P.G. (2024). *The Internet of Things and EU Law: Cybersecurity, Privacy and Data Protection Challenges*. [Publisher not specified].
- Savaglio, C., Fortino, G., Zhou, M. & Ma, J. (Eds.) (2024). *Internet of Things: Technologies, Communications and Computing*. [Publisher not specified].
- Soldatos, J. (2023). *Internet of Things: Technologies, Challenges, and Applications*. [Publisher not specified].
- Raj, P. & Raman, A.C. (2022). *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*. CRC Press.
- Cirani, S., Ferrari, G., Picone, M. & Veltri, L. *Internet of Things: Architectures, Protocols and Standards*. Wiley.
- Buyya, R. & Vahid Dastjerdi, A. (Eds.) (2016). *Internet of Things: Principles and Paradigms*. Morgan Kaufmann.
- Dian, F.J. & Vahidnia, R. (2020). *IoT Use Cases and Technologies*. British Columbia Institute of Technology.

Additional Reading

- Vermesan, O. & Friess, P. (2020). *Internet of Things – The Call of the Edge: Everything Intelligent Everywhere*. [Publisher not specified].
- Qureshi, K.N., Newe, T., Jeon, G. & Chehri, A. (Eds.) (2024). *Cybersecurity Vigilance and Security Engineering of Internet of Everything*. [Publisher not specified].
- Hu, F. (2020). *Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations*. [Publisher not specified].

- Sinclair, B. (2020). *IoT Inc: How Your Company Can Use the Internet of Things to Win in the Outcome Economy*. [Publisher not specified].
- Minteer, A. *Analytics for the Internet of Things (IoT): Intelligent Analytics for Your Intelligent Devices*. [Publisher not specified].
- Raj, P. & Raman, A.C. (2017). *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*. CRC Press.

UNIT SPECIFICATIONS

Unit Title
Machine Learning: Supervised Learning and Unsupervised Learning

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module equips students with thorough understanding of both supervised and unsupervised learning techniques, combining theoretical learning with practical applications to enable students to design, implement, and evaluate machine learning models applicable to a range of tasks. Students explore various learning paradigms, machine learning theory including version spaces, PAC learning, and VC dimension, perspectives and issues related to learning algorithms including biases and generalisation capabilities, and model evaluation techniques. The module covers supervised learning algorithms including decision trees, linear regression, logistic regression, neural networks, support vector machines, multilayer perceptron, and kernel methods for classification and regression tasks, alongside unsupervised learning techniques including clustering and dimensionality reduction for discovering patterns and structures in data. Through lectures, practical exercises using libraries like scikit-learn, TensorFlow, and PyTorch, and hands-on projects, students develop the knowledge and skills to implement algorithms, fine-tune model parameters, preprocess data, perform feature engineering, and interpret and visualise results for informed decision-making.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:
- Demonstrate proficiency in understanding various learning paradigms and their applications in supervised and unsupervised learning.
- Show ability to analyse perspectives and issues related to learning algorithms, including biases and generalisation capabilities.
- Exhibit competence in implementing supervised learning algorithms such as decision trees, neural networks, and support vector machines for classification and regression tasks.
- Demonstrate skill in utilising unsupervised learning techniques such as clustering and dimensionality reduction to discover patterns and structures in data.
- Show capability to evaluate and compare different learning algorithms based on their performance metrics and computational complexity.

Knowledge
Upon successful completion of this module, learners will have engaged with:
- Concepts such as version spaces, PAC learning, and VC dimension in the context of machine learning theory.
- Various supervised learning algorithms, including decision trees, linear regression, logistic regression, neural networks, and support vector machines.
- Advanced supervised learning techniques, such as multilayer perceptron and kernel methods, in the context of non-linear classification.
- Model evaluation techniques and algorithms for assessing the performance and generalisation ability of machine learning models.

Skills
Upon successful completion of this module, learners will have developed:
- Capability in implementing supervised learning algorithms and fine-tuning model parameters to achieve optimal performance on classification and regression tasks.

- Proficiency in applying unsupervised learning techniques to explore and extract meaningful patterns from large datasets.
- Competence in interpreting and visualising results obtained from machine learning models to gain insights into underlying data structures.
- Ability to practice preprocessing and feature engineering to enhance the performance of machine learning models.
- Skills in applying dimensionality reduction techniques effectively to reduce the complexity of high-dimensional data while preserving essential information.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Implementing and fine-tuning supervised learning algorithms for classification and regression tasks, considering both linear and non-linear models.
- Applying unsupervised learning techniques to identify patterns and structures within data, utilising clustering and dimensionality reduction methods.
- Evaluating and comparing different machine learning algorithms based on their performance metrics and computational complexity.
- Interpreting and visualising the results obtained from machine learning models to derive actionable insights and make informed decisions.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in utilising machine learning libraries like scikit-learn, TensorFlow, or PyTorch for implementing various algorithms.
- Mastery of data preprocessing techniques using pandas, NumPy, or similar libraries to clean and prepare datasets effectively.
- Skills in visualising and interpreting model performance metrics using tools such as Matplotlib or Seaborn.

Module Workload

Component	**Hours**
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering foundational concepts and theories of machine learning, supervised and unsupervised learning paradigms, algorithm theory, and model evaluation, supplemented with real-world examples illustrating practical applications.
Supervised Placement and Practice Hours: Practical exercises providing opportunities to implement machine learning algorithms and techniques using relevant tools and libraries, and hands-on projects allowing application of knowledge to solve real-world problems under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of programming exercises, exploration of machine learning tools and techniques, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, presentations, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction, practical application, and project-based learning:

- Lectures: Introduction to foundational concepts and theories of machine learning, including learning paradigms, machine learning theory (version spaces, PAC learning, VC dimension), supervised learning algorithms (decision trees, linear regression, logistic regression, neural networks, support vector machines, multilayer perceptron, kernel methods), unsupervised learning techniques (clustering, dimensionality reduction), model evaluation, and performance metrics.

- Practical Exercises: Hands-on activities providing students with opportunities to implement machine learning algorithms using libraries like scikit-learn, TensorFlow, and PyTorch, practice data preprocessing and feature engineering, and experiment with algorithm parameters and configurations.
- Hands-On Projects: Real-world problem-solving activities allowing students to apply knowledge to practical scenarios, design and implement complete machine learning solutions, and develop critical thinking and practical skills in machine learning.
- Real-World Examples: Integration of case studies and practical applications illustrating how machine learning techniques are used across various domains and industries.
- Digital Learning Tools: Use of machine learning frameworks, data processing libraries (pandas, NumPy), visualisation tools (Matplotlib, Seaborn), online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development, critical thinking, and application of machine learning principles to solve real-world problems.

Assessment Strategy
Assessment Weightings
- Written Assignments: 40%
- Presentations: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific machine learning topics, such as analysis of learning algorithms, evaluation of supervised versus unsupervised approaches, examination of model evaluation techniques, or assessment of computational complexity and performance trade-offs. This assesses their ability to articulate technical concepts, critically evaluate machine learning approaches, and develop evidence-based recommendations.

Presentations (30%):
Learners will deliver an individual presentation (10 minutes, approximately 850 words or maximum 10 slides) communicating insights on machine learning concepts, algorithm implementation, model performance analysis, or practical applications. This assesses their capability to communicate technical concepts effectively, present analytical findings, and engage professional audiences using digital presentation tools.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of machine learning, including learning paradigms, machine learning theory, supervised learning algorithms, unsupervised learning techniques, model evaluation, performance metrics, and computational complexity considerations. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, delivery of a revised presentation, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to machine learning tools and development environments, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
- Flach, P. (2021). *Machine Learning: The Art and Science of Algorithms that Make Sense of Data* (2nd ed.). Wiley.
- Murphy, K.P. (2022). *Probabilistic Machine Learning: An Introduction*. MIT Press.

Additional Reading

- Bashir, H. (2022). *Deep Learning for Vision Systems*. Springer.

- Joshi, P. (2021). *TensorFlow 2.0 in Action*. Manning Publications**.**

UNIT SPECIFICATIONS

Unit Title
Data Structure & Algorithm

Level
5

Learning Time Hours
150

Credit Value
6

Module Overview
This module covers key elements of digital system abstractions and fundamental concepts in data structures and algorithms essential for efficient software development and problem-solving. Students explore digital information representation, digital logic, Boolean algebra, and state elements, alongside mathematical foundations including asymptotic notation, summations, and recurrence relations for algorithm analysis. The module encompasses the design and implementation of fundamental data structures including trees, FIFO queues, stacks, priority queues, and heaps, searching algorithms including binary search trees, balanced search trees, and hash tables, sorting algorithms including insertion sort, merge sort, quicksort, and heap sort, and graph algorithms including depth-first search and connected components. Students learn to design basic components of digital systems using combinational and sequential circuits, employing tools such as Verilog. Through theoretical instruction, virtual laboratories, practical assignments, and collaborative problem-solving sessions, students develop the knowledge and skills to analyse computational complexity, design efficient data structures and algorithms, and implement solutions for real-world computing problems.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Analyse asymptotic notation to compare the efficiency of algorithms.
- Develop data structures such as trees, queues, stacks, and heaps.
- Design advanced searching algorithms, including binary search trees and hash tables.
- Integrate sorting algorithms like insertion sort and quicksort into software applications.
- Implement graph algorithms for depth-first search and connected components.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The mathematical foundations of algorithm analysis, including asymptotic notation, summations, and recurrence relations.
- The structure and operations of data structures like trees, FIFO queues, stacks, priority queues, and heaps.
- The principles and mechanisms of searching algorithms, identifying the advantages and disadvantages of different approaches.
- Various sorting algorithms, including their computational complexities and appropriate use cases in different computing contexts.
- Common graph algorithms and their applications in solving computing problems, such as searching and sorting within graph structures.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to analyse complex computational problems using mathematical tools and principles.
- Proficiency in designing efficient data structures and algorithms in programming languages such as Verilog or similar tools.
- Competence in selecting and applying appropriate searching techniques to optimise data retrieval and manipulation.
- Ability to implement sorting algorithms to ensure accurate and efficient data organisation in software projects.
- Skills in applying graph traversal techniques to practical problems, enhancing problem-solving skills in graph-related domains.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Applying asymptotic notation, summations, and recurrence relations to evaluate and enhance the performance of algorithms.
- Designing, implementing, and utilising fundamental data structures such as trees, FIFO queues, stacks, priority queues, and heaps in practical applications.
- Implementing and comparing searching algorithms like binary search trees, balanced search trees, and hash tables.
- Applying various sorting algorithms such as insertion sort, merge sort, quicksort, and heap sort to organise data efficiently.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Developing and refining algorithms using advanced techniques such as divide-and-conquer and dynamic programming.
- Translating theoretical concepts into efficient, practical implementations using programming languages like Verilog.
- Gaining expertise in using programming tools to design, implement, and test digital systems.
- Utilising graphical and simulation tools to visualise, analyse, and validate the behaviour of digital circuits and algorithms.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering foundational theoretical aspects such as digital system abstractions, mathematical concepts (asymptotic notation, summations, recurrence relations), data structures, algorithms, and digital logic principles.
Supervised Placement and Practice Hours: Virtual laboratories where students use tools like Verilog to implement and simulate digital systems, design data structures, and develop algorithms under supervision, alongside collaborative problem-solving sessions.
Self-Study Hours: Independent learning activities including review of materials, completion of programming assignments, practice with algorithm implementation, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, presentations, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction, practical application, and collaborative learning:

- Lectures: Introduction to foundational theoretical concepts including digital system abstractions, digital information representation, digital logic, Boolean algebra, state elements, mathematical foundations (asymptotic notation, summations, recurrence relations), data structures (trees, queues, stacks, heaps), searching algorithms, sorting algorithms, and graph algorithms.
- Virtual Laboratories: Hands-on sessions where students use tools like Verilog to implement and simulate digital systems, design combinational and sequential circuits, and develop data structures and algorithms in practical contexts.
- Practical Assignments: Programming exercises enabling students to implement data structures, develop searching and sorting algorithms, analyse computational complexity, and solve algorithmic problems.
- Collaborative Problem-Solving Sessions: Group activities designed for discussion, exploration of algorithmic approaches, and collaborative development of solutions to complex computational problems.
- Virtual Office Hours and Additional Resources: Comprehensive support available to students for clarification of concepts, guidance on assignments, and assistance with technical challenges.

- Digital Learning Tools: Use of programming environments, simulation tools, Verilog development platforms, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises development of both theoretical understanding and practical skills in data structures, algorithms, and digital system design.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Presentations: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific data structures and algorithms topics, such as analysis of algorithm complexity, evaluation of data structure efficiency, examination of searching or sorting techniques, or assessment of graph algorithm applications. This assesses their ability to articulate technical concepts, apply mathematical analysis, and develop evidence-based solutions.

Presentations (30%):
Learners will deliver an individual presentation (10 minutes, approximately 850 words or maximum 10 slides) communicating insights on data structures, algorithm design and analysis, computational complexity, or practical implementations using tools like Verilog. This assesses their capability to communicate technical concepts effectively, present analytical findings, and engage professional audiences using digital presentation tools.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of data structures and algorithms, including mathematical foundations, asymptotic analysis, data structure operations, searching algorithms, sorting algorithms, graph algorithms, and digital system design principles. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, delivery of a revised presentation, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to virtual laboratories and programming tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Sedgewick, R. & Wayne, K. (2021). *Algorithms* (4th ed.). Addison-Wesley.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C. (2022). *Introduction to Algorithms* (4th ed.). MIT Press.
- Weiss, M.A. (2020). *Data Structures and Algorithm Analysis in C++* (4th ed.). Pearson.

Additional Reading

- Goodrich, M.T., Tamassia, R. & Goldwasser, M.H. (2021). *Data Structures and Algorithms in Java* (6th ed.). Wiley.
- Aziz, A., Lee, T.-H. & Prakash, A. (2021). *Elements of Programming Interviews: The Insiders' Guide*. EPI.

UNIT SPECIFICATIONS

Unit Title
Exploring the Fundamentals of Web Security

Level

6

Learning Time Hours
150

Credit Value
6

Module Overview
This module equips students with solid foundation in web security fundamentals, enabling them to understand, identify, and mitigate security threats in web-based systems. Students explore web application architecture, components of the Internet and their vulnerabilities, HTTP and HTTPS protocols, web authentication technologies, and common attack trends including injection flaws, programming bugs, XSS, SQL injection, and memory corruption exploits. The module covers essential topics including email security protocols (PGP, S/MIME), wireless network security, mobile security management, disaster recovery protocols, ethical hacking techniques, cyber laws, security audit procedures, and cybersecurity solutions. Students learn to implement security measures including web application firewalls, browser security, e-commerce platform protection, XML security, Content Security Policy, serialisation security, DNS rebinding protection, and logging collection and analysis. Through theoretical lectures, practical demonstrations, hands-on exercises in simulated scenarios, interactive discussions, and case studies, students develop the skills to implement effective security measures and safeguard web applications against evolving cyber-attacks.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Analyse the components of the Internet to identify weak points and mitigate security risks.
- Differentiate between HTTP and HTTPS protocols and evaluate their significance for web security.
- Identify various web authentication technologies and their roles in securing web applications.
- Assess recent attack trends and types of web security threats to implement effective defence strategies.
- Distinguish the importance of web application security from network security principles.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Web application architecture and common attack trends, including injection flaws and programming bugs.
- Email security protocols such as PGP and S/MIME, and their role in securing online communications.
- Wireless network security components and strategies for securing wireless networks.
- Mobile security management principles, disaster recovery protocols, and ethical hacking techniques.
- Cyber laws, security audit procedures, and cybersecurity solutions for mitigating online threats.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to manage web application configurations and implement web application firewalls for infrastructure security.
- Proficiency in identifying and mitigating web hacking techniques, including XSS, SQL injection, and memory corruption exploits.
- Competence in securing web browsers and e-commerce platforms against security vulnerabilities and malicious code.
- Ability to conduct security testing to assess vulnerabilities in wireless networks.
- Skills in implementing XML security, Content Security Policy, and logging collection and analysis for web applications.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Proficiency in identifying and mitigating various web security threats, including injection flaws and malicious code.
- Advanced understanding of email security protocols such as PGP and S/MIME for securing online communications.
- Competence in securing wireless networks and implementing security measures to protect against cyber-attacks.
- Skills in conducting security testing and forensic analysis to investigate cybersecurity incidents effectively.
- Ability to apply ethical hacking techniques and penetration testing methodologies to identify vulnerabilities in web applications and networks.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Ability in conducting security testing and analysing results to identify vulnerabilities and weaknesses in web applications.
- Competence in utilising XML security, Content Security Policy, and logging collection for enhanced web application security.
- Ability to implement advanced security measures such as serialisation security and DNS rebinding to protect against evolving web threats.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including theoretical lectures covering key concepts such as web application architecture, common attack trends, web security protocols (HTTP/HTTPS), authentication technologies, email security, wireless network security, mobile security, and cybersecurity solutions.
Supervised Placement and Practice Hours: Practical demonstrations illustrating implementation of security measures like web application firewalls and encryption protocols, and hands-on exercises allowing students to apply concepts in simulated scenarios under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of security testing exercises, exploration of web security tools and techniques, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy

- This module is delivered through a combination of theoretical instruction, practical application, and experiential learning:
- Theoretical Lectures: Introduction to web security fundamentals, including web application architecture, Internet components and vulnerabilities, HTTP and HTTPS protocols, web authentication technologies, attack trends (injection flaws, XSS, SQL injection, memory corruption), email security protocols, wireless and mobile security, disaster recovery, ethical hacking, cyber laws, and security audit procedures.

Practical Demonstrations: Instructor-led sessions illustrating implementation of security measures including web application firewalls, encryption protocols, browser security configurations, e-commerce platform protection, and security testing tools.

Hands-On Exercises: Practical activities allowing students to apply security concepts in simulated scenarios, identify vulnerabilities, implement protective measures, conduct penetration testing, and analyse security incidents.

Interactive Discussions: Facilitated sessions deepening understanding and fostering critical thinking about web security challenges, emerging threats, and effective solutions.

Case Studies: Analysis of real-world web security incidents, breaches, and mitigation strategies illustrating practical application of security principles.

Digital Learning Tools: Use of web security testing tools, vulnerability scanners, penetration testing platforms, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development, critical analysis, and application of web security principles to protect web applications against cyber-attacks.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%

- Quizzes: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific web security topics, such as analysis of web application vulnerabilities, evaluation of security protocols, examination of authentication mechanisms, or assessment of defensive strategies against specific attack types. This assesses their ability to articulate technical concepts, critically evaluate security approaches, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key web security concepts, protocols, attack types, mitigation techniques, and security best practices. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of web security fundamentals, including web application architecture, HTTP/HTTPS protocols, authentication technologies, common attack trends, email security, wireless and mobile security, ethical hacking, security testing, and implementation of protective measures. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
- Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to web security testing tools and virtual labs, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts
- Hoffman, A. (2021). *Web Application Security: A Beginner's Guide*. McGraw-Hill Education.
- Stuttard, D. & Pinto, M. (2021). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. Wiley.
- Schlesinger, D.E.D. (2022). *Securing the Web: A Comprehensive Guide to Web Application Security*. Apress.

Additional Reading
- Schneier, B. (2021). *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley.
- Thomas, J.J.D. (2023). *Introduction to Cyber Security: The Comprehensive Guide for Beginners*. Cybersecurity Press.

UNIT SPECIFICATIONS

Unit Title
Mathematical Modelling

Level
6

Learning Time Hours
150

Credit Value
6

Module Overview

This module equips students with theoretical knowledge and practical skills required for mathematical modelling of dynamic phenomena across various domains. Students explore fundamental modelling approaches including decay and growth models in linear and nonlinear forms, compartmental models for population dynamics and financial systems, differential equations for planetary motions and satellite orbits, and constant-coefficient linear difference equations with applications in economics and finance. The module also covers computational modelling techniques including regular expressions, automata, and hidden Markov models in computational linguistics, alongside computational techniques for information extraction, question answering, summarisation, natural language processing, and machine translation. Through theoretical lectures, practical demonstrations, hands-on exercises using computational tools and software, and collaborative projects, students develop the knowledge and skills to develop mathematical models, analyse complex dynamic systems, predict system behaviour, and apply computational techniques for problem-solving across diverse fields.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Apply decay and growth models in linear and nonlinear forms to analyse dynamic phenomena across various fields.
- Develop mathematical models for compartments and population dynamics, applying them to real-world scenarios such as pandemics and financial dynamics.
- Utilise differential equations to model planetary motions, satellite orbits, and circular motion in diverse applications.
- Comprehend and apply basic models and theories of constant-coefficient linear difference equations in economic and financial contexts.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Decay and growth models, including linear and nonlinear formulations, and their applications in analysing dynamic systems.
- The significance of compartmental models in modelling population dynamics, financial dynamics, and other dynamic phenomena.
- Differential equations and their role in modelling planetary motions, satellite orbits, and circular motion.
- Basic models and theories related to constant-coefficient linear difference equations, with applications in economics and finance.

Skills
Upon successful completion of this module, learners will have developed:

- Proficiency in developing mathematical models for dynamic phenomena using decay and growth models, compartmental models, and differential equations.
- Capability to apply mathematical models to analyse and predict complex systems like population dynamics and planetary motions.
- Competence in utilising modelling techniques such as regular expressions, automata, and hidden Markov models in computational linguistics.
- Ability to employ computational techniques for tasks like information extraction, question answering, and summarisation in natural language processing and machine translation.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Developing and applying mathematical models to analyse dynamic phenomena in various fields.
- Utilising advanced modelling techniques to predict complex systems and dynamics accurately.
- Interpreting and applying computational techniques for tasks such as information extraction and question answering.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in utilising computational tools and software for mathematical modelling and analysis.
- Ability to process and analyse large datasets using digital tools and techniques.
- Skills in applying computational algorithms for solving complex problems in various domains.
- Competence in utilising digital platforms for collaborative research and project work.

Module Workload

Component	Hours
Total Contact Hours	30

Component	Hours
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering theoretical concepts in mathematical modelling, including decay and growth models, compartmental models, differential equations, difference equations, and computational modelling techniques.
Supervised Placement and Practice Hours: Practical sessions focusing on applying theoretical concepts using computational tools, and hands-on exercises reinforcing learning and ensuring comprehensive understanding of mathematical modelling under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of modelling exercises, exploration of computational tools and software, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, programming/mini projects, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction, practical application, and hands-on learning:

- Lectures: Introduction to theoretical concepts in mathematical modelling, including decay and growth models (linear and nonlinear), compartmental models for population and financial dynamics, differential equations for motion and orbits, constant-coefficient linear difference equations, and computational modelling techniques (regular expressions, automata, hidden Markov models).
- Practical Demonstrations: Instructor-led sessions illustrating application of mathematical models to real-world scenarios, including pandemics, financial dynamics, planetary motions, and computational linguistics applications.
- Hands-On Exercises: Programming and modelling activities enabling students to develop mathematical models using computational tools, analyse dynamic systems, implement algorithms, and solve complex problems.
- Collaborative Projects: Group-based activities fostering teamwork and enabling application of mathematical modelling techniques to real-world challenges across various domains.
- Digital Learning Tools: Use of mathematical modelling software, computational tools, data analysis platforms, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises comprehensive understanding of mathematical modelling through combination of theoretical learning and practical application using computational techniques.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Programming/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific mathematical modelling topics, such as analysis of decay and growth models, evaluation of compartmental modelling approaches, examination of differential equation applications, or assessment of difference equations in economic contexts. This assesses their ability to articulate theoretical concepts, apply mathematical reasoning, and develop evidence-based solutions.

Programming/Mini Project (30%):
Learners will develop a programming project or mini project demonstrating practical application of mathematical modelling techniques, including development of models for dynamic phenomena, implementation using computational tools, analysis of complex systems, and interpretation of results. This assesses their capability to apply theoretical knowledge, create functional models, and solve real-world problems.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of mathematical modelling, including decay and growth models, compartmental models, differential equations, difference equations, computational modelling techniques, and applications across various domains. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, revision and resubmission of the programming project, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to computational tools and modelling software, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Anthony, M. & Biggs, N. (2021). *Mathematical Models for Society and the Environment*. Springer.
- McGowan, L.P. (2021). *Mathematical Modeling: A Comprehensive Introduction*. Wiley.
- Huijben, H.H.G.M. (2022). *Differential Equations for Engineers and Scientists*. CRC Press.

Additional Reading

- de Jong, S.R. (2022). *Introduction to Mathematical Modelling*. Springer.
- Smith, R.D. (2023). *Applied Mathematical Modeling: A Multidisciplinary Approach*. Academic Press.
- Osterwalder, A. & Pigneur, Y. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. [Publisher not specified].

UNIT SPECIFICATIONS

Unit Title
Introduction to Quantum Computing

Level
6

Learning Time Hours
150

Credit Value
6

Module Overview
This module provides students with comprehensive understanding of quantum computation fundamentals and their practical applications, enabling engagement with quantum computing technologies and contribution to advancements in the field. Students explore key concepts including the circuit model, fundamental principles of quantum physics, manipulation of qubits, quantum gates (single-qubit and multiple-qubit), quantum operations, quantum algorithms (including superdense coding and Shor's algorithm), quantum computational complexity theory, and error correction techniques for achieving fault-tolerant quantum computation. The module covers classical and quantum error correction methods, analysis of quantum system states and behaviour, and evaluation of algorithm efficiency and performance. Through lectures covering theoretical concepts, practical demonstrations illustrating quantum computing applications, and hands-on exercises using software tools and simulation environments, students develop the knowledge and skills to design and implement quantum circuits, apply quantum algorithms, analyse computational complexity, and develop fault-tolerant quantum computation systems.

Learning Outcomes
Competences

Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Apply the fundamental principles of quantum computation in practical contexts, including the manipulation of qubits and quantum gates.
- Apply principles of quantum mechanics to analyse the states and behaviour of quantum systems.
- Design and implement quantum algorithms for various computational tasks.
- Assess computational complexity and error correction techniques in quantum computation.
- Evaluate and compare different quantum algorithms based on their efficiency and performance.

Knowledge

Upon successful completion of this module, learners will have engaged with:

- Key concepts in quantum computation, including the circuit model and fundamental principles of quantum physics.
- Quantum gates and operations, including single-qubit and multiple-qubit gates.
- Quantum algorithms such as superdense coding and Shor's algorithm.
- Quantum computational complexity theory and error correction techniques.
- Classical and quantum error correction methods, highlighting their significance in achieving fault-tolerant quantum computation.

Skills

Upon successful completion of this module, learners will have developed:

- Capability to design and implement quantum circuits by utilising both single and multiple qubit gates effectively.
- Proficiency in applying quantum algorithms to solve computational problems efficiently.
- Competence in analysing and evaluating the computational complexity of quantum algorithms.
- Ability to implement error correction techniques in quantum computation.
- Skills in developing fault-tolerant quantum computation systems.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Designing and implementing quantum circuits for various computational tasks, utilising single and multiple qubit gates.
- Analysing and evaluating the behaviour of quantum systems, including their states and quantum operations.
- Applying principles of quantum mechanics to solve complex computational problems efficiently.
- Assessing computational complexity and error correction techniques in quantum computation.

Module-Specific Digital Skills and Competences

Learners will develop digital skills and competencies including:

- Proficiency in utilising quantum computing software and simulation tools for designing and testing quantum circuits.
- Capability in data analysis techniques to interpret and visualise the behaviour of quantum systems and algorithms.
- Skills in programming languages commonly used in quantum computing, such as Qiskit or Quipper, for algorithm implementation.

Module Workload

Component	Hours
Total Contact Hours	30
Supervised Placement and Practice Hours	20
Self-Study Hours	95
Assessment Hours	5
Total Learning Hours	150
ECTS Credits	6

Contact Hours: Structured learning activities including lectures covering theoretical concepts of quantum computation, quantum physics principles, quantum circuits, quantum gates, quantum algorithms, computational complexity, and error correction techniques.
Supervised Placement and Practice Hours: Practical demonstrations illustrating application of quantum computing concepts, and hands-on exercises allowing students to implement and experiment with quantum circuits and algorithms using software tools and simulation environments under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of quantum computing exercises, exploration of quantum programming languages and tools, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, programming/mini projects, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction, practical application, and experiential learning:

- Lectures: Introduction to theoretical concepts of quantum computation, including circuit model, fundamental principles of quantum physics, qubits and quantum states, quantum gates (single-qubit and multiple-qubit operations), quantum algorithms (superdense coding, Shor's algorithm, and others), quantum computational complexity theory, and error correction techniques (classical and quantum methods).
- Practical Demonstrations: Instructor-led sessions illustrating practical application of quantum computing concepts, implementation of quantum circuits, execution of quantum algorithms, and use of quantum computing software and simulation tools.
- Hands-On Exercises: Programming and implementation activities enabling students to design quantum circuits, implement quantum algorithms using programming languages like Qiskit or Quipper, experiment with quantum operations, and analyse algorithm performance in simulation environments.
- Digital Learning Tools: Use of quantum computing software (Qiskit, Quipper), simulation environments, data analysis and visualisation tools, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises development of theoretical understanding and practical skills in quantum computing through combination of conceptual learning and hands-on implementation.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Programming/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific quantum computing topics, such as analysis of quantum algorithms, evaluation of computational complexity, examination of error correction techniques, or assessment of quantum circuit design approaches. This assesses their ability to articulate theoretical concepts, apply quantum mechanics principles, and develop evidence-based analysis.

Programming/Mini Project (30%):
Learners will develop a programming project or mini project demonstrating practical application of quantum computing concepts, including design and implementation of quantum circuits, development of quantum algorithms using programming languages like Qiskit or Quipper, analysis of algorithm performance, and implementation of error correction techniques. This assesses their capability to apply theoretical knowledge, create functional quantum computing solutions, and solve computational problems.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of quantum computing, including fundamental principles, quantum physics concepts, quantum gates and operations, quantum algorithms, computational complexity theory, error correction techniques, and fault-tolerant quantum computation. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, revision and resubmission of the programming project, and/or a resit examination, as appropriate.

- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to quantum computing software and simulation tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- Nielsen, M.A. & Chuang, I.L. (2021). *Quantum Computation and Quantum Information* (10th Anniversary Edition). Cambridge University Press.
- Preskill, J. (2021). Quantum Computing in the NISQ Era and Beyond. *Quantum*.
- Deutsch, D. (2021). *The Beginning of Infinity: Explanations That Transform the World*. Penguin Books.

Additional Reading

- Granade, C.G., et al. (2021). *Quantum Algorithms for Fixed Qubit Architectures*. Cambridge University Press.
- Lloyd, S. (2022). *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos*. Knopf.

UNIT SPECIFICATIONS

Unit Title
System Analysis and Designing

Level
6

Learning Time Hours
150

Credit Value
6

Module Overview
This module develops learners' awareness of system analysis and design in an organisational context, introducing various techniques used within systems analysis and design and the methodologies employed in the system development process. Students explore the software development life cycle including requirements gathering, design, implementation, testing, and maintenance, alongside key components of systems analysis including problem identification, feasibility analysis, and system design. The module covers different approaches to system analysis and design including structured analysis, object-oriented analysis, and agile methodologies, various types of information systems including business information systems, decision support systems, and enterprise resource planning systems, human-computer interface (HCI) design principles, system design methodologies including prototyping and rapid application development, and security and control considerations. Through lectures, workshops, and practical exercises conducted in a virtual learning environment using modelling software and collaboration platforms, students develop the knowledge and skills to produce system design documents, create system models using techniques such as data flow diagrams and UML, and collaborate with stakeholders to design effective information systems.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Explain how systems analysis and design contribute to each stage of the software development life cycle, including requirements gathering, design, implementation, testing, and maintenance.
- Identify and describe the key components of systems analysis, including problem identification, feasibility analysis, requirements gathering, and system design.
- Analyse different approaches to system analysis and design, such as structured analysis, object-oriented analysis, and agile methodologies.
- Evaluate system design requirements in terms of functionality, usability, scalability, and security.
- Discuss the importance of each stage in traditional software life cycle approaches and evaluate various software development lifecycle models.
- Differentiate between hard and soft system methodologies and discuss their application in system analysis and design.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Various types of information systems, including business information systems, decision support systems, and enterprise resource planning systems.
- The principles of human-computer interface (HCI) design and its impact on system usability and user experience.
- Concepts and methodologies related to system design, including prototyping, agile development, and rapid application development.
- Security and control considerations in system design, including authentication, authorisation, and encryption techniques.

Skills
Upon successful completion of this module, learners will have developed:
- Capability to produce comprehensive system design documents that outline system requirements, functionalities, and design specifications.
- Proficiency in creating models of software systems using various techniques such as data flow diagrams, entity-relationship diagrams, and UML diagrams.
- Competence in designing security and control mechanisms to protect system data and resources from unauthorised access and manipulation.
- Ability to implement testing strategies to ensure system reliability, performance, and compliance with specified requirements.
- Skills in collaborating with stakeholders to gather feedback and iteratively refine system designs to meet user needs and enhance functionality.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:
- Analysing system components and design requirements.
- Evaluating software development lifecycle models.
- Applying system design methodologies to produce system designs.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:
- Proficiency in HCI principles and methodologies.
- Knowledge of system design concepts and methodologies.
- Ability to use digital tools for system design and modelling.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including lectures covering theoretical concepts and principles of system analysis and design, supplemented by demonstrations and case studies illustrating practical applications.
Supervised Placement and Practice Hours: Workshops and practical exercises providing opportunities for hands-on application of system analysis and design techniques using digital tools such as modelling software and collaboration platforms under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of system design exercises, exploration of modelling tools and methodologies, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, quizzes, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction and practical application in a virtual learning environment:

Lectures: Introduction to theoretical concepts and principles of system analysis and design, including software development life cycle stages, systems analysis components (problem identification, feasibility analysis, requirements gathering, system design), approaches to system analysis and design (structured analysis, object-oriented analysis, agile methodologies), software development lifecycle models, hard and soft system methodologies, types of information systems, HCI design principles, and security considerations.

Workshops: Collaborative sessions providing structured opportunities to apply system analysis and design techniques, work through case studies, and develop system models using digital tools.

Practical Exercises: Hands-on activities conducted in the virtual learning environment enabling students to use modelling software (data flow diagrams, entity-relationship diagrams, UML diagrams), collaboration platforms, and other digital tools to design information systems and create system documentation.

Demonstrations and Case Studies: Real-world examples illustrating application of system analysis and design methodologies in organisational contexts, showcasing best practices and common challenges.

Digital Learning Tools: Use of modelling software, collaboration platforms, online submission platforms, simulation software, virtual labs, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development and application of system analysis and design techniques to solve real-world organisational challenges.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Quizzes: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific system analysis and design topics, such as analysis of software development lifecycle models, evaluation of system design approaches, examination of HCI principles, or assessment of security and control mechanisms. This assesses their ability to articulate theoretical concepts, critically evaluate system design methodologies, and develop evidence-based recommendations.

Quizzes (30%):
Regular quizzes will assess learners' understanding of key system analysis and design concepts, software development lifecycle stages, systems analysis components, design methodologies, HCI principles, and security considerations. These formative assessments provide ongoing feedback and support progressive learning.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of system analysis and design, including software development life cycle, systems analysis components, approaches to system analysis and design, software development lifecycle models, hard and soft system methodologies, types of information systems, HCI design principles, system design methodologies, and security and control considerations. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, simulation software, and virtual labs will be used for assessment.

Pass Mark: 50%

Reassessment Policy

- Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, completion of alternative quizzes, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to modelling software and collaboration platforms, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- McGregor, J.F. & McGregor, G.W. (2021). *Systems Analysis and Design: An Active Approach*. Cengage Learning.
- Hoffer, George & Valacich (2021). *Modern Systems Analysis and Design*. Pearson.
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.

Additional Reading

- Ambler, S.W. (2021). *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. Wiley.
- Rainer, R.K. & Prince, B. (2022). *Introduction to Information Systems*. Wiley.

UNIT SPECIFICATIONS

Unit Title

Overview of Blockchain and its Security

Level

6

Learning Time Hours

150

Credit Value

6

Module Overview

This module explores the fundamentals of blockchain technology, including its applications beyond cryptocurrencies, decentralised consensus mechanisms, smart contracts, and security considerations. Students examine distributed record-keeping systems, fault modelling in distributed systems including the Byzantine Generals problem, consensus algorithms and their scalability challenges, and various distributed ledger technologies. The module covers cryptographic primitives employed in blockchain, Bitcoin blockchain architecture, Ethereum smart contracts, permissioned blockchain platforms like Hyperledger Fabric, privacy-preserving technologies such as Zcash and zk-SNARKS, decentralised applications (dApps), vulnerabilities and attacks on blockchains including Sybil attacks and selfish mining, and regulatory and ethical considerations. Through theoretical lectures, practical exercises, hands-on projects, case studies, group activities, and guest lectures from industry speakers, students develop skills in analysing blockchain use cases, designing decentralised applications, implementing smart contracts, assessing security threats, and understanding the potential impact of blockchain adoption across various industries.

Learning Outcomes

Competences

Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Ensure the integrity of data by recognising the necessity and implications of distributed record-keeping systems.
- Model faults and adversaries within distributed systems, including the Byzantine Generals problem.
- Analyse consensus algorithms to identify scalability challenges and propose solutions for improved performance.
- Evaluate various distributed ledger technologies and their applications in ensuring data transparency and security.
- Implement strategies for fault tolerance and recovery in distributed systems to enhance reliability.

Knowledge

Upon successful completion of this module, learners will have engaged with:

- Cryptographic primitives and consensus mechanisms employed in blockchain technology.
- The Bitcoin blockchain, Ethereum smart contracts, and Hyperledger Fabric as permissioned blockchain platforms.
- Privacy-preserving technologies like Zcash and zk-SNARKS, along with vulnerabilities and attacks on blockchains.
- The role of decentralised applications (dApps) and their architecture in blockchain ecosystems.
- Regulatory and ethical considerations surrounding blockchain technology and its implementation in various industries.

Skills

Upon successful completion of this module, learners will have developed:

- Expertise in implementing and deploying smart contracts on blockchain platforms, considering verification challenges and Turing completeness.
- Proficiency in assessing and mitigating security threats in blockchain networks, including Sybil attacks and selfish mining.
- Capability to evaluate different consensus mechanisms and select appropriate solutions based on application requirements.
- Competence in implementing security protocols to safeguard blockchain applications against common vulnerabilities.
- Skills in developing effective testing strategies for smart contracts to ensure functionality and security before deployment.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Proficiency in modelling faults and adversaries within distributed systems, including the Byzantine Generals problem.
- Advanced understanding of consensus algorithms and their scalability challenges, with the capability to propose solutions for improved scalability.
- Competence in assessing and implementing blockchain applications, including smart contracts and permissioned blockchain platforms like Hyperledger Fabric.
- Skills in evaluating security threats in blockchain networks and implementing measures to mitigate vulnerabilities such as Sybil attacks and selfish mining.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiency in cryptographic techniques such as hash functions and digital signatures.
- Skills in designing and deploying smart contracts on blockchain platforms.
- Competence in assessing and mitigating security threats in blockchain networks.

Module Workload

Component	Hours
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including theoretical lectures covering blockchain fundamentals, cryptographic primitives, consensus mechanisms, distributed ledger technologies, smart contracts, security considerations, and regulatory frameworks, alongside discussions and guest lectures from industry speakers providing insights into current trends and challenges.
Supervised Placement and Practice Hours: Practical exercises for implementing blockchain solutions, group activities for modelling faults and analysing consensus algorithms, and hands-on projects working on real-world case studies to understand practical applications of blockchain technology under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of blockchain development exercises, exploration of blockchain platforms and tools, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, presentations/mini projects, and the final examination.

Pedagogy
This module is delivered through a combination of theoretical instruction, practical application, and collaborative learning:

- Theoretical Lectures: Introduction to blockchain fundamentals, including distributed record-keeping systems, fault modelling (Byzantine Generals problem), consensus algorithms, cryptographic primitives, Bitcoin blockchain, Ethereum smart contracts, permissioned blockchain platforms (Hyperledger Fabric), privacy-preserving technologies, dApps architecture, security vulnerabilities, and regulatory considerations.

- Practical Exercises: Hands-on activities enabling students to implement smart contracts, deploy blockchain applications, assess security threats, and apply cryptographic techniques.
- Hands-On Projects: Real-world project-based learning allowing students to design decentralised applications, develop blockchain solutions, and analyse practical use cases.
- Group Activities: Collaborative sessions for modelling faults in distributed systems, analysing consensus algorithms, and solving blockchain-related challenges.
- Case Studies: Analysis of real-world blockchain implementations across various industries, examining applications, challenges, and outcomes.
- Guest Lectures and Industry Speakers: Insights from practitioners providing perspectives on current trends, challenges, and opportunities in blockchain technology.
- Collaborative Learning Environments: Facilitated discussions and peer learning promoting comprehensive understanding of blockchain concepts and practical applications.
- Digital Learning Tools: Use of blockchain development platforms, smart contract frameworks, cryptographic tools, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises development of both theoretical understanding and practical skills for blockchain application development and security analysis.

Assessment Strategy
Assessment Weightings
- Written Assignments: 40%
- Presentation/Mini Project: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific blockchain topics, such as analysis of consensus mechanisms, evaluation of distributed ledger technologies, examination of smart contract security, or assessment of blockchain applications in specific industries. This assesses their ability to articulate technical concepts, critically evaluate blockchain technologies, and develop evidence-based recommendations.

Presentation/Mini Project (30%):
Learners will deliver an individual presentation (10 minutes, approximately 850 words or maximum 10 slides) or develop a mini project demonstrating practical application of blockchain technology, including design and implementation of decentralised applications, smart contract development, security analysis, or evaluation of blockchain use cases. This assesses their capability to apply theoretical knowledge, create functional blockchain solutions, communicate technical concepts effectively, and meet specified requirements.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of blockchain technology and security, including distributed record-keeping systems, fault modelling, consensus algorithms, cryptographic primitives, blockchain platforms (Bitcoin, Ethereum, Hyperledger Fabric), smart contracts, privacy-preserving technologies, security threats, and regulatory considerations. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, delivery of a revised presentation or resubmission of the mini project, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to blockchain development platforms and tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts
- Vasilakos, C.M.K. (2021). *Blockchain Technology: Principles and Applications*. Wiley.
- Ravi, S. (2021). *Mastering Blockchain: Unlocking the Power of Bitcoin, Ethereum, Blockchain Technology, and Smart Contracts*. Packt Publishing.

- Drescher, D. (2021). *Blockchain Basics: A Non-Technical Introduction in 25 Steps*. Apress.
- Tapscott, A. & Tapscott, D. (2021). *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. Penguin.
- Papp, M.C.M.K.A. (2022). *Smart Contracts: The Blockchain Technology That Will Replace Lawyers*. Independently Published.

Additional Reading

- Bashir, I. (2021). *Mastering Blockchain: Unlocking the Power of Cryptography and Distributed Ledger Technology*. [Publisher not specified].
- Drescher, D. (2020). *Blockchain Basics: A Non-Technical Introduction in 25 Steps* (2nd ed.). [Publisher not specified].
- Prusty, N. (2021). *Building Blockchain Projects*. [Publisher not specified].
- Wysopal, C. (2021). *Smart Contracts: The Essential Guide to Using Blockchain Smart Contracts for Your Business*. [Publisher not specified].
- Norman, A.T. (2021). *Blockchain Technology Explained: A Practitioner's Guide to Using Blockchain Technology*. [Publisher not specified].

UNIT SPECIFICATIONS

Unit Title
IT Project Management

Level
6

Learning Time Hours
150

Credit Value
6

Module Overview

This module develops learners' skills in managing Information Technology projects to implement systems or change in their organisations, with particular relevance for middle and senior managers whose responsibilities include the introduction of operational or strategic change. Students explore fundamental networking concepts including the OSI model structure and functions, TCP/IP operations, client-server architecture and socket interfaces, and the Linux operating system within open-source computing contexts. The module covers network protocol configuration, socket programming using the C language, Linux installation and system administration, DNS servers and clients configuration, and network security devices and measures. Through theoretical lectures, practical laboratory sessions, hands-on assignments, interactive discussions, and case studies, students develop the knowledge and skills to configure network protocols, implement client-server architecture, utilise Linux commands and tools for system management and troubleshooting, configure DNS infrastructure, design secure network architectures, and manage IT projects effectively in organisational settings.

Learning Outcomes

Competences

Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Apply the OSI model and TCP/IP operations effectively in network-related tasks and troubleshooting scenarios.
- Manage the implementation and configuration of client-server architecture using socket interfaces in real-world networking environments.
- Apply practical skills in Linux installation and usage.
- Demonstrate adeptness in socket programming using the C language.

Knowledge

Upon successful completion of this module, learners will have engaged with:

- The structure and functions of the OSI model and TCP/IP in the context of computer networking.
- How client-server architecture operates and the role of sockets in communication.
- The foundational concepts and historical context of the Linux operating system within the field of open-source computing.
- The functionality and components of DNS servers, clients, and various network security devices.

Skills

Upon successful completion of this module, learners will have developed:

- Proficiency in configuring network protocols based on the OSI model and TCP/IP.
- Capability to implement client-server architecture in application development, utilising sockets for communication.
- Competence in utilising Linux commands and tools effectively for system management and troubleshooting.
- Ability to configure DNS servers and clients, ensuring proper domain name resolution and network security measures.

Module-Specific Learner Skills

In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Advanced troubleshooting capabilities in network configurations.
- Enhanced communication skills for technical concepts.
- Proficiency in Linux system administration tasks.
- Ability to design secure network architectures.

Module-Specific Digital Skills and Competences

Learners will develop digital skills and competencies including:

- Proficiency in network protocol analysis tools like Wireshark.
- Advanced skills in Bash and Python language for automation purposes.
- Awareness and practice of cybersecurity measures.
- Competence in deploying and managing cloud computing environments.

Module Workload

Component	Hours
Total Contact Hours	30
Supervised Placement and Practice Hours	20
Self-Study Hours	95
Assessment Hours	5
Total Learning Hours	150
ECTS Credits	6

Contact Hours: Structured learning activities including theoretical lectures covering foundational concepts such as OSI model structure, TCP/IP operations, client-server architectures, Linux operating system fundamentals, DNS infrastructure, and network security principles.
Supervised Placement and Practice Hours: Practical laboratory sessions providing opportunities to apply knowledge through hands-on exercises in configuring network devices, programming socket applications, implementing security measures, and Linux system administration under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of networking and programming assignments, exploration of network tools and Linux environments, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, presentations, and the final examination.

Pedagogy

This module is delivered through a combination of theoretical instruction, practical application, and experiential learning:

- Theoretical Lectures: Introduction to foundational networking concepts including OSI model structure and functions, TCP/IP operations and protocols, client-server architecture and socket communication, Linux operating system fundamentals and history, DNS infrastructure components, and network security devices and measures.
- Practical Laboratory Sessions: Hands-on sessions enabling students to configure network devices, implement network protocols, develop socket applications using C programming, install and configure Linux systems, set up DNS servers and clients, and implement security measures.
- Hands-On Assignments: Programming and configuration exercises challenging students to apply learning to real-world scenarios, fostering critical thinking and problem-solving skills in network management and IT project implementation.
- Interactive Discussions: Facilitated sessions deepening understanding of technical concepts and encouraging peer learning and collaborative problem-solving.

- Case Studies: Analysis of real-world IT project management scenarios, network implementations, and organisational change initiatives illustrating practical applications and best practices.
- Digital Learning Tools: Use of network protocol analysis tools (Wireshark), programming environments (C, Bash, Python), Linux systems, network simulation tools, online submission platforms, code repositories, and video conferencing for instruction, collaboration, and assessment.

The pedagogical approach emphasises practical skill development and application of networking and IT project management principles to organisational contexts.

Assessment Strategy

Assessment Weightings

- Written Assignments: 40%
- Presentations: 30%
- Final Examination: 30%

Assessment Methods

Written Assignments (40%):

Learners will complete written assignments (maximum 1,250 words) addressing specific IT project management and networking topics, such as analysis of OSI model and TCP/IP protocols, evaluation of client-server architectures, examination of Linux system administration approaches, or assessment of network security measures. This assesses their ability to articulate technical concepts, critically evaluate networking solutions, and develop evidence-based recommendations.

Presentations (30%):

Learners will deliver an individual presentation (10 minutes, approximately 850 words or maximum 10 slides) communicating insights on IT project management, network configurations, socket programming implementations, Linux system administration, or network security architectures. This assesses their capability to communicate technical concepts effectively, present analytical findings, and engage professional audiences using digital presentation tools.

Final Examination (30%):

A comprehensive final examination will evaluate overall understanding of IT project management and advanced computer networks, including OSI model and TCP/IP operations, client-server architecture, socket programming, Linux system administration, DNS configuration, and network security measures. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy

Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, delivery of a revised presentation, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to laboratory facilities and networking tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading

Core Texts

- Tanenbaum, A.S. & Wetherall, D.J. (2020). *Computer Networks* (5th ed.). Pearson.
- Stallings, W. (2021). *Data and Computer Communications* (10th ed.). Pearson.
- Forouzan, B.A. (2021). *Data Communications and Networking* (5th ed.). McGraw-Hill.
- Lucas, M.W. (2021). *Network Performance and Security: Testing and Analyzing Using Open-Source Tools* (2nd ed.). No Starch Press.

Additional Reading

- Gillam, J.W. (2021). *Linux Network Programming* (2nd ed.). Springer.
- Martin, B. (2021). *Linux in Action* (2nd ed.). Manning Publications.
- Santos, O. (2021). *The Linux Command Line* (2nd ed.). No Starch Press.

UNIT SPECIFICATIONS

Unit Title
E-Commerce

Level
6

Learning Time Hours
150

Credit Value
6

Module Overview
The E-Commerce module provides students with comprehensive understanding of the principles and practices of electronic commerce in today's digital landscape, equipping them with skills necessary to design, implement, and manage effective e-commerce solutions. Students explore various aspects of online business including key e-commerce business models (B2B, B2C, C2C), technological infrastructure required for e-commerce operations including web development, payment systems, and security measures, online marketing strategies including digital marketing techniques, search engine optimisation (SEO), social media marketing, and customer relationship management (CRM), legal and ethical issues surrounding online transactions including data protection regulations and consumer rights, and e-commerce analytics tools for assessing performance and informing decision-making. Through combination of theoretical knowledge and practical application delivered in a fully online format using synchronous and asynchronous learning activities, live video sessions, recorded lectures, discussion forums, collaborative tools, practical assignments using e-commerce platforms and digital marketing tools, and webinars featuring industry experts, students develop the competencies to analyse user behaviour, track performance metrics, and propose innovative solutions to real-world e-commerce challenges.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Identify various e-commerce business models (B2B, B2C, C2C) and their implications for business operations and consumer behaviour.
- Implement e-commerce technologies, including web development tools, payment gateways, and security protocols, to design effective online business solutions.
- Create digital marketing strategies using SEO, social media, and other online marketing techniques to enhance brand visibility and customer engagement.
- Assess legal and ethical issues associated with e-commerce, including data protection regulations and consumer rights, to ensure compliance in online transactions.
- Utilise e-commerce analytics tools to analyse user behaviour and track performance metrics for data-driven decision-making.
- Propose innovative solutions to real-world e-commerce challenges by analysing case studies to enhance operational efficiency and customer satisfaction.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Various e-commerce business models (B2B, B2C, C2C) and their roles in the digital economy.
- The technological components of e-commerce systems, including web development, payment processing, and security measures.
- Digital marketing principles and strategies used in e-commerce, including search engine optimisation (SEO) and social media marketing.
- The legal and ethical considerations surrounding e-commerce, including data protection laws and consumer rights.
- Key metrics and analytics used to evaluate e-commerce performance and user engagement.
- The challenges and opportunities in the e-commerce landscape, including trends in consumer behaviour and technological advancements.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to develop a functional e-commerce website using web development tools and platforms.
- Proficiency in implementing secure payment processing systems to facilitate online transactions.
- Competence in creating and managing digital marketing campaigns using various online channels to drive traffic and sales.
- Ability to analyse user data and e-commerce metrics to evaluate website performance and customer behaviour.
- Skills in conducting research to identify legal and ethical standards applicable to e-commerce operations.

- Capability to collaborate effectively in teams to design and present comprehensive e-commerce solutions for real-world business scenarios.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Developing the ability to use web development tools and platforms to create and manage e-commerce websites.
- Gaining skills in analysing user data and e-commerce metrics to inform marketing strategies and enhance website performance.
- Acquiring the skills to design and implement effective digital marketing campaigns, utilising SEO, social media, and email marketing.
- Understanding and applying legal and ethical principles relevant to e-commerce transactions and data protection.
- Enhancing problem-solving skills through the analysis of case studies and the development of innovative solutions to real-world e-commerce challenges.
- Cultivating teamwork and communication skills by working collaboratively on e-commerce projects and presentations.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficiently using HTML, CSS, and JavaScript to design and build responsive e-commerce websites.
- Navigating and managing popular e-commerce platforms (e.g., Shopify, WooCommerce) for setting up and operating online stores.
- Implementing and managing secure digital payment gateways, including credit card processing and e-wallets.
- Applying SEO techniques to optimise e-commerce websites for better visibility and ranking on search engines.
- Utilising analytics tools (e.g., Google Analytics) to track website performance, analyse user behaviour, and generate reports for decision-making.
- Developing skills in managing social media accounts and creating targeted advertising campaigns to drive traffic and engagement on e-commerce platforms.

Module Workload

Component	Hours
Total Contact Hours	30
Supervised Placement and Practice Hours	20
Self-Study Hours	95
Assessment Hours	5
Total Learning Hours	150
ECTS Credits	6

Contact Hours: Structured learning activities including live video sessions where instructors introduce key concepts and facilitate interactive discussions, alongside access to recorded lectures, reading materials, and online resources.
Supervised Placement and Practice Hours: Practical assignments involving use of e-commerce platforms and digital marketing tools, discussion forums and collaborative tools enabling peer engagement, group projects simulating real-world e-commerce scenarios, and webinars featuring industry experts providing insights into current e-commerce trends under supervision.
Self-Study Hours: Independent learning activities including review of materials, completion of e-commerce development exercises, exploration of digital marketing tools and platforms, and preparation for assessments at learners' own pace.
Assessment Hours: Time allocated for completing written assignments, presentations, and the final examination.

Pedagogy
This module is delivered through a fully online format utilising a blend of synchronous and asynchronous learning activities:

- Live Video Sessions: Interactive lectures introducing key e-commerce concepts including business models, technological infrastructure, digital marketing strategies, legal and ethical considerations, and analytics, with facilitated discussions.
- Recorded Lectures: On-demand video content enabling flexible learning and review of e-commerce principles and practices at students' own pace.
- Online Resources: Access to reading materials, tutorials, and multimedia resources supporting understanding of e-commerce concepts and practical applications.
- Discussion Forums: Collaborative online spaces enabling students to engage with peers, share insights, discuss e-commerce challenges, and explore contemporary issues.
- Collaborative Tools: Digital platforms supporting group projects where students work together to design e-commerce solutions, simulating real-world business scenarios.
- Practical Assignments: Hands-on activities involving use of e-commerce platforms (Shopify, WooCommerce), web development tools (HTML, CSS, JavaScript), payment gateways, digital marketing tools, and analytics platforms (Google Analytics).
- Webinars with Industry Experts: Guest sessions providing valuable insights into current e-commerce trends, best practices, and emerging technologies.
- Online Learning Management System: Platform for submitting work, accessing resources, participating in discussions, and tracking progress.

The pedagogical approach creates an engaging virtual learning environment emphasising practical application of e-commerce principles and development of digital competencies.

Assessment Strategy
Assessment Weightings
- Written Assignments: 40%
- Presentations: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific e-commerce topics, such as analysis of business models, evaluation of digital marketing strategies, examination of legal and ethical considerations, or assessment of e-commerce technologies and security measures. This assesses their ability to articulate concepts, critically evaluate e-commerce approaches, and develop evidence-based recommendations.

Presentations (30%):
Learners will deliver an individual presentation (10 minutes, approximately 850 words or maximum 10 slides) communicating insights on e-commerce solutions, digital marketing campaigns, website design, analytics findings, or innovative approaches to e-commerce challenges. This assesses their capability to communicate effectively, present analytical findings, and engage professional audiences using digital presentation tools.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of e-commerce, including business models, technological infrastructure, web development, payment systems, security measures, digital marketing strategies, SEO, CRM, legal and ethical considerations, and analytics. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions conducted through online assessment platforms.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.
- Process: Reassessment may involve resubmission of assignments, delivery of a revised presentation, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including individual tutorials, access to e-commerce platforms and digital marketing tools, and online resources, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts
- Chaffey, D. (2024). *E-Business and E-Commerce Management: Strategy, Implementation and Practice*. [Publisher not specified].

- Miller, D. (2024). *Building a StoryBrand: Clarify Your Message So Customers Will Listen*. [Publisher not specified].
- Larsson, T. (2024). *Ecommerce Evolved: The Essential Playbook To Build, Grow & Scale A Successful Ecommerce Business*. [Publisher not specified].
- Laudon, K.C. & Traver, C.G. (2023). *E-Commerce 2023: Business, Technology, Society*. [Publisher not specified].
- Chaffey, D. (2021). *Digital Business and E-Commerce Management*. [Publisher not specified].
- Pantano, E. & Dennis, C. (2022). *Internet Retailing and Future Perspectives*. [Publisher not specified].
- Laudon, K.C. & Traver, C.G. (2021). *E-commerce: Business, Technology, Society*. Pearson.
- Chaffey, D. (2019). *Digital Marketing: Strategy, Implementation and Practice*. Pearson.
- Kalakota, R. & Whinston, A.B. (2017). *Frontiers of E-Commerce*. Addison-Wesley.
- Payne, A. & Frow, P. (2020). *Customer Relationship Management: Strategy and Implementation*. Cambridge University Press.
- Peltier, J.W. & Schibrowsky, J.A. (2018). *E-Commerce: Business Models, Technologies, and Applications*. Routledge.

Additional Reading

- Ries, E. (2024). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. [Publisher not specified].
- Pantano, E. (2020). *Retail Futures: The Good, the Bad and the Ugly of the Digital Transformation*. [Publisher not specified].
- Ecommerce Europe & EuroCommerce (2023). *European E-commerce Report 2023*. [Publisher not specified].
- Shapiro, C. & Varian, H.R. (1999). *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business Review Press.
- Clifton, B. (2020). *SEO: The Essential Guide to Search Engine Optimization*. Wiley.
- Turban, E. & Volonino, L. (2020). *Information Technology for Management: On-Demand Strategies for Performance, Growth and Sustainability*. Wiley.

UNIT SPECIFICATIONS

Unit Title
Knowledge Management

Level
6

Learning Time Hours
150

Credit Value
6

Module Overview
This module equips learners with comprehensive understanding of knowledge management (KM) processes, focusing on the creation, capture, representation, storage, and reuse of intellectual assets within organizations. Students apply tools and techniques for knowledge acquisition, assessment, evaluation, and dissemination to real-world business scenarios, gaining insights into knowledge generation, coordination, transfer, and reuse. The module explores key concepts, models, and frameworks in knowledge management, the role of digital technologies in capturing, storing, and disseminating knowledge, knowledge management software and digital tools facilitating collaboration (Microsoft SharePoint, Confluence), the impact of artificial intelligence (AI) and machine learning (ML) on improving knowledge management processes, security challenges and solutions for managing digital knowledge assets, and the importance of knowledge assets and intellectual capital in creating competitive advantage. Through asynchronous and synchronous online learning methods including pre-recorded video lectures, discussion forums, live webinars with instructors and industry experts, and practical assignments with knowledge management tools, the course enhances learners' ability to manage information technologies that support effective knowledge management strategies aligned with organizational objectives.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Manage the implementation of knowledge management systems within an organisation to ensure alignment with strategic objectives.
- Supervise teams in executing knowledge-sharing activities that enhance collaboration and innovation.

- Monitor the effectiveness of knowledge management practices and ensure their proper use in decision-making processes.
- Advise stakeholders on best practices and strategies for improving organisational knowledge management.
- Ensure compliance with organisational policies during knowledge-sharing initiatives and knowledge asset utilisation.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- The role of digital technologies in capturing, storing, and disseminating knowledge within organisational contexts.
- The functions of knowledge management software and digital tools in facilitating collaboration and knowledge sharing.
- The impact of artificial intelligence (AI) and machine learning (ML) on improving knowledge management processes.
- The security challenges and solutions associated with managing digital knowledge assets.
- Key concepts, models, and frameworks in knowledge management and their application in various organisational contexts.
- The role of technology in facilitating knowledge management systems and enhancing organisational learning.
- Different knowledge management strategies and tools to support decision-making and innovation.
- The importance of knowledge assets and intellectual capital in creating a competitive advantage for organisations.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to identify key concepts, models, and frameworks in knowledge management and their application in various organisational contexts.
- Proficiency in operating knowledge management processes to capture, store, and disseminate information effectively in an organisation.
- Ability to analyse the role of technology in facilitating knowledge management systems and enhancing organisational learning.
- Competence in evaluating different knowledge management strategies and tools to support decision-making and innovation.
- Skills in explaining the importance of knowledge assets and intellectual capital in creating a competitive advantage for organisations.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Applying knowledge management tools and technologies to enhance organisational learning and decision-making.
- Developing strategies for effective knowledge sharing and collaboration across teams and departments.
- Critically assessing the impact of knowledge management practices on organisational performance and innovation.
- Implementing processes for capturing and managing organisational knowledge assets to support business goals.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Utilising digital platforms and tools (e.g., intranets, knowledge repositories, and cloud-based systems) to manage and disseminate organisational knowledge effectively.
- Leveraging data analytics tools to extract insights from knowledge repositories and support decision-making.
- Implementing enterprise knowledge management systems (e.g., Microsoft SharePoint, Confluence) to facilitate collaboration and knowledge sharing across digital platforms.
- Securing digital knowledge assets using appropriate cybersecurity measures to ensure data integrity and compliance with regulations.

Module Workload

Component	**Hours**
Total Contact Hours	**30**
Supervised Placement and Practice Hours	**20**

Component	Hours
Self-Study Hours	**95**
Assessment Hours	**5**
Total Learning Hours	**150**
ECTS Credits	**6**

Contact Hours: Structured learning activities including pre-recorded video lectures providing foundational knowledge on knowledge management processes, frameworks, digital technologies, AI and ML applications, and security considerations, alongside live webinars offering real-time interaction with instructors and industry experts.
Supervised Placement and Practice Hours: Practical assignments enabling hands-on experience with knowledge management tools (SharePoint, Confluence, knowledge repositories, analytics platforms), discussion forums facilitating engagement and collaboration among students on knowledge management scenarios, and supervised implementation of knowledge-sharing strategies.
Self-Study Hours: Independent learning activities including review of curated reading materials, completion of knowledge management exercises, exploration of digital platforms and tools, engagement with resource library materials, and preparation for assessments.
Assessment Hours: Time allocated for completing written assignments, presentations, regular quizzes, and the final examination.

Pedagogy
This module is delivered online through a combination of asynchronous and synchronous learning methods:

- Pre-recorded Video Lectures: On-demand instructional content providing foundational knowledge on knowledge management concepts, processes, frameworks, digital technologies, AI and ML applications, enterprise systems, and security measures.
- Curated Reading Materials: Selected resources supporting understanding of knowledge management principles, tools, strategies, and best practices.
- Discussion Forums: Collaborative online spaces facilitating engagement among students, promoting knowledge sharing, peer learning, and exploration of real-world knowledge management challenges.
- Live Webinars: Synchronous sessions offering opportunities for real-time interaction with instructors and industry experts, providing insights into current knowledge management trends and practices.
- Practical Assignments: Hands-on activities enabling experience with knowledge management tools and platforms (SharePoint, Confluence, analytics tools) to implement strategies for capturing, storing, and disseminating organizational knowledge.
- Regular Quizzes: Formative assessments evaluating understanding of key concepts and reinforcing learning throughout the module.
- Virtual Office Hours: Individual support opportunities for clarifying concepts, discussing challenges, and receiving personalized guidance.
- Resource Library: Comprehensive collection of materials providing access to further learning resources, case studies, and industry examples.

The pedagogical approach creates an engaging online learning environment emphasising practical application of knowledge management principles and development of digital competencies for organizational contexts.

Assessment Strategy
Assessment Weightings

- Written Assignments: 40%
- Presentations: 30%
- Final Examination: 30%

Assessment Methods
Written Assignments (40%):
Learners will complete written assignments (maximum 1,250 words) addressing specific knowledge management topics, such as analysis of KM frameworks and processes, evaluation of digital technologies and AI/ML applications in knowledge management, examination of knowledge-sharing strategies, or assessment of enterprise knowledge management systems and their organizational impact. This assesses their ability to articulate concepts, critically evaluate knowledge management approaches, and develop evidence-based recommendations aligned with strategic objectives.

Presentations (30%):
Learners will deliver an individual presentation (10 minutes, approximately 850 words or maximum 10 slides) communicating insights on knowledge management implementations, digital platform comparisons (SharePoint vs. Confluence), knowledge asset management strategies, or innovative approaches to organizational learning and collaboration. This assesses their capability to communicate effectively, present analytical findings, and engage professional audiences using digital presentation tools.

Final Examination (30%):
A comprehensive final examination will evaluate overall understanding of knowledge management, including KM processes (gathering, assessment, storage, distribution, use), digital technologies and tools, AI and ML applications, enterprise systems, security considerations, knowledge assets and intellectual capital, and strategies for enhancing organizational performance and innovation. The examination will include a mix of multiple-choice, short-answer, and problem-solving questions conducted through online assessment platforms.

Digital learning tools such as online submission platforms, code repositories, and video conferencing software will be used for assessment.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of assignments, delivery of a revised presentation, and/or a resit examination, as appropriate.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including virtual office hours, access to knowledge management platforms and tools, and resource library materials, is available to help students address identified areas for development and prepare effectively for reassessment.

Essential Reading
Core Texts

- K.K. & G.T. (2023). *Knowledge Management for the Intelligent Organization: The Art of Turning Knowledge into Value*. Routledge.
- M.K. (2021). *Knowledge Management: Theoretical and Practical Perspectives*. Emerald Publishing Limited.
- F.H. & J.A. (2022). *Knowledge Management: A European Perspective*. Palgrave Macmillan.

Additional Reading

- C.M. (2021). *Digital Knowledge Management in Europe: Practices and Innovations*. IGI Global.
- E.T. (2022). *Knowledge Management Strategies: A European Approach to Management and Innovation*. Springer.

UNIT SPECIFICATIONS

Unit Title
Capstone Project

Level
6

Learning Time Hours
300

Credit Value
12

Module Overview
The Capstone Project module provides students with the opportunity to apply their knowledge and skills to solve real-world computing problems, demonstrating mastery of computer science concepts through comprehensive project work. Common objectives include applying knowledge from the programme, conducting in-depth research, showcasing problem-solving skills, demonstrating effective project management, communicating findings clearly, exhibiting technical proficiency, applying critical thinking, considering ethical considerations, and promoting collaboration. Students explore foundational theories in computer science including algorithms, data structures, computational complexity, and formal languages, acquire in-depth knowledge in specialized domains such as artificial intelligence or cybersecurity, develop strong research methodologies, apply project management frameworks using Agile

methodologies, and identify interdisciplinary connections to address complex problems. Through virtual platforms and online resources including remote guidance and mentorship, online research resources, virtual collaboration tools, cloud-based development environments, and digital documentation systems, students design and implement solutions, collaborate within multidisciplinary teams, adapt to new technologies, and demonstrate enhanced problem-solving, critical thinking, communication, creativity, and innovation capabilities.

Learning Outcomes
Competences
Upon successful completion of this module, learners will demonstrate the responsibility and autonomy to:

- Solve intricate challenges in computer science using advanced problem-solving techniques.
- Manage comprehensive projects through effective organisation and execution.
- Apply advanced concepts and tools relevant to specialised fields with technical expertise.
- Assess solutions critically to optimise project outcomes through critical thinking skills.
- Propose innovative approaches and solutions to enhance project effectiveness.
- Navigate uncertainties and obstacles with adaptability and resilience.
- Uphold ethical conduct and professional responsibility throughout project execution.

Knowledge
Upon successful completion of this module, learners will have engaged with:

- Foundational theories in computer science, including algorithms, data structures, computational complexity, and formal languages, in the context of developing problem-solving abilities.
- In-depth knowledge in a specific domain, such as artificial intelligence or cybersecurity.
- Strong research methodologies to critically evaluate existing literature and contribute to advancements in computer science.
- Project management frameworks to effectively plan, execute, and monitor projects using Agile methodologies.
- Interdisciplinary connections between computer science and other fields to address complex, real-world problems.

Skills
Upon successful completion of this module, learners will have developed:

- Capability to design and implement algorithms to solve complex problems efficiently, ensuring optimal performance in various applications.
- Proficiency in utilising development frameworks and tools to create robust software applications, leveraging modern programming practices.
- Ability to debug and troubleshoot software issues effectively, employing systematic approaches to enhance code quality and system reliability.
- Competence in collaborating within multidisciplinary teams to share knowledge, enhance project outcomes, and foster innovative solutions.
- Skills in adapting to new technologies and methodologies, continually improving capabilities to meet evolving project and industry demands.

Module-Specific Learner Skills
In addition to the core programme skills outlined in Section B, learners will demonstrate enhanced abilities in:

- Enhanced problem-solving skills to address complex challenges in computer science.
- Improved critical thinking abilities for analysing and evaluating technical solutions.
- Advanced communication skills for effectively conveying complex ideas to diverse audiences.
- Heightened creativity and innovation in proposing novel approaches to technical problems.

Module-Specific Digital Skills and Competences
Learners will develop digital skills and competencies including:

- Proficient use of cutting-edge software development tools and technologies.
- Competent utilisation of data analytics platforms for extracting valuable insights from extensive datasets.
- Skilled implementation of cybersecurity measures to safeguard digital assets and mitigate potential risks.
- Adept navigation and utilisation of cloud computing platforms for scalable and adaptable computing resources.
- Proficient application of emerging technologies like artificial intelligence and machine learning for pioneering solutions.

Module Workload

Component	**Hours**
Total Contact Hours	**60**
Supervised Placement and Practice Hours	**100**

Component	Hours
Self-Study Hours	**115**
Assessment Hours	**15**
Total Learning Hours	**300**
ECTS Credits	**12**

Contact Hours: Structured learning activities including virtual orientation sessions introducing project requirements, timelines, expectations, and access to online resources, alongside regular virtual meetings with faculty mentors for discussing project progress, addressing queries, and receiving feedback on research and implementation strategies.
Supervised Placement and Practice Hours: Remote project implementation using virtual development environments, cloud-based platforms, version control systems, and collaboration tools, virtual collaboration through video conferencing, discussion forums, and project management software, online reviews and presentations to faculty mentors and peers, and supervised development and testing of solutions.
Self-Study Hours: Independent learning activities including utilisation of online libraries, databases, and academic journals for research and literature review, implementation work on projects, exploration of technologies and methodologies, preparation of documentation and reports, and reflection on learning experiences through online reflection activities and self-assessment tools.
Assessment Hours: Time allocated for completing project deliverables including presentations and demonstrations, thesis/project reports, software coding, and documentation.

Pedagogy
This module is delivered through virtual platforms and online resources, with the teaching approach adapted to the digital environment:

- Virtual Orientation Sessions: Online workshops providing information about project requirements, timelines, expectations, introduction to faculty mentors, and guidance on accessing online resources.
- Remote Guidance and Mentorship: Faculty mentors provide support through video conferencing, email, and online chat platforms, with regular virtual meetings scheduled to discuss project progress and provide feedback on research and implementation strategies.
- Online Research Resources: Access to online libraries, databases, and academic journals enabling students to conduct research, gather literature relevant to project topics, and stay updated with latest developments in their field.
- Virtual Collaboration Tools: Online platforms including video conferencing, discussion forums, and project management software facilitating peer collaboration, communication, virtual group discussions, resource sharing, and remote collaboration on project tasks.
- Remote Project Implementation: Students implement projects remotely using virtual development environments, cloud-based platforms, collaboration tools, online coding platforms, version control systems like GitHub, and testing environments to develop and test solutions.
- Online Presentations and Reviews: Reviews, presentations, and evaluations conducted through video conferencing tools, enabling students to present project progress, findings, and outcomes to faculty mentors and peers virtually, with feedback and discussions facilitated through online platforms.
- Digital Documentation and Reporting: Project documentation, reports, and presentations created and shared digitally using online document editing tools, slide presentation software, and virtual whiteboards, with electronic submission of deliverables and online evaluations.
- Remote Reflection and Evaluation: Students reflect on learning experiences and project outcomes through online reflection activities, discussion forums, and self-assessment tools, with evaluation of performance and outcomes conducted remotely and feedback provided digitally.

The pedagogical approach ensures effective delivery of instruction and support in the digital environment while maintaining high standards of project quality and learning outcomes.

Assessment Strategy
Assessment Weightings

- Presentation and Demonstration: 40%
- Thesis/Project Report: 30%
- Software Coding: 20%
- Documentation: 10%

Assessment Methods

Presentation and Demonstration (40%):
Learners will deliver a comprehensive presentation and demonstration of their capstone project to faculty mentors and peers using video conferencing tools. This component assesses overall presentation quality, clarity of writing, coherence of argumentation, adherence to academic conventions (referencing), communication effectiveness, technical demonstration proficiency, and ability to respond to questions and defend design decisions. The presentation showcases the completed solution, its functionality, and the problem-solving approach employed.

Thesis/Project Report (30%):
Learners will prepare a detailed thesis/project report documenting their chosen programming language, algorithms used, test cases, problem definition, and justification of their appropriateness for addressing the identified problem. The report includes discussions on data collection methods, analysis techniques, ethical considerations, literature review, methodology, implementation details, results, evaluation, and conclusions. This assesses their ability to articulate technical concepts comprehensively, critically evaluate approaches, and communicate findings clearly through written documentation.

Software Coding (20%):
Learners will develop well-defined, error-free code following strong syntax and semantics standards using appropriate development frameworks and tools. The coding component assesses technical proficiency, algorithm design and implementation, software quality, debugging and troubleshooting capabilities, use of modern programming practices, and adherence to coding standards. Code is submitted through online platforms and evaluated for functionality, efficiency, maintainability, and documentation quality.

Documentation (10%):
Learners will submit complete documentation outlining project introduction, problem statement, objectives, methodology, and expected outcomes. Documentation is assessed for clarity, feasibility, alignment with academic standards, comprehensiveness, organization, and adherence to conventions. This component includes project proposals, design documents, user manuals, technical specifications, and other supporting materials that demonstrate thorough project planning and execution.

Digital Learning Tools

- Learning Management System (LMS): Used for document submission, announcements, progress tracking, and resource distribution.
- Plagiarism Detection Software: Tools like Turnitin or SafeAssign ensure academic integrity in project submissions.
- Research Databases: Access to online libraries and databases for literature review and research data gathering.
- Collaboration Tools: Platforms like Google Workspace or Microsoft Teams for communication and collaborative work with supervisors and peers.
- Version Control Systems: GitHub and similar platforms for code management and collaboration.

Pass Mark: 50%

Reassessment Policy
Students who do not achieve the required standard are afforded one opportunity for reassessment.

- Process: Reassessment may involve resubmission of project deliverables, revised presentations and demonstrations, updated thesis/project reports, improved software coding, and enhanced documentation, as appropriate based on identified deficiencies.
- Timing: Reassessments are scheduled in accordance with the academic calendar, with specific deadlines communicated to affected students.
- Support: Additional academic support, including virtual mentorship sessions, access to online resources and development tools, and targeted guidance on addressing identified areas for improvement, is available to help students prepare effectively for reassessment.

Essential Reading
Core Texts

- Polonsky, M.J. & Waller, D.S. (2022). *Designing and Managing a Research Project: A Business Student's Guide*. [Publisher not specified].
- Booth, W.C., Colomb, G.G. & Williams, J.M. (2020). *The Craft of Research*. [Publisher not specified].
- Saunders, M., Lewis, P. & Thornhill, A. (2021). *Research Methods for Business Students*. [Publisher not specified].
- Bell, J. & Waters, S. (2020). *Doing Your Research Project*. [Publisher not specified].
- O'Leary, Z. (2023). *The Essential Guide to Doing Your Research Project*. [Publisher not specified].
- Bock, J. (2019). *The Capstone Project: A Comprehensive Guide*. Springer.
- Blanco, S. (2018). *Research Methods for Computer Science: A Practical Guide*. Academic Press.
- Pfleeger, S.L. & Pfleeger, C.J. (2015). *Security in Computing*. Pearson.

- Satzinger, J.W., Jackson, R.B. & Burd, S.D. (2016). *Systems Analysis and Design in a Changing World*. Cengage Learning.
- Schwalbe, K. (2018). *Information Technology Project Management*. Cengage Learning.
- Laudon, K.C. & Laudon, J.P. (2020). *Management Information Systems: Managing the Digital Firm*. Pearson.

Additional Reading

- Burke, J. & Dempsey, M. (2021). *Undertaking Capstone Projects in Education: A Practical Guide for Students*. [Publisher not specified].
- Burke, J. & Dunne, S. (2021). *Undertaking Capstone and Final Year Projects in Psychology: Practical Guide for Students*. [Publisher not specified].
- Sommerville, I. (2019). *Software Engineering*. Addison-Wesley.
- Bennett, S. & Fowler, J. (2018). *Developing Quality Technical Information: A Handbook for Writers and Editors*. Wiley.