BUISNESS AND TECHNOLOGY INTERNATIONAL CAMPUS

B-TIC LEVEL 4 & 5

EXTENDED DIPLOMA IN APPLIED COMPUTING

VERSION 01

September 2025

B-TIC

PROFESSIONALISM SIMPLIFIED

16/09/2025

CONTENTS

Changes of Specification with Previous Version

Previous Version Number	-	Current Version Number	01
Summary of changes			**Page number**
N/A			N/A

QUALIFICATION TITLE			B-TIC LEVEL 4 AND 5 EXTENDED DIPLOMA IN APPLIED COMPUTING		
Qualification Number (QN)			L4 & L5 - 250808		
Level			Level 4 & Level 5		
Duration			2 Years		
Grading System			Pass / Fail		
Awarding Organization			Business and Technology International Campus LTD		
Credit Value			240		
Total Qualification Time (TQT):	2400	**Guided Learning Hours (GLH)**	1200	**Independent Learning Hours (ILH)**	1200
Entry Requirements:			Learners must be at least 17 years old. A level 3 qualification or Equivalent IELTS 5.5 or equivalent for non-native speakers. No formal requirement		
Mode of Delivery			Full Time, Part Time, Face to Face Blended Distanced or any other mode that meet the learner needs without any unlawful or unfair discrimination to any learner		
Assessment Method			Course work and appropriate methods		
Language of Instruction & Assessment			English		
Placement requirements			N/A		

About this Qualification

The B-TIC Level 4 and 5 Extended Diploma in Applied Computing is designed to equip learners with a strong blend of theoretical knowledge and practical skills required for success in the digital and IT sector. The programme develops core expertise in areas such as software development, databases, networking, cybersecurity, data analysis, web and mobile applications, and IT project management.

Learners are encouraged to apply their knowledge to real-world scenarios, strengthening problem-solving, critical thinking, and innovation. Through a balance of classroom learning, practical projects, and independent research, the qualification prepares graduates for progression to higher education or direct entry into professional roles within computing and related industries.

Qualification Objectives:

1. To provide knowledge, skills and understanding to achieve high performance.
2. To assist every learner, a successful completion of qualifications meets learner needs through a variety of delivery modes with location freedom.
3. To allow every learner a successful completion of qualification at their own pace with time freedom
4. To make opportunities to enter employment or to progress in current employment
5. To make eligible to progress to higher education qualifications in related fields
6. To provide navigation map for professional body memberships

How to meet these objectives:

1. Teaching & assessing an up-to-date skill, knowledge and understanding for higher performance.
2. Assisting a range of delivery, learning and assessment modes
3. Allowing flexible timeframes and duration to complete the qualifications.
4. Help to ground in employment or to progress in current employment as learners apply to divers roles and responsibilities in the relevant industry
5. Developing university partnerships & Developing academic skills for learners and helping appropriate unit selection to advanced entry progression to higher education in universities
6. Assisting progression to professional qualifications from professional bodies

To Develop Transferable Skills

Transferable skills (employability skills) are core to improving the career prospects and personal professional development of learners.

B-TIC qualifications intend to develop the following skills:

Problem-solving skills

- Critical thinking
- Creative skill to solve routine & non-routine problems
- Digital technology skills and

Independent skills

- Self-awareness and self-management
- Adaptability and resilience
- Reflection, planning and
- Prioritising.

Interpersonal skills

- Leadership skills
- Communication skills & Presentation Skills
- Team-ability skills
- Negotiating skills

Commercial skills

- Awareness of the relevant industry sector
- Understanding customer needs
- Finance & Budgets
- Management Skills

Qualification Framework

B-TIC commits all the necessary actions to design its qualification in accordance with the Framework for Higher Education Qualifications (FHEQ) in England, Wales and Northern Ireland, the Regulated Qualifications Framework (RQF), the Quality Assurance Agency (QAA) Subject Sector Benchmarks and the European Qualification Framework. Further, B-TIC sought to design the qualification in accordance with the relevant professional body's requirements.

Equivalences

B-TIC Level 4 & 5 diplomas have been designed equivalent to Level 4 & 5 of the Regulated Qualifications Framework (RQF), which is at the same level as the diploma.

Collaboration in Qualification Design

Learners completing B-TIC qualifications may enter employment or progress to higher education. Thus, our qualifications are designed in collaboration with employers, universities, higher education institutes, professionals, professional bodies, businesses and educational institutions, learners and centres to ensure that qualifications meet the needs and expectations of stakeholders.

Progression

B-TIC develops articulation arrangements with universities locally and internationally that provide advanced entry for B-TIC learners from Level 3 to Level 8. However, university admission criteria may change from time to time and remain at their discretion.

On successful completion of the B-TIC qualifications, learners may:

- Progress to University with advanced entry
- Find an employment
- Continue current employment with progressive competencies
- Commit to Continuing Professional Development (CPD) and
- Start a business

Prior Learning Assessment and Recognition (PLAR)

PLAR is where learners can meet entry criteria to entire a course or unit assessment requirements through Skill, knowledge and understanding which they already possess.

PLAR can be referred by different institution as follows:

- Accreditation of Prior Learning (APL)
- Accreditation of Prior Experiential Learning (APEL)
- Accreditation of Prior Achievement (APA)
- Accreditation of Prior Learning and Achievement (APLA)

B-TIC has developed a PLAR policy that allows learners to skip formal learning which learners already know with appropriate demonstration as per the B-TIC PLAR Policy.

Equality and Diversity

B-TIC provide equal opportunity for every learner without any disadvantage in terms of age, disability, gender, marital status, civil partnership, pregnancy, maternity, race, colour, nationality, ethnicity, national origins, religion, sexual orientation or any other grounds in accordance with the B-TIC Equality and Diversity Policy.

Where a feature of a specific qualification disadvantage group of learners we provide explicit clarification In this regard in the qualification specification.

Qualification Structure

The B-TIC Level 4 & 5 Extended Diploma in Applied Computing consists of 6 mandatory units for each level.

Unit Number	Compulsory Units	Credit	TQT	GLH	ILH
4COM1	Academic Writing and Computer Application	20	200	100	100
4APC2	Applied Programming and Problem Solving	20	200	100	100
4APC3	Networking and Secure Communications	20	200	100	100
4APC4	Database Design and Development	20	200	100	100
4APC5	Web Design and Development	20	200	100	100
4APC6	Mathematics for Computing and Algorithmic Thinking	20	200	100	100
Compulsory Units					
5APC1	Enterprise Application Development	20	200	100	100
5APC2	Advanced Database Design	20	200	100	100
5APC3	Data Engineering and Big Data Technologies	20	200	100	100
5APC4	Cloud Computing	20	200	100	100
5APC5	Introduction to AI/ML	20	200	100	100
5APC6	Industry Investigation and Employement	20	200	100	100

Total Qualification Time (TQT)

Total Qualification Time (TQT) is the total count of hours a learner would commit to complete on all varieties of activities demonstrating learning outcome achievements of a specific whole qualification.

TQT consists of "Guided Learning Hours" (GLH) activities and all the other forms of "Independent Learning Hours" (ILH) activities.

Guided Learning Hours (GLH)

Guided Learning Hours (GLH) are the estimated time duration that the leaner is guided to learn in facilitation of Lecturers, Supervisors, Tutors, Trainers and facilitated workshops (hereafter called "FACILITATORS") etc. for learners enabling to complete module learning outcomes at the appropriate standard.

GLH counts real time activities completed by the learner under direct instruction, supervision and / or immediate guidance) via Physical or Virtual Attendance such as,

- Supervised induction sessions
- Class room lecturers
- Class room discussions
- Assignments Reviews
- Personal meetings
- Learner feedback with a teacher in real time
- Supervised independent learning
- Supervised Literature Review
- Classroom-based learning supervised by a teacher
- Work-based learning supervised by a teacher
- Question and answer sessions
- Presentation and Feedback
- Live webinar or telephone tutorial with a teacher in real time
- E-learning supervised by a teacher in real time
- E-mail conversations
- Formative assessment and feed back
- Summative assessment and feed back
- Professional Discussions
- Interview
- Reviewing cases with lecturers
- Building a poster with facilitator
- Invigilated examinations or assessments.

All other forms of learning activities under immediate guidance, Instruction, observation or Supervision of a facilitator may be counted into Guided Learning Hours (GLH).

Independent Learning Hours (ILH)

"Independent Learning Hours" (ILH) are hours of learning activities that learners undertake without direct instruction, supervision or immediate guidance.

The verity of independent learning activities in absence of direct instruction, supervision or Immediate Guidance of facilitator may happen depending on the qualifications and level of qualification, such as

- Independent preparation for classes
- Independent reading
- Independent research

- Independent work on projects
- Independent compilation of a portfolio of work experience
- Independent e-learning
- Independent e-assessment
- Independent coursework
- Independent watching a pre-recorded podcast or webinar
- Independent work-based learning
- Independent discussion
- Independent voluntary work

Any other form of learning, education or training, in the absence of direct instruction, supervision or immediate guidance of a facilitator, may be counted as Independent Learning Hours (ILH).

Credit

The credit value specifies the awarded number of credits to a learner who has covered and assessed the learning outcomes of a unit at a specific level standard of a qualification.

The awarded credits are recorded in a credit transcript that can lead to a qualification that supports your entry to the next progress of higher education.

Each credit represents 10 hours of learning time, which is equivalent to 10 hours of “TOTAL QUALIFICATION TIME”.

The credit value of the unit is constant in all contexts regardless of result (Pass or Fail), the assessment method or the mode of delivery.

Learners will only be awarded credits for the successful completion of the whole unit.

Specific requirement of individual learners and the differing delivery styles may cause variation in the actual time to complete a qualification; thus, Values for Total Qualification Time, Guided Learning Hours, and Credit are estimates

Unit Specification Level 04

Academic Writing and Computer Applications

Unit Title	Academic Writing and Computer Applications
Unit Code	4COM1
Unit Level	Level 4
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This unit intends for the learners to understand, develop skill sets and apply academic writing skills for academic assessments of high quality.

Unit Objectives

- To study academic writing skills for academic purposes.
- To use an appropriate range of physical and electronic academic sources.
- To learn to use the work of others for academic purposes
- To use different referencing styles
- To develop methodical approach to write academic reports and presenting to professional audience
- To develop presentation documents and to Oral present before professional audience
- To learn computer application skills & Office processing packages for academic writing

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to approach academic question to answer	1.1 Answer the question with reliable and valid academic argument.	Command word; Topic word; Scope word; Answer the question; Develop the position; Evidence based reference; verifiable reference; Impersonal reference; Objective; Qualified; Precise reference (reliable and exact)
	1.2 Critically review Literature	Critical Thinking Model; Critical thinking; critical reading; Critical writing describe (What, When, Where, Who); analyze (How, Why, If so); evaluate (so what, is next); Bookmarking, note-taking, and synthesising.
	1.3 Follow writing structure	TDR structure; IBC structure; five-paragraph essay; report writing; abstract; executive summary; presentation structure; proposal structure.
LO 2 Be able to search academic sources for academic writing	2.1 Use different types of academic Sources and evidence	Books; journals; newspapers; YouTube; social media; Wikipedia platforms; television news; theses; conference papers; reports; lecturer notes; statistics; interviews; blogs; ads; maps; diagrams; observations; websites. Government data.
	2.2 Search Academic sources	Search engines: Google, Search GPT, Google scholar; online libraries: Questia, Open Library; Learning management systems; index journals; Kindle; Title of the source; title of the journal; author's Name;
	2.3 Evaluate different Sources of information	Published in a scientific or academic journal; published book: currency; age of source; another academic expert or specialist; written in professional or academic language; provide references; Relevancy:
LO 3 Be able to construct a reliable and valid academic answer.	3.1 Understand the common types of plagiarism	Acknowledgment; risk avoidance; plagiarism ethics; justification Clone; hijack; inaccurate referencing; patchwork; Cutting and pasting from other documents; Quoting without quotation marks or references; Paraphrasing without referencing; Summarising without referencing; Using an image, source and/or diagram without.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to construct a reliable and valid academic answer.	3.2 Use academic sources to write academic answers	Academic language; Paragraphs and argument flows; Writing to word counts; Paraphrasing; Summarising; Direct quotes Referencing academic works; Harvard style; Oxford style; OSCOLA; Vancouver style; American Psychological Association style; EndNote; footnote in-text citation; single author citation; multiple author citation; Author as organisation; Signpost" words and" phrases Introductory Phrases: introduce a sequence of points. Cross-referencing; adding ideas; exemplifying; focusing on a specific point; generalisation; concession; similarity; comparison; contrast; alternative point; stating the obvious; rephrasing; and introducing a reason, result, proof or logical conclusion. Building on the last section, conclusion.
	3.3 Use sources to present academic answers	Presentation PPT; visual image; posture; body language; referencing Declamation; audience analysis; paralanguage; gesture; movement; eye contact; vocal variety; listening; poise.
LO 4 Be able to use computer applications in academic writing	4.1 Apply word processing applications	Word Processing
	4.2 Apply presentation applications	PowerPoint Processing
	4.3 Apply Statistical computer applications	Spreadsheet; Excel Processing SPSS; Minitab

Essential Reading and Resource List:

Text books:

Craswell, G., and Poore, M. (2012) *Writing for academic success.* 2nd ed. New York: SAGE Publications
Gaskin, S., Vargas, A., Geoghan, D., and Graviett, A. (2020) *GO! All in one: Computer concepts and applications.* 4th ed. Harlow: Pearson.
Nordell, R. (2022) Microsoft Office 365: In practice. McGraw-Hill Education.
Murray, R. (2019) Writing for academic journals. 4th ed. Open University Press.
Poatsy, M. A., and Grauer, R. T. (2021) Exploring Microsoft Office 365: Excel, Word, PowerPoint, and Access in depth. Harlow: Pearson.
Rainer, R. K., & Prince, B. (2023) Introduction to information systems: Enabling and transforming business. 10th ed. New Jersey: Wiley.
Swales, J. M., & Feak, C. B. (2012) *Academic writing for graduate students: Essential tasks and skills.* 3rd ed. University of Michigan Press.

Journals & Newspapers:

Journal of Second Language Writing
Journal of English for Academic Purposes
Across the Disciplines
Applied Computer Systems
IEEE Computer Society Digital Library
Computers and Mathematics with Applications

Websites:

https://owl.purdue.edu
http://www.phrasebank.manchester.ac.uk

Applied Programming and Problem Solving

Unit Title	Applied Programming and Problem Solving
Unit Code	4APC2
Unit Level	Level 4
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module develops programming skills and problem-solving techniques through practical applications. Students learn to design, implement, and test solutions, enhancing logical thinking, coding proficiency, and analytical abilities for real-world computing challenges.

Unit Objectives

- To understand and apply fundamental programming concepts and syntax to develop functional code.
- To implement control flow structures to create efficient, logical program behavior.
- To collaborate effectively on coding projects using version control and teamwork principles.
- To test and debug programs systematically to ensure accuracy, reliability, and performance.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to understand fundamental programming concepts and syntax.	1.1 Explain fundamental programming constructs	Variables and data types; primitive vs complex types; role of constants and literals; syntax rules and naming conventions; use of conditionals for decision making (if, else, switch); looping structures (for, while, do-while); control flow management; error prevention with proper use; best practices for code readability; initialization and assignment; scope and lifetime of variables; debugging common syntax errors; importance of comments and documentation; code maintainability and clarity.
	1.2 Write code using basic syntax and naming conventions	Readable code; well-structured code; Consistent naming conventions; indentation and whitespace usage; modular code blocks; use of functions for reusability; commenting style and placement; descriptive identifiers; logical grouping of statements; avoiding code duplication; error handling practices; code formatting standards; readability for collaborative development; following language-specific style guides; separation of concerns.
	1.3 Identify and resolve syntax and logical errors in sample code	Types of syntax errors; common logical errors; debugging techniques; use of error messages; testing with sample inputs; compiler/interpreter feedback; use of debugging tools and breakpoints; systematic troubleshooting; validating output correctness; fixing infinite loops and off-by-one errors; code tracing and walkthrough; error prevention strategies; writing test cases.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 2 Be able to implement programs using basic data structures and control flow mechanisms.	2.1 Choose appropriate data structures for a given problem	Selects suitable structures based on access pattern and data type; creates and modifies arrays, lists, sets, and dictionaries; performs operations like append, insert, pop, update, and clear; accesses data using keys and indices; applies slicing and iteration; uses nested structures (e.g., list of dictionaries); chooses mutable vs immutable types as needed; uses data structures in real-world contexts such as inventory, records, and user profiles.
	2.2 Apply control flow using conditionals, loops, and function calls	Implement decision-making using if-else, elif, and nested conditions; applies switch/case or match statements where applicable; writes for and while loops for repeating tasks; uses break and continue to control loop behavior; defines and invokes user-defined functions with parameters and return values; uses recursion or nested loops when appropriate; manages basic input/output for interaction; organizes code using structured logic blocks.
	2.3 Develop programs that combine logic, data, and structure to solve simple tasks	Builds console-based tools (e.g., calculator, number guessing game, simple CRUD); integrates loops, functions, and conditionals effectively; validates user input (e.g., data type, range, format); uses menus or prompts for interaction; structures code for clarity and reuse; handles basic errors or edge cases; stores temporary data using appropriate structures; demonstrates logical flow and output correctness in final solutions.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to use version control and collaborative coding practices effectively.	3.1 Use Git and GitHub for code versioning and collaboration	Initializing repositories; staging and committing changes; branching and merging; resolving conflicts; pull requests and code reviews; collaboration workflows; version history navigation; using remotes and push/pull operations; tagging and releases; reverting changes; managing branches; understanding distributed version control concepts.
	3.2 Follow best practices for commit messages, branches, and pull requests	Writing clear, concise commit messages; branch naming conventions; creating feature and bugfix branches; effective pull request descriptions; review feedback handling; continuous integration triggers; avoiding large commits; maintaining clean history; merge strategies; collaboration etiquette.
	3.3 Participate in peer review and collaborative coding exercises effectively	Providing constructive feedback; responding to reviews professionally; integrating suggestions; resolving merge conflicts; understanding code quality; team communication; shared coding standards; pair programming techniques; task division; knowledge sharing.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 4 Be able to test, debug, and evaluate code for quality, maintainability, and performance.	4.1 Understand how to write unit and integration tests using modern testing frameworks	Writing test cases for functions and modules; using frameworks like Jest, JUnit, pytest; test coverage importance; automating tests; test-driven development basics; integration vs unit testing differences; mock objects and stubs; handling edge cases; regression testing; continuous testing in pipelines.
	4.2 Explain how to use debugging tools and strategies	Using breakpoints and watch variables; reading stack traces; isolating defects; testing hypotheses; stepwise execution; log file analysis; performance bottleneck detection; root cause analysis; regression bug fixes; verifying bug fixes.
	4.3 Evaluates and improves code quality based on performance, readability, and maintainability	Refactoring for simplicity; removing dead code; optimizing algorithms; improving naming and structure; reducing complexity; profiling and benchmarking; documenting code; adhering to coding standards; peer reviews.

Essential Reading and Resource List:

Textbooks:

Deitel, P. and Deitel, H. (2017) *Intro to Java Programming: Comprehensive Version*. 11th edition. Harlow: Pearson.
Conn, C. & McLean, R. (2019) – *Bulletproof Problem Solving: The One Skill That Changes Everything*. Hoboken: Wiley.
Watanabe, K. (2019) Pr*oblem Solving 101: A Simple Book for Smart People.* New York: Portfolio.

Journals & Newspapers:

Educational Computing (2024)
BJET (2024)
Human-LLM study (2025)
CDIO & Sustainability (2025)
Digital Twin study (2025)
Think-Aloud Study (2022)

Websites:

https://www.geeksforgeeks.org
https://docs.github.com

Networking and Secure Communications

Unit Title	Networking and Secure Communications
Unit Code	4APC3
Unit Level	Level 4
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module explores networking principles and secure communication techniques. Students learn to design, configure, and protect networks, applying protocols, encryption, and security measures to ensure reliable, safe, and efficient data exchange in modern systems.

Unit Objectives

- To explain the fundamentals of computer networks and their key components.
- To apply TCP/IP principles to configure and troubleshoot network communications.
- To analyze and implement various network protocols for efficient data transfer.
- To evaluate and apply security measures to protect network systems and communications.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to understand fundamental networking concepts, protocols, and the TCP/IP model.	1.1 Describe OSI and TCP/IP layers and their roles	Definitions of OSI and TCP/IP models; structure and purpose of each layer from physical to application; role of encapsulation and de-encapsulation in data flow; differences between the two models and their practical use cases; significance of layered design in networking; common protocols at each layer (e.g., IP, TCP, HTTP, Ethernet); end-to-end communication; error detection and recovery at different layers; benefits of modular design for troubleshooting and scalability.
	1.2 Explain common protocols and their uses	Definitions and characteristics of HTTP, TCP, UDP, and IP; differences between connection-oriented (TCP) and connectionless (UDP) protocols; role of IP in addressing and routing across networks; function of TCP in ensuring reliability and error recovery; role of UDP in supporting low-latency, best-effort applications (e.g., video, gaming); significance of HTTP in application-level communication; understanding port numbers and their association with services; examples of real-world uses of each protocol.
	1.3 Identify network devices (switch, router, firewall) and their roles	Characteristics and purpose of switches in MAC-based data forwarding and LAN segment connectivity; role of routers in IP routing, path determination, and traffic forwarding across networks; understanding firewall roles for traffic inspection, access control, and threat mitigation; differences between Layer-2 and Layer-3 devices and their responsibilities in a network; examples of common device uses within enterprise and home environments; significance of network design for scalability, security, and efficiency.
LO 2 Be able to design and configure a basic network topology, including IP addressing and subnetting.	2.1 Construct IP addressing and subnetting schemes	Definitions of IP addressing and subnetting; IPv4 and IPv6 addressing differences and notation; CIDR and its role in defining subnet boundaries; subnet masks and their use in segmenting IP networks; calculating network, broadcast, and host ranges; considerations for private vs public IP addressing; planning address allocations for scalability and efficiency; impact of subnet design on traffic routing and isolation.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
	2.2 Configure routers, switches, and end devices for connectivity	Basics of setting IP addresses and subnet masks on routers, switches, and end devices; establishing routing between local area and external networks; using switch configuration for VLAN assignment and traffic separation; verifying IP settings for correct connectivity and data flow; understanding static vs dynamic addressing methods (e.g., DHCP); role of gateways in inter-network communication; applying access controls and port security settings for basic protection.
	2.3 Verify connectivity using tools like ping and traceroute	Purpose of network diagnostic tools (ping, traceroute) for testing IP connectivity; interpreting results for latency, packet loss, and route analysis; identifying common connectivity issues from command output; using ICMP and other protocols for connectivity verification and troubleshooting; isolating network segments with traceroute for targeted problem solving; verifying end-to-end connections across routers and switches; assessing network performance and identifying bottlenecks for improvement.
LO 3 Be able to learn and troubleshoot common network connectivity and performance issues.	3.1 Use command-line and analysis tools (ping, traceroute, Wireshark) for diagnostics	Purpose and function of network diagnostic tools in identifying connectivity and performance issues; use of ping for basic IP connectivity and packet loss detection; traceroute for mapping the path and identifying bottlenecks across routers and links; Wireshark for packet capture, inspection, and protocol-level analysis; interpreting tool outputs for actionable insights and issue resolution; application of these tools in both LAN and WAN environments; identifying trends and anomalies that impact performance and reliability.
	3.2 Identify and resolve common network configuration errors	Recognizing IP conflicts and addressing scheme errors across devices and segments; resolving misconfiguration of subnet masks, gateways, and DNS settings; understanding the role of routing tables and static vs dynamic routing errors; verifying switch port and VLAN assignment for connectivity consistency; applying command-line troubleshooting steps (show commands, interface status) to isolate faults; restoring connectivity by applying configuration best practices; documenting issues and preventive measures for future reference.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
	3.3 Explain network topology and test results	Developing accurate network topology maps for design review and maintenance; recording IP addressing schemes and device roles within a network for traceability; explaining test results and observations clearly for team collaboration; highlighting root causes and proposed fixes in incident documentation; using schematic tools and templates for visual network illustrations; aligning test outcomes with design objectives and performance standards; utilizing documentation for future network optimization and resilience planning.
LO 4 Be able to evaluate network security threats and implement basic protective measures.	4.1 Identify common threats (malware, DoS, phishing) and their impacts	Recognizing common network and application-level threats and their behavior patterns; understanding the impact of malware, DoS, and phishing attacks on connectivity, data integrity, and service availability; identifying entry points and vectors for attacks within a network environment; assessing potential risk levels and damage scenarios for an organization; correlating threat types with vulnerabilities in protocols, devices, and applications; reinforcing the role of staff awareness and policies in threat mitigation.
	4.2 Analyze basic security controls (firewalls, ACLs, strong passwords)	Applying firewall rules and access control lists (ACLs) to filter and govern traffic flow across the network; creating strong password policies and account management practices for access security; deploying basic intrusion detection or prevention measures for critical points within the topology; applying device hardening techniques (disabling unused services, updating firmware) to reduce attack surfaces; segmenting the network to isolate sensitive or high-risk areas; aligning security controls with best practices and industry standards.
	4.3 Evaluate security measures and recommend improvements	Assessing the effectiveness of implemented security controls through testing and review; identifying gaps in current security posture using audits and vulnerability scans; proposing enhancements in access controls, monitoring, and threat detection methods; aligning security policies with evolving threats and compliance requirements; creating actionable recommendations for future improvements in hardware, software, and protocols; promoting a culture of continuous improvement and resilience within network design and administration.

Essential Reading and Resource List

Textbooks:

Kurose, J. F. and Ross, K. W. (2020) *Computer Networking: A Top-Down Approach.* 8th edition. Boston: Pearson.
Forouzan, B. A. (2017) *Data Communications and Networking.* 5th edition. New York: McGraw-Hill Education.
Stallings, W. (2020) *Network Security Essentials: Applications and Standards*. 6th edition. Boston: Pearson.

Journals & Newspapers:

Computer and Network Security
IEEE Security & Privacy
IET Information Security
Foundations and Trends in Networking
Computer Networks
IEEE/ACM Trans. on Networking

Websites:

https://www.netacad.com
https://www.wireshark.org

Database Design and Development

Unit Title	Database Design and Development
Unit Code	4APC4
Unit Level	Level 4
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module covers database design and development principles. Students learn to model data, create efficient database structures, implement queries, and ensure data integrity for effective storage, retrieval, and management in real-world applications.

Unit Objectives

- To explain the principles and structure of relational databases.
- To design and model databases using appropriate tools and techniques.
- To write and execute SQL statements for data manipulation and retrieval.
- To apply querying methods to analyze and manage stored data effectively.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to understand relational database concepts and data modeling techniques.	1.1. Define relational database concepts and terminology	structure and role of relational databases in data storage and retrieval; tables, rows, columns, and primary keys; highlights the role of relationships (one-to-one, one-to-many, many-to-many) in data modeling; relational databases for data consistency, integrity, and normalization; schema, constraints, and indexes for database design.
	1.2 Construct and explains entity-relationship (ER) models for data design	entities, attributes, and relationships; cardinality and participation constraints; standard ER notation (crow's foot, UML); models one-to-one, one-to-many, and many-to-many relationships; composite and derived attributes; ER models to relational schemas; completeness and accuracy; best practices for clarity; models for stakeholder understanding; business rules; models based on feedback; ER models with system requirements.
	1.3 Apply normalization techniques to optimize database design	Normalization purpose and benefits; 1NF, 2NF, and 3NF rules; data redundancy and anomalies; decomposes tables to meet normal forms; normalization with query performance; understands functional dependencies; trade-offs in denormalization; normalization process; constraints to maintain integrity; normalization in design workflow; tools or manual methods; design correctness.
LO 2 Be able to design normalized databases that adhere to best practices for efficiency and scalability.	2.1 Design normalized relational schemas based on business requirements	Schemas using normalization rules (1NF, 2NF, 3NF); primary and foreign keys correctly; referential integrity; normalization with system performance; appropriate data types and constraints; schema design clearly; models relationships accurately; Scalability and future growth; indexing strategies; to naming conventions; storage efficiency; schema based on feedback.
	2.2 Implement constraints and indexing to maintain data integrity and improve performance	primary key, foreign key, unique, and check constraints; indexes to optimize query speed; trade-offs between indexing and update overhead; constraints to enforce business rules; data consistency and accuracy; monitor performance impacts; apply security controls to restrict data access; clustered vs non-clustered indexes; constraint and index use; implementations with best practices; constraint effectiveness test; index of data growth update.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
	2.3 Evaluate and optimizes database design for efficiency and scalability	schema for potential bottlenecks; denormalization selectively to improve performance; partitioning and sharding strategies; queries via schema adjustments; data types and storage options review; impact of indexing on large datasets assessment; concurrency and transaction management plan; backup and recovery considerations integration; design supports high availability; evaluation findings documentation; security and compliance needs incorporation; design iteratively updates.
LO 3 Be able to implement SQL queries to manipulate and retrieve data effectively.	3.1 Write accurate SQL queries to retrieve data from single and multiple tables	SELECT statements with appropriate columns and aliases; WHERE clauses for filtering results; JOIN operations (INNER, LEFT, RIGHT, FULL) to combine tables; GROUP BY and HAVING for aggregation; subqueries for complex retrievals; NULL values effectively; results using ORDER BY; DISTINCT to eliminate duplicates; set operations (UNION, INTERSECT); queries for performance; queries against sample data; query logic document
	3.2 Create and modify data using SQL DML commands	NSERT statements to add records; UPDATE to modify existing data accurately; DELETE safely; transactions to ensure data integrity; rollback and commit operations; constraints during data manipulation; bulk data operations efficiently; data changes with SELECT queries; best practices to avoid data corruption; optimizes DML operations; data manipulation steps; tests changes in controlled environments.
	3.3 Use SQL functions and operators for data processing	Aggregate functions (COUNT, SUM, AVG, MIN, MAX); string, date, and numeric functions for data transformation; CASE statements for conditional logic; arithmetic and comparison operators effectively; NULL handling in expressions; functions for complex calculations; optimizes query readability and maintainability; function usage; functions with edge cases; compatibility with SQL standards; functions into data retrieval and reporting.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 4 Be able to learn how to optimize database performance, ensuring data integrity and security.	4.1 Analyze database performance using monitoring and profiling tools	Tools to monitor query execution times and resource usage; slow-running queries and bottlenecks; indexing effectiveness; transaction logs and locks; concurrency and deadlocks; execution plans; storage utilization assessment; findings and recommends improvements documentation; performance tuning strategies; testing optimizations; Caching where appropriate; minimal impact on availability.
	4.2 Implementation of data integrity and security measures	Constraints (primary key, foreign key, unique, check) to enforce data accuracy; transactions to maintain consistency; user roles and permissions; encryption for sensitive data; backup and recovery strategies; audit trails and logging; with data protection regulations; input validation; monitors security breaches; security policies; tests data integrity after modifications; security configurations.
	4.3 Optimize database design and queries for scalability and efficiency	Schema design to reduce redundancy and improve access speed; selectively de-normalize for performance gains; Partitioning and indexing strategies; Queries to reduce complexity; balance read/write loads; plan for horizontal and vertical scaling; test system under load; evaluate hardware and software resource utilization; integrate automated maintenance tasks; document optimization steps; incorporates feedback for continuous improvement; ensure alignment with business needs.

Essential Reading and Resource List

Textbooks:

Coronel, C. and Morris, S. (2019) *Database Systems: Design, Implementation, and Management.* 13th edition. Boston: Cengage Learning.
Harrington, J. L. (2016) *Relational Database Design and Implementation.* 4th edition. Burlington: Morgan Kaufmann.
Connolly, T. and Begg, C. (2015) *Database Systems: A Practical Approach to Design, Implementation, and Management.* 6th edition. Harlow: Pearson Education.

Journals & Newspapers:

ACM TODS
Journal of Database Management
Expert Systems with Applications
Grafiati's curated references
Database Systems Journal
Int. Journal of Information & Knowledge Management

Websites:

https://www.w3schools.com/sql
https://www.oracle.com/database/technologies

Web Design and Development

Unit Title	Web Design and Development
Unit Code	4APC5
Unit Level	Level 4
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module introduces web design and development principles. Students learn to create responsive, user-friendly websites using HTML, CSS, and JavaScript, focusing on layout, styling, interactivity, and best practices for modern web applications.

Unit Objectives

- To apply HTML to structure and present web content effectively.
- To use CSS to style and enhance the visual appearance of websites.
- To implement JavaScript to create interactive and dynamic web features.
- To apply web development principles to design responsive, user-friendly websites.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able understand the structure and semantics of HTML for creating accessible, standards-compliant web pages.	1.1 Explain the basic structure of an HTML document	Elements like doctype, html, head, and body; the role of metadata and title tags; how elements nest within the DOM tree; block-level vs inline elements; importance of document hierarchy; semantic tags for meaningful content organization; impact of well-formed HTML browser rendering; best practices for clean markup; distinguish between deprecated and modern tags; Relationship between HTML, CSS, and JavaScript; accessibility principles in markup; HTML structure clearly.
	1.2 Apply semantic HTML tags correctly for content meaning and accessibility	Headings (h1–h6) to structure content logically; Sectioning elements (article, section, nav, aside); List implementation (ordered, unordered, definition); Media tags (img, video, audio) with appropriate attributes; form elements with labels and field sets; accessibility attributes (aria-label, role); using non-semantic divs and spans unnecessarily; importance of alt text and captions; screen readers and assistive technologies; SEO benefits through semantic markup; validates HTML against standards; cross-browser compatibility test
	1.3 Create accessible and standards-compliant HTML pages	HTML code passes W3C validation; character encoding declarations (UTF-8); applies language attributes for localization; incorporation of accessibility best practices (keyboard navigation, contrast); content for readability and usability; responsive design principles; external resources correctly (scripts, stylesheets); inline styles and deprecated elements; tests with accessibility tools and validators; code to support maintenance; legal and ethical accessibility requirements; continuously updates skills with evolving web standards.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 2 Be able to apply CSS to design and style responsive, visually appealing, and user-friendly websites.	2.1 Use CSS syntax, selectors, and properties effectively	basic syntax rules for clean, readable stylesheets; element, class, ID, and attribute selectors correctly; text, color, and background properties for design consistency; cascade and specificity for overriding styles; the box model (margin, padding, border) appropriately; applies positioning and layout properties (float, flexbox, grid) effectively; units and measurement (px, %, rem, em) intentionally; separates structure (HTML) from presentation (CSS) clearly; naming conventions like BEM for maintainability; best practices for DRY (Don't Repeat Yourself) styling; validates styles for cross-browser compatibility.
	2.2 Implement responsive design techniques for varying screen sizes	Media queries for mobile, tablet, and desktop breakpoints; fluid layout techniques with flexbox and grid; relative units for text and elements (em, rem, %); images and media adaption with max-width settings; mobile-first design approaches for scalability; incorporation of viewport meta tags for mobile optimization; CSS frameworks or custom code for responsiveness; prioritize legibility and usability across devices; testing across multiple screen sizes and platforms; performance and load efficiency for mobile connections; progressive enhancement for older or limited browsers; documentation of responsive design decisions clearly.
	2.3 Enhance user experience and accessibility through advanced CSS	State-based styles (hover, focus, active) for interactive elements; animations and transitions for subtle, user-friendly feedback; Color contrast and readable font sizes for accessibility; Custom properties (CSS variables) for maintainable design themes; flexbox and grid for complex layout scenarios; Compatibility across browsers and devices; Validates styles for accessibility standards (WCAG 2.1);

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to use JavaScript to implement interactive and dynamic website features.	3.1 Use JavaScript for DOM manipulation and user interaction	Selects and edits elements using methods like get ElementById and querySelector; adds events for clicks, hover, and form submissions for responsiveness; applying dynamic content insertion, updating, and removal effectively; separates behavior from structure for maintainability and clarity; ensuring compatibility across browsers and screen sizes for a seamless experience.
	3.2 Implement asynchronous behavior and AJAX requests	Fetch, XMLHttp Request, and async/await for data retrieval and updating; handling JSON data, REST APIs, and status code responses reliably; error handling and fallback strategies for resilience; refreshing page sections in real-time for dynamic user experience; security and performance when managing asynchronous data flows.
	3.3 Integrate JavaScript libraries and frameworks effectively	Leveraging popular libraries (jQuery, Axios) and frameworks (React, Angular) for enhanced interactivity; component-based design and state management for maintainability; building tools (Webpack, npm) for modularization and optimization; cross-browser, device, and accessibility support for a wider audience; documentation of usage patterns clearly for collaboration and future scalability.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 4 Be able to apply best practices for website accessibility, performance, and deployment.	4.1 Evaluate website accessibility and applies best practices	Checking semantic HTML structure; using ARIA roles and landmarks; ensuring screen reader compatibility; providing alt text for images; maintains sufficient color contrast; implementing keyboard navigation support; focus indicators and tab order testing; follows WCAG guidelines; captions for media; content flashing; responsive layouts for all devices; promoting inclusive design principles.
	4.2 Optimize website performance for speed and efficiency	Page load times and critical rendering path; implementation of caching strategies and HTTP compression; minifying CSS, JavaScript, and HTML; optimize image formats and sizes; applies lazy loading for offscreen content; using asynchronous script loading and code splitting; performance monitoring via Lighthouse and WebPageTest; elimination of render-blocking resources; optimization of font loading; ensures mobile-friendly responsiveness; reduction of server requests; third-party script impact management.
	4.3 Prepare and manages website deployment effectively	Setting up simple deployment steps using Git and hosting platforms; organizing team roles for deployment tasks; website on staging or local servers before launch testing; HTTPS and strong passwords security; site uptime monitoring and fixing common issues; backups of website files and databases; working together to document deployment steps; updates and troubleshooting preparation; free or low-cost hosting services; best practices for reliable website delivery.

Essential Reading and Resource List

Textbooks:

Duckett, J. (2011) *HTML and CSS: Design and Build Websites*. Indianapolis: Wiley.
Freeman, E. and Robson, E. (2020) *Head First HTML and CSS*. 2nd edition. Sebastopol: O'Reilly Media.
Keith, J. and Andrew, R. (2010) *DOM Scripting: Web Design with JavaScript and the Document Object Model*. 2nd edition. Berkeley: Apress.

Journals & Newspapers:

Journal of Web Development and Web Designing, IJESE article (2024)
IJRPR article on responsive layout techniques (2024)
Webology
ACM TOIT
World Wide Web
"First Coding Steps" program report

Websites:

https://developer.mozilla.org
https://www.w3schools.com

Mathematics for Computing and Algorithmic Thinking

Unit Title	Mathematics for Computing and Algorithmic Thinking
Unit Code	4APC6
Unit Level	Level 4
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module develops mathematical skills and algorithmic thinking for computing. Students learn to apply mathematical concepts, logic, and structured problem-solving to design efficient algorithms and support effective software and system development.

Unit Objectives

- To apply principles of logic and number systems to computing problems.
- To analyse and use sets and relations in mathematical modelling.
- To explain the mathematical reasoning behind common data structures and algorithms.
- To develop algorithmic solutions using structured, logical, and mathematical approaches.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to apply core mathematical concepts relevant to programming and computational logic.	1.1. Understanding of number systems and base conversions	Binary, decimal, octal, and hexadecimal conversions; signed vs unsigned values; bit-level representation; use of two's complement; base conversion techniques; numeric overflow concepts; application in memory and addressing systems.
	1.2 Apply Boolean logic and logical operations in computing problems	AND, OR, NOT, XOR operators; truth tables and logic circuits; Boolean expressions in control flow (if/else, switches); simplification using Boolean algebra; use in decision-making and search conditions; mapping logic gates to program behavior.
	1.3 Solve problems using sets, relations, and basic algebra	Union, intersection, difference, and Cartesian product; Venn diagram interpretation; binary relations and equivalence; data grouping and filtering using sets; solving simple equations; relationships in databases and programming structures.
LO 2 Be able to use mathematical reasoning and problem-solving techniques in algorithmic contexts.	2.1 Apply pattern recognition and generalization in computing tasks	Arithmetic and geometric progressions; models logic using patterns; extrapolating values in loops; forming recurrence relations; detecting loops and nested patterns; applying modular arithmetic in problem-solving.
	2.2 Demonstrate use of flowcharts and pseudocode to model logic	Representing conditional and looping structures visually; converting scenarios into pseudocode; defining input/output clearly; applying modular design using functions; identifying syntax-agnostic logic representation; bridging logic to code transition.
	2.3 Use dry runs and trace tables to validate logic and control flow	Algorithm execution step-by-step simulation; tracing variable states and changes; logic errors and edge cases; validation of input/output at each iteration; explanation of impact of input variation on logic; uses tracing to debug and refine.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to understand and apply fundamental data structures in solving problems.	3.1 Describe the purpose and structure of basic data structures	Arrays, stacks, queues, linked lists, trees; outlining use cases and access patterns; comparison static vs dynamic storage; explanation of linear vs non-linear forms; showing practical application in program design and memory use.
	3.2 Implement simple data structures using programming logic	Building arrays with indexing; simulating stack push/pop, queue enqueue/dequeue; pointers or references; controlling flow using loops and conditions; tracking size and capacity; insert/delete operations safely handling.
	3.3 Analyzing time and space complexity of basic operations	Big O notation; complexity of insert/search/sort comparison; relating complexity to input size;evaluation of trade-offs in structure choice; identification of bottlenecks in memory use; prediction of scaling behavior under increasing data.
LO 4 Be able to evaluate basic algorithms for computation and data processing.	4.1 Explain sorting and searching algorithms	Steps of bubble, selection, insertion sort; implementation linear and binary search; evaluation of performance on ordered/unordered data; algorithm behavior with sample inputs comparison; stability and efficiency discussion.
	4.2 Design algorithms using iteration and recursion	Iterative solutions with for/while loops; logic with base and recursive cases; models factorial, Fibonacci, tree traversal; contrasting recursion vs iteration; manages call stack and termination logic.
	4.3 Evaluate algorithm correctness and efficiency using test cases	test cases to validate edge, boundary, and average cases; detection of logical and runtime errors; measures execution time and memory; comparison of algorithm outputs with expected results; documentation of reasoning for test design.

Essential Reading and Resource List

Textbooks:

Rosen, K. H. (2018) *Discrete Mathematics and Its Applications*. 8th edition. New York: McGraw-Hill Education.
Goodrich, M. T., Tamassia, R. and Goldwasser, M. H. (2013) D*ata Structures and Algorithms in Java.* 6th edition. Hoboken: Wiley.
Skiena, S.S. (2020) *The Algorithm Design Manual* (3rd Edition). London: Springer.

Journals & Newspapers:

Mathematics in Computer Science
Journal of Logic and Computation
Journal of Symbolic Computation
Mathematical Structures in Computer Science

Websites:

https://www.brilliant.org
https://www.khanacademy.org

Unit Specification Level 05

Enterprise Application Development

Unit Title	Enterprise Application Development
Unit Code	5APC1
Unit Level	Level 5
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module focuses on developing robust enterprise applications. Students learn advanced programming, software architecture, and integration techniques to design, build, and deploy scalable, secure, and maintainable business solutions in real-world environments.

Unit Objectives

- To develop enterprise-level applications using Java, .NET, or Node.js frameworks.
- To apply software architecture and design patterns for scalable and maintainable solutions.
- To integrate databases, APIs, and external services within enterprise applications.
- To implement security, testing, and deployment best practices for robust applications.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to apply object-oriented programming (OOP) concepts to design robust, maintainable software.	1.1 Explain core OOP concepts and principles	Defining classes, objects, inheritance, encapsulation, and polymorphism; distinguishing abstraction and implementation; access modifiers and constructors; method overloading and overriding; design principles like SOLID; relates concepts to software maintainability and reuse; uses UML diagrams for design visualization.
	1.2 Design software solutions using OOP methodologies	Class hierarchies and relationships; encapsulation to protect data; inheritance to promote code reuse; polymorphism for flexible behavior; interfaces and abstract classes appropriately; real- world problems into objects and interactions; design with diagrams and comments.
	1.3 Implement OOP solutions using languages like Java, .Net, or Node.js	Clean, modular, and reusable code; applies constructors, getters, and setters correctly; exception handling and error management; integrates APIs and libraries with OOP principles; unit tests for classes and methods; follows coding standards and best practices; debugs and refactors code for clarity and efficiency.
LO 2 Be able to develop and integrate applications using Java and .NET platforms following best practices.	2.1 Develop applications using Java and .NET frameworks	Code leveraging Java and .NET syntax and libraries; core features like classes, methods, and exception handling; platform-specific tools and IDEs efficiently; language conventions and design patterns; debugging and testing techniques; integrates external libraries and APIs; manages project structure and dependencies.
	2.2 Integrate components and services within applications	Connecting modules through APIs, web services, or middleware; applies RESTful and SOAP communication where appropriate; management of data persistence with databases or ORM tools; authentication and authorization securely; asynchronous programming models; front-end and back-end interaction; integration processes document
	2.3 Follow best practices in application development and maintenance	Coding standards and naming conventions; version control and collaboration tools; modular and maintainable code; automated testing and continuous integration; optimization of performance and resource usage; ensuring scalability and extensibility; maintenance ofcomprehensive documentation.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to build and consume RESTful APIs using Node.js and other relevant technologies.	3.1 Design RESTful APIs following standard principles	Endpoints using HTTP methods (GET, POST, PUT, DELETE); REST architecture constraints; uses meaningful resource URIs; request and response formats (JSON, XML); management of status codes and error messages; documents API specifications clearly; stateless interactions; versioning strategies.
	3.2 Implement RESTful APIs using Node.js and related tools	Using express or similar frameworks to build APIs; routing set up middleware, and controllers; connecting APIs to databases for CRUD operations; authentication and authorization (JWT, OAuth); validation and sanitizes user input; handling errors and exceptions gracefully; writes unit and integration tests for endpoints.
	3.3 Consume RESTful APIs effectively in applications	HTTP requests using fetch, Axios, or other libraries; asynchronous operations with promises and async/await; processes API responses and errors properly; integration of APIs into frontend or backend workflows; management of API rate limits and retries; documentation of API usage and limitations; testing API consumption under different scenarios.
LO 4 Be able to evaluate, test, and optimize software for performance, security, and scalability.	4.1 Evaluate software performance and identifies bottlenecks	Profiling and monitoring tools to measure CPU, memory, and network usage; algorithm efficiency and resource consumption; slow or inefficient code sections; benchmarking software under different loads; documenting performance findings clearly; suggesting optimization strategies; application of caching and lazy loading techniques.
	4.2 Test software for functionality, security, and reliability	Design and execution unit, integration, and system tests; security testing for vulnerabilities (XSS, SQL injection, CSRF); automated testing tools and frameworks; validation input and handles exceptions securely; code reviews and static analysis; documentation of test cases and results comprehensively; ensuring software meets requirements.
	4.3 Optimizes software for scalability and maintainability	Implements design patterns and modular architecture; refactors code to reduce complexity and improve readability; manages concurrency and parallelism effectively; applies load balancing and distributed computing concepts; monitors system under stress and scales resources as needed; maintains clear documentation; plans for future feature expansion.

Essential Reading and Resource List

Textbooks:

Horstmann, C. S. (2020) *Core Java Volume I – Fundamentals.* 12th edition. Boston: Pearson.
Troelsen, A. and Japikse, P. (2021) *Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming*. 10th edition. New York: Apress.
Valerio, D. (2021) *REST API Development with Node.js.* Birmingham: Packt Publishing.

Journals & Newspapers:

IEEE TSE, ACM TSE/M
IEEE Software
Journal of Systems and Software, Innovations in Systems & Software Engineering
Empirical Software Engineering
European Journal of Information Systems, Integration Journal

Websites:

https://learn.microsoft.com/dotnet
https://nodejs.org/en/docs

Advanced Database Design

Unit Title	Advanced Database Design
Unit Code	5APC2
Unit Level	Level 5
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module explores advanced database design concepts. Students learn to optimize database performance, implement complex queries, manage large-scale data, and apply emerging technologies for efficient, secure, and scalable data management solutions.

Unit Objectives

- To design complex database models using enhanced ER diagrams.
- To apply normalisation techniques to optimise database structure and eliminate redundancy.
- To develop advanced SQL queries for efficient data manipulation and retrieval.
- To implement and manage databases using Relational Database Management Systems (RDBMS) effectively.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able model complex data requirements using advanced ER and EER diagramming techniques.	1.1 Understand advanced ER and EER concepts and components	Entities, relationships, and attributes; distinguish strong and weak entities; specialization, generalization and aggregation; cardinality and participation constraints; incorporation of inheritance and subtype/supertype hierarchies; representation of complex relationships with proper notation; constraints and business rules modeling.
	1.2 Create detailed ER and EER diagrams for complex data models	Diagrams reflecting real-world scenarios accurately; Crow's foot and Chen notations appropriately; multi-valued and derived attributes modeling; entity hierarchies and polymorphism; documentation of assumptions and design decisions; validation of diagrams with stakeholders for accuracy.
	1.3 Analyzes data models for completeness and correctness	Redundancy and inconsistencies; Normalization principles to improve design; resolving many- to-many relationships; integrity constraints are enforced; diagram review for clarity and completeness; incorporation of feedback from domain experts; preparation of diagrams for database implementation.
LO 2 Be able to apply normalization theory to design optimized, resilient, and maintainable relational database structures.	2.1 Understand normalization concepts and normal forms	First (1NF), second (2NF), and third normal forms (3NF); Boyce-Codd Normal Form (BCNF) and higher normal forms; functional dependencies and keys; anomalies caused by poor design; relates normalization to data integrity; redundancy reduction.
	2.2 Applies normalization to database schemas effectively	Transformation of unnormalized schemas into 1NF, 2NF, and 3NF; elimination partial and transitive dependencies; decomposing relations without loss of data; balancing normalization with practical performance needs; documentation of normalization steps clearly; using case studies to validate designs.
	2.3 Designs resilient and maintainable relational structures	Schema supports data consistency and integrity; avoidance of update, insertion, and deletion anomalies; creation of well-defined primary and foreign keys; anticipation of future scalability and modifications; integration of normalization principles with indexing and query optimization; preparation of schemas for efficient implementation.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to develop complex queries using advanced SQL and implement database solutions within an RDBMS environment.	3.1 Write advanced SQL queries for complex data retrieval	Multi-table joins (INNER, LEFT, RIGHT, FULL); nested and correlated subqueries; window functions and ranking; grouping, aggregation, and filtering; set operations (UNION, INTERSECT, EXCEPT); efficient and readable queries; documentation of query logic clearly.
	3.2 Implement database modifications and transactions	INSERT, UPDATE, DELETE with conditions; transactions using COMMIT, ROLLBACK, and SAVEPOINT; data integrity with constraints and triggers; concurrency and locking mechanisms; stored procedures and functions for modularity; changes thoroughly; bestpractices.
	3.3 Deploy and maintains database solutions within RDBMS	Schemas optimization for query performance; configuration of indexes and views appropriately; monitoring and tuning query execution plans; backup and recovery strategies; security measures like roles and permissions; documentation of database design and operations; collaboration of for ongoing maintenance and updates.
LO 4: Be able to evaluate and implement performance optimization, data security, and transaction control in relational databases.	4.1 Analyzes database performance and identifies optimization opportunities	Monitoring of query execution times and resource usage; examining indexes and their effectiveness; query plans and profiling tools usage; bottlenecks and inefficient queries; recommendation of indexing, partitioning, and caching strategies; evaluation of normalization vs. denormalization trade-offs; performance tuning steps documentation.
	4.2 Implement data security measures within the database	Defining user roles and permissions; encryption for data at rest and in transit; management of authentication and access controls; auditing database activity and logs; enforcing data privacy regulations and compliance; prevention of SQL injection and other threats; documentation of security policies and procedures
	4.3 Control database transactions to ensure integrity and consistency	Using ACID properties to manage transactions; application of locking and isolation levels appropriately; handling deadlocks and concurrency issues; implementation of commit and rollback mechanisms; testing transaction behavior under concurrent access; maintenance of data consistency across multiple operations; documentation of transaction management strategies.

Essential Reading and Resource List

Textbooks:

Elmasri, R. and Navathe, S. B. (2016) Fundamentals of Database Systems. 7th edition. Boston: Pearson.
Date, C. J. (2019) An Introduction to Database Systems. 8th edition. Boston: Pearson.
Stephens, R. and Plew, R. (2021) Sams Teach Yourself SQL in 24 Hours. 7th edition. Indianapolis: Sams Publishing.

Journals & Newspapers

Journal of Database Management, Storey
IJDMS (AER diagrams)
genetic algorithm normalization
Teorey et al. methodology
ACM TODS
Data & Knowledge Engineering

Websites:

https://docs.oracle.com/en/database
https://www.postgresql.org/docs

Data Engineering and Big Data Technologies

Unit Title	Data Engineering and Big Data Technologies
Unit Code	5APC3
Unit Level	Level 5
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module introduces data engineering and big data technologies. Students learn to design, build, and manage large-scale data pipelines, leveraging tools and frameworks to process, store, and analyze massive datasets efficiently.

Unit Objectives

- To explain the architecture and use of HDFS for distributed data storage.
- To develop data processing solutions using Apache Spark for big data analytics.
- To analyse and apply NoSQL databases for handling unstructured and semi-structured data.
- To compare and differentiate between data lakes and data warehouses for efficient data management.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to understand the architecture and components of distributed data systems	1.1 Explain the core principles of distributed computing and Big Data	Defining distributed computing; explanation of scalability and fault tolerance; outlining key challenges in large-scale data processing; distinguishing between centralized and distributed architectures; introduction of Big Data characteristics (volume, velocity, variety); relates to industry examples
	1.2 Describe the structure and function of HDFS and related components	Explanation of Name Node, Data Node, replication, and block storage; outlines data redundancy mechanisms; compares with traditional file systems; describes data write/read process in HDFS; discusses fault recovery in HDFS; highlights limitations of HDFS.
	1.3 Identify components of modern data engineering pipelines	Introduction of batch vs streaming pipelines; outlining key tools (e.g., HDFS, Spark, Kafka); description of stages: ingestion, processing, storage, visualization; connects pipeline components to real-world applications; modularity and scalability emphasis
LO 2 Be able to apply data processing techniques using Apache Spark and related tools	2.1 Implements data transformations using Spark Core and Spark SQL	RDDs and Data Frames for data manipulation; filtering, grouping, and aggregation; writing SQL-like queries with Spark SQL; understanding lazy evaluation and DAG; optimizes execution with caching and partitioning; handles structured and semi-structured data.
	2.2 Performs data processing tasks for batch and real-time analytics	Configuring Spark jobs; Spark Streaming or Structured Streaming; ingesting data from multiple sources (CSV, Kafka, JSON); management of Spark sessions; evaluation of throughput and latency; handling failures and stateful processing.
	2.3 Analyzes and improves Spark job performance	Monitoring job execution via Spark UI; identification of bottlenecks in shuffling and serialization; application of partition tuning and broadcast joins; optimization of memory usage; configuration tuning techniques; comparison of performance across datasets and job types.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to evaluate NoSQL databases for flexible, high-volume data storage	3.1 Explain NoSQL models and their use cases	key-value, document, column, and graph models; compares schema flexibility vs RDBMS; relating NoSQL models to applications (IoT, social media); evaluation of data consistency and availability needs; alignment of models with business requirements.
	3.2 Perform basic operations in a NoSQL database	Creation and queries collections or tables in MongoDB/Cassandra; insert, update, and delete operations; filtering data using query syntax; management of indexes and keys; handling nested or unstructured data; interpreting query results.
	3.3 Evaluation NoSQL database scalability, availability, and trade-offs	Explanation of replication and sharding strategies; application of CAP theorem to real-world scenarios; assessment of horizontal scaling features; comparison of performance under different workloads; identification of trade-offs between consistency and partition tolerance; deployment architecture.
LO 4 Be able to compare data lakes and data warehouses, and their role in modern analytics	4.1 Describe the characteristics and architecture of data lakes and warehouses	Explanation of differences in schema enforcement (schema-on-read vs write); comparison of storage types and formats; outlining stages of ingestion and querying; defining data zones in a data lake (raw, curated, trusted); describes warehouse schema and structure.
	4.2 Compare use cases and workloads across lakes and warehouses	Identification of suitable tasks for each system (BI, ML, ETL); analysis of cost and performance considerations; alignment of systems to structured vs unstructured data; evaluation of user personas (analyst, engineer, scientist); provides real-world examples of each model.
	4.3 Identify tools and platforms supporting modern data architectures	Lists cloud services (AWS S3, Redshift, GCP BigQuery, Azure Synapse); explanation of ETL/ELT pipeline design using modern tools; introduces concepts like lake house architecture; evaluation of integration of storage, compute, and analytics tools; mapping platforms to project needs.

Essential Reading and Resource List

Textbooks:

Karau, H., Warren, R., and Zaharia, M. (2015) *Learning Spark: Lightning-Fast Big Data Analysis.* 1st edition. Sebastopol: O'Reilly Media.
White, T. (2015) *Hadoop: The Definitive Guide.* 4th edition. Sebastopol: O'Reilly Media.
Guller, M. (2015) *Big Data Analytics with Spark: A Practitioner's Guide to Using Spark for Large Scale Data Analysis.* Berkeley: Apress.

Journals & Newspapers:

Journal of Big Data
Data & Knowledge Engineering
International Journal of Data Warehousing and Mining
Data Lakes Survey
Data Technologies and Applications

Websites:

https://spark.apache.org/docs
https://www.databricks.com/learn

Cloud Computing

Unit Title	Cloud Computing
Unit Code	5APC4
Unit Level	Level 5
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module explores cloud computing concepts, architectures, and services. Students learn to design, deploy, and manage scalable, flexible cloud-based solutions using major platforms, emphasizing security, virtualization, and resource optimization.

Unit Objectives

- To understand the fundamentals of cloud computing and its evolution from traditional models.
- To analyse the benefits and challenges associated with adopting cloud technologies.
- To explore major cloud platforms and their services for scalable application deployment.
- To apply basic AI and ML concepts within cloud environments to enhance computing solutions.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO1 Be able to understand concepts of cloud computing	1.1 Explains fundamental cloud computing concepts	Cloud computing and its key characteristics; service models (IaaS, PaaS, SaaS); deployment models (public, private, hybrid); scalability, elasticity, and on-demand resources; benefits like cost-efficiency and accessibility; cloud to IT modernization relationship.
	1.2 Compare of cloud computing with traditional IT infrastructure	Cloud services; on-premises hardware; differences in management, maintenance, and cost; flexibility and resource provisioning; scalability and disaster recovery options; security and compliance challenges; analysis of use cases for each approach.
	1.3 Describe cloud computing components and architecture	Cloud data centers, virtualization, and multi-tenancy; networking and storage in cloud; APIs and management tools; service orchestration and automation; common cloud platforms; architecture to performance and reliability.
LO 2 Be able to analyze the benefits and challenges associated with cloud adoption.	2.1 Explain key benefits of cloud adoption	Cost savings and reduced capital expenditure; scalability and resource elasticity; improved accessibility and collaboration; faster deployment and innovation; automatic updates and maintenance; benefits for business agility and competitiveness.
	2.2 Analise challenges in cloud adoption	Security risks and data privacy concerns; compliance and regulatory issues; dependency on internet connectivity; potential vendor lock-in; migration complexity and legacy system integration; performance variability and downtime risks.
	2.3 Analyzes strategies to manage cloud adoption challenges	Security best practices and encryption; multi-cloud and hybrid strategies; disaster recovery and backup plans; phased migration and testing; service-level agreements (SLAs); monitoring cloud performance and costs; risk mitigation approaches documentation.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to identify and use core cloud service models and deployment models	3.1 Describe cloud service models	Describes Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS); explains their characteristics, benefits, and trade-offs; provides industry examples (e.g., AWS EC2, Google App Engine, Microsoft 365)
	3.2 Explain cloud deployment models	Explains public, private, hybrid, and community cloud deployment models; compares advantages and limitations of each; relates to scenarios such as startups, government systems, or enterprise IT
	3.3 Match real-world use cases to service and deployment models	Matches given business scenarios to the most suitable cloud service and deployment models; justifies the choice with consideration for cost, scalability, security, and operational needs
LO 4 Be able to evaluate leading cloud platforms and their core services	4.1 Identify leading cloud platforms and their market position	Major providers like AWS, Microsoft Azure, Google Cloud Platform; comparing platform popularity and adoption; summarizing core offerings and specialties; reviewing service availability and global reach; assessing vendor reputation and support; following industry trends and innovations.
	4.2 Describe core services offered by major cloud platforms	Compute services (VMs, containers, serverless); outlining storage options (object, block, file storage); detailing networking services (VPC, CDN, load balancing); introduction database and analytics offerings; covering security and identity management; reviews of monitoring and management tools.
	4.3 Evaluate cloud platforms based on features, costs, and usability	Pricing models and cost management tools; ease of deployment and integration; evaluates scalability, reliability, and performance; security features and compliance certifications; learning resources and community support; evaluation findings for informed decisions.

Essential Reading and Resource List

Textbooks:

Velte, T., Velte, A., and Elsenpeter, R. (2010) *Cloud Computing: A Practical Approach*. New York: McGraw-Hill Education.
Rittinghouse, J. W. and Ransome, J. F. (2017) *Cloud Computing: Implementation, Management, and Security*. 2nd edition. Boca Raton: CRC Press.
Bahga, A. and Madisetti, V. (2013) *Cloud Computing: A Hands-On Approach.* Fairfax: University Press.

Journals & Newspapers:

Journal of Cloud Computing: Advances, Systems and Applications
Future Generation Computer Systems
International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Cloud Computing Benefits" by Ayob Sether
The Future of Cloud Computing: Benefits and Challenges" by R. Islam

Websites:

https://aws.amazon.com
https://learn.microsoft.com/en-us/training/azure

Introduction to AI/ML

Unit Title	Introduction to AI/ML
Unit Code	5APC5
Unit Level	Level 5
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This module introduces artificial intelligence and machine learning fundamentals. Students explore algorithms, data processing, and model training to develop intelligent systems, focusing on practical applications and ethical considerations in modern AI/ML technologies.

Unit Objectives

- To understand core concepts and techniques of artificial intelligence and machine learning.
- To implement basic machine learning algorithms for data analysis and prediction.
- To process and prepare data for training AI/ML models effectively.
- To evaluate ethical considerations and real-world applications of AI and machine learning technologies.

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

On completion assessment of this module the learner should:

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to understand the concepts and history of AI	1.1 Define Artificial Intelligence	Defines Artificial Intelligence; explains how AI enables machines to perform tasks that typically require human intelligence; outlines capabilities such as perception, reasoning, and decision-making; provides everyday examples (e.g., virtual assistants, recommendation systems, autonomous vehicles)
	1.2 Describe key milestones in AI development	Describes key milestones in AI; early symbolic AI, expert systems, and the rise of machine learning; landmark events such as the Turing Test, IBM Deep Blue's chess victory, and breakthroughs in deep learning; relates developments to current AI capabilities.
	1.3 Identify main branches of AI and their uses	Main branches of AI including computer vision, natural language processing, robotics, and expert systems; purpose of each branch; relevant industry use cases (e.g., facial recognition in security, chatbots in customer service, robotic process automation in manufacturing).
LO 2 Be able to describe core machine learning concepts	2.1 Explain how machine learning differs from traditional programming	Difference between traditional programming (rule-based) and machine learning (data-driven); outlines how models learn patterns from data; examples such as email spam detection and predictive text.
	2.2 Describe types of machine learning and their applications	Three main types of machine learning - supervised, unsupervised, and reinforcement learning; explain their principles, data requirements, and example applications such as price prediction, customer segmentation, game-playing AI).
	2.3 Define key machine learning terminology	Essential ML terms such as dataset, features, labels, training, testing, and accuracy; role of each term in the model development process; Example, predicting student grades based on study hours.

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 3 Be able to recognize common AI/ML tools, platforms, and applications	3.1 Identify common AI/ML frameworks and their uses	AI/ML frameworks; TensorFlow, PyTorch, and Scikit-learn; primary functions in model development and deployment; tools to example projects (e.g., image classification, sentiment analysis).
	3.2 Describe cloud-based AI/ML services	Cloud-based AI/ML services and their advantages; how services such as AWS SageMaker, Azure Machine Learning, and Google AI Platform provide infrastructure, scalability and pre- built models; relates to rapid prototyping scenarios.
	3.3 Give examples of AI/ML applications in industries	AI/ML applications in industries including healthcare (diagnostic imaging), finance (fraud detection), and retail (product recommendations); how AI adds value in each case.
LO 4 Be able to understand ethical considerations and future trends in AI/ML	4.1 Identify risks and challenges in AI	common risks in AI such as bias, privacy breaches, and over-reliance on automation; potential societal impact of each risk with real-world examples.
	4.2 Explain ethical principles for AI	Importance of ethical AI practices; outlines principles such as fairness, accountability, transparency, and privacy; provides examples of how ethical guidelines influence AI system design.
	4.3 Describe emerging trends in AI/ML	Emerging AI/ML trends including generative AI, AI-powered automation, and AI for climate change; explain their potential benefits and challenges; relates trends to future workforce needs.

Essential Reading and Resource List

Textbooks:

Stuart J. Russell and Peter Norvig (2020) *Artificial Intelligence: A Modern Approach,* Pearson
Aurélien Géron, (2019), *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow,* O'Reilly Media
Andrew Ng, *Machine Learning Yearning" by Andrew Ng,*
Christopher M. Bishop. (2006) *Pattern Recognition and Machine Learning*, Springer

Journals & Newspapers:

Journal of Machine Learning Research (JMLR)Future Generation Computer Systems
IEEE Transactions on Neural Networks and Learning Systems
Artificial Intelligence

Websites:

https://www.coursera.org/learn/machine-learning
https://www.kaggle.com/
https://scikit-learn.org/stable/
https://www.deeplearning.ai/

Industry Investigation and Employement

Unit Title	Industry Investigation and Employement
Unit Code	5APC6
Unit Level	Level 5
Credit Value	20
Guided Learning Hours	100
Independent Learning Hours	100
Unit Type	Core
Grading System	Pass / Fail

Unit Introduction

This unit intends for the learners to understand and develop skill sets to secure a job in the industry.

Unit Objectives

- To study industry Trend and labor market
- To assess the competency needs
- To prepare CVs and for interviews
- To secure a job

Assessment

Learning Outcomes to meet	All 4 Learning Outcomes
Assessment Criteria to cover	All ACs of each Learning Outcome
Assessment method	Coursework, Practical or other appropriate methods
Word count	3000 or Equivalent

Assignment briefs are part of this unit specification and available to all the centres and learners.
Centre assess all the units internally and B-TIC externally verify for quality assurance

Learning Outcomes	Assessment Criteria	Unit Teaching Content
LO 1 Be able to assess personal competency needs within the context of employment	1.1 Industry Analysis for employment	Industry Overview: What does the industry do? Key players; Market size and growth trends; Employment Statistics; Number of people employed; Growth/decline in employment; Major job roles. Key Drivers of Employment: Technological advancements, consumer demand, government policies, and economic factors. Urbanisation: Growing cities drive demand for housing and infrastructure. Government Infrastructure Spending: Stimulus packages and infrastructure bills. Technological Integration: Skills in Demand; Technical and soft skills; Certification or degree requirements; Trends & Challenges; Automation; Remote work; Regulatory issues; Labor shortages/surpluses; Outlook; Projected job growth; emerging roles; long-term stability.
	1.2 Job investigation	Job or career path; finding job openings; potential employer; job market analysis; Company culture and values; Salary and benefits; Work-life balance; Employee reviews; Opportunities for growth; Job stability and reputation; Responsibilities; Job Market & Demand; Salary Range; Qualifications; Membership of professional bodies; Career Growth Paths; Job Market Research & Trends: LinkedIn Talent Insights; Indeed Hiring Lab; Google Trends + Google Job Search API; Lightcast.
	1.3 Competency analyses	Core Competencies; Technical Competencies; Behavioral / Interpersonal Competencies; Managerial / Strategic Competencies.
LO 2 Be able to assess the competency needs for the industry	2.1 Review industry operations in order to identify personal development needs	Industry Trends; Objective; Read industry reports; Emerging technologies and practices. Industry Conferences, webinars, expert podcasts; Are there new tools or platforms becoming standard; growing emphasis on data; sustainability; automation; Benchmark Key Roles and Competencies; Compare your role to industry standards; Action Steps: Review job descriptions for similar roles in leading organisations. Examine LinkedIn profiles of professionals in similar positions. Competency frameworks

Learning Outcomes	Assessment Criteria	Unit Teaching Content
	2.2 Undertake a skills audit to define areas of personal learning needs	Learning styles: visual, aural, verbal, physical, logical, social, and solitary; Identifying own learning style. Continuous Professional Development (CPD): Training versus development; personal need versus employer need. Identifying personal needs: skills audit, future plans, employer needs, skills gaps, and company goals.
	2.3 Develop a personal development plan (PDP)	Setting goals: SMART goals (specific, measurable, attainable, relevant, time-bound); learning goals versus employment goals. Learning plan: goals, actions, resources. Personal Development Plan (PDP); Personal Vision Statement; Strengths; Areas for Improvement; Action Plan; Resources; Support Network; Mentor/Coach; Peers/Colleagues; Family/Friends; Communities/Groups; Review and Reflection; Next Review Date.
LO 3 Be able to manage social media for employment industry update	3.1 Select social media to find employment.	LinkedIn; X; Facebook; Instagram; TikTok; Professional networking; up-to-date profile; Using keywords from desired industry; following relevant groups, companies and recruiters; Engage with posts and share insights to stay visible; Open to Work. Hashtags: Share thoughts on industry trends to showcase expertise. Create short videos to share your resume, job search tips, or career journey; Consistency; Clean up your profiles; Engage

Learning Outcomes	Assessment Criteria	Unit Teaching Content
	3.2 Content management of social media	Planning, creating, scheduling, publishing, and analysing content across various social platforms. Strategy Development: Audience Research; Platform Selection; Goals & KPIs. Content Planning: Content Calendar; Content Mix; Content Creation: Design & Copywriting; Formats; Branding Scheduling & Publishing: Tools; Best Times to Post Community Management; Engagement; DMs; Moderation; Relationship Building; Analytics & Reporting; Track Metrics; Reports. Optimization: A/B Testing; Trend Adaptation Consistency Across Platforms
	3.3 Develop professional Network in social media	Choose the Right Platforms; Profile Optimization; Professional Photo; Summary/About Section. Professional background, skills, and goals; Share Valuable Content; Engage Consistently; Connect Strategically; Join industry-specific groups or hashtags; Tag people or companies; Attend virtual events or webinars; Follow industry leaders
LO 4 Be able to secure employment	4.1 Search an employment opportunities	Job search: Job title or field; Industry; Location; Remote, hybrid, or specific cities; Work type: ull-time, part-time, freelance, internships Online Job Boards: LinkedIn, Indeed, Glassdoor, Monster; Remote OK (for remote work); career websites Search using keywords; Set Up Alerts and Use Filters;

Learning Outcomes	Assessment Criteria	Unit Teaching Content
	4.2 Apply for a jobs	Covering Letter: Tailor it to the Job; Concise (1 Page Max, 3–4 paragraphs, Professional Language) Salutation; Show Enthusiasm and Personality; Focus on Achievements; Quantitative information; End with a Call to Action; Express interest in an interview; Thanking; Format Professionally; Application forms; online application Customize cover letter; Don't mass-apply; Track your applications (e.g., in a spreadsheet or with tools like Notion or Huntr). Network; reach out to people in your industry on LinkedIn; attend job fairs or online events. Be Consistent; Set a routine—apply for a few jobs every day CV template; writing or editing CV; Microsoft Word Templates; Minimalist CV by Hloom; Google Docs' built-in templates; Spearmint Template; Canva; Sure! Here are some solid CV writing tips to help you craft a standout resume, whether you're aiming for your first job or levelling up in your career. Key Principles: Tailor it to the Job; Concise; Clean Format. Word count Essential Sections: Contact Information; Personal Profile (Optional); Key Skills; Work Experience (reverse chronological order); Education; Certifications & Training; Use Action Verbs (managed, developed, increased, led, optimized); Quantify Achievements; Avoid Jargon Unless it's industry-specific. Check Spelling & Grammar; someone else to proofread; Save as PDF.
	4.3 Face an Iinterview	Research the company; Practice common interview questions, general questions, and Role-specific questions; STAR Method for Behavioral Questions; Prepare Questions for the Interviewer; Logistics and Appearance; Follow Up. Types of interview questions: General or Traditional Questions; Behavioral Questions; Situational or Hypothetical Questions; Technical or Skill-Based Questions; Case Study or Problem-Solving Questions; Cultural Fit Questions; Brainteasers or Puzzle Questions

Essential Reading and Resource List

Textbooks:

Bolton, G. (2014) *Reflective Practice Writing and Professional Development.* London: Sage Publications Ltd.
Chandler, I., and McCarthy, J. (2022) *Your graduate job search made easy.* 5th ed. Bath: Trotman.
Green, M. (2021) *Careers in construction: Building a sustainable future.* Abingdon: Routledge.
Lussier, R. N., and Hendon, J. R. (2021) *Human resource management: Functions, applications, and skill development.* 5th ed. Los Angeles: SAGE Publications.
Pritchard, A. (2006) *Ways of Learning: Learning Theories and Learning Styles in the Classroom.* Abingdon: Routledge.
Raelin, J.A. (2008) *Work-based Learning: Bridging Knowledge and Action in the Workplace*. London: Jossey-Bass.
Tarrant, P. (2013) *Reflective Practice and Professional Development.* London: Sage.

Journals & Newspapers:

Journal of Industrial Relations
Industrial and Labor Relations Review
European Journal of Industrial Relations
The Wall Street Journal (WSJ)
Harvard Business Review (HBR)

Websites:

https://www.ciob.org
https://www.goconstruct.org