Programme Handbook

For

Bachelor of Science in Applied Computing and Information Technology (BSc ACIT)

Royal Thimphu College

Royal University of Bhutan

July 2026

This programme handbook should be read in conjunction with the RTC Student Handbook.

Acknowledgements:

This programme was developed at and is copyrighted by the Royal Thimphu College. The Royal Thimphu College is permitted to offer this programme as an academic affiliate of the Royal University of Bhutan, and allows other colleges to offer the programme as per RUB guidelines.

Table of Contents

1. Programme Specification **1**
1.1. Basic Information on the Programme 1
1.2. Aims and Learning Outcomes of the Programme 1
1.3. Career-related Opportunities 3
1.4. Justification of the Level of Award and Title 3
1.5. Justification and Demand for the Programme 5
1.6. Programme Structure 7
1.7. Learning and Teaching Approach 10
1.8. Assessment approach 11
1.9. Regulations 12
1.10. Programme Management, Quality Assurance, and Enhancement 13
2. Module Descriptors **17**
Year 1 Semester I 17
Module Code and Title: CCS101 Digital Logic and Design 17
Module Code and Title: PGL101 Fundamentals of Programming 20
Module Code and Title: WEB101 Foundations of Web Development 23
Module Code and Title: MTH101 Discrete Mathematics 29
Module Code and Title: HCI101 Mindful Technology and Digital Wellbeing 33
Year 1 Semester II 39
Module Code and Title: CCS102 Operating Systems 39
Module Code and Title: DMS101 Database Management System 43
Module Code and Title: PGL102 Object-Oriented Programming 47
Module Code and Title: MTH102 Calculus 50
Module Code and Title: EAP101 Intermediate English for Academic Purposes 53
Year 2 Semester I 56
Module Code and Title: Computer Organisation and Architecture 56
Module Code and Title: DMS202 Database Programming 60
Module Code and Title: NET201 Networking Fundamentals 63
Module Code and Title: PGL203 Data Structures and Algorithms 67
Module Code and Title: EAP102 Upper Intermediate English for Academic Purposes 70
Year 2 Semester I 73
Module Code and Title: NET202 System Administration 73
Module Code and Title: SDP201 Systems Analysis and Design 77
Module Code and Title: WEB202 Dynamic Front-end Development 81
Module Code and Title: MTH203 Statistics and Probability 85
Module Code and Title: DZG101 རྫོང་ཁ་བཀའ་དྲིན་ལས་ཀྱི་སྡེབ་ཚན། 88
Year 3 Semester I 93
Module Code and Title: EMT301 Introduction to Artificial Intelligence & Machine Learning 93
Module Code and Title: PGL304 Mobile App Development 97
Module Code and Title: SDP302 Software Engineering 102
Module Code and Title: WEB303 Back-end Development 108

Module Code and Title: HCI302 Human Computer Interaction 111
Year 3 Semester II 115
Module Code and Title: CYS301 Introduction to Cyber Security 115
Module Code and Title: BIS301 Data Analytics and Business Intelligence 119
Module Code and Title: EMT302 Internet of Things (IoT) 124
Module Code and Title: RIT301 Research Methodology 129
Module Code and Title: EDP101 Entrepreneurship 136
Year 4 Semester I 145
Module Code and Title: EMT403 Blockchain Technology and Applications 145
Module Code and Title: BIS402 Enterprise Systems and Business Context 150
Module Code and Title: HCI403 IT Project Management 154
Module Code and Title: HCI404 IT Laws and AI Ethics 158
Module Code and Title: ITP401 Project 163
Year 4 Semester II 170
Module Code and Title: ITP402 ACIT Capstone Project 170

1. Programme Specification

1.1. Basic Information on the Programme

Name of the home base college:	Royal Thimphu College
Title of Award:	Bachelor of Science in Applied Computing and Information Technology
Duration and mode of study:	Four years, full-time
Award granting body:	Royal University of Bhutan
Date of Initial Approval:	**June 6 (**59th AB)

1.2. Aims and Learning Outcomes of the Programme

1.2.1. Aims of the Programme

The Bachelor of Science in Applied Computing and Information Technology (BSc ACIT) aspires to develop technologically skilled, ethically conscious, and socially responsible professionals who will play a pivotal role in Bhutan's digital transformation. More than imparting technical expertise, the programme seeks to cultivate leaders who embody resilience, mindfulness, and ethical computing, ensuring that innovation aligns with national priorities, cultural values, and global standards.

The programme will train the graduates in cutting-edge IT disciplines including Artificial Intelligence (AI) applications, cybersecurity, cloud computing, and Internet of Things (IoT) while simultaneously developing mindful leadership, interdisciplinary problem-solving, and ethical reasoning skills. Through experiential learning, capstone project, and structured mentorship, students will gain the ability to design digital solutions that enhance well-being, equity, and sustainable urban growth, particularly in alignment with Bhutan's Gelephu Mindfulness City (GMC) initiative.

By fostering adaptability, resilience, and ethical stewardship, the programme aims to equip students with the agility to thrive in evolving technology landscapes, thereby empowering them to drive innovation responsibly whether in smart city development, ethical AI platforms, cybersecurity resilience, or sustainable IT infrastructure. Ultimately, graduates will not only excel in technological proficiency but also become trusted contributors to digital progress, ensuring that computing serves human development and societal flourishing rather than narrow commercial interests.

1.2.2. Learning Outcomes of the Programme

On completion of the programme, graduates will be able to:

Academic and Cognitive Skills

i. apply analytical, mathematical, and computational reasoning to solve complex problems
ii. demonstrate competency in programming, web development, database management, and cybersecurity practices

iii. demonstrate digital literacy and technical competencies relevant to national and global industry demands
iv. apply computational thinking to break down complex problems into manageable components
v. critically evaluate emerging technologies of AI, blockchain, cloud computing for their societal and business impact
vi. demonstrate research skills by designing experiments, analysing data, and interpreting results for evidence-based decision making
vii. integrate interdisciplinary knowledge of business, social sciences, and technology to propose holistic solutions.

Professional and Career-Oriented Skills

i. demonstrate applied and industry-ready computing skills across multiple IT domains
ii. apply project management, stakeholder communication and application delivery practices in professional IT environments
iii. apply risk assessment, data management, and secure it practices ensuring reliable and resilient computing solutions
iv. manage end-to-end it projects, balancing scope, timelines, and resources effectively
v. demonstrate entrepreneurial skills by identifying opportunities for tech-enabled ventures
vi. apply industry standards and certifications to professional practice
vii. develop professional portfolios that demonstrate applied projects, internships, certifications, and technical competencies
viii. develop innovative and market-ready solutions by applying technical and entrepreneurial expertise.

Personal Development and Life Skills

i. promote ethical reasoning and responsible innovation aligned with sustainability and national values
ii. demonstrate lifelong learning, adaptability, and entrepreneurial thinking in evolving technological environments
iii. diversify growth pathways by preparing graduates for both industry roles and advanced study
iv. demonstrate adaptability and resilience when responding to technological change and uncertainty
v. demonstrate financial literacy in managing budgets for it projects and entrepreneurial initiatives
vi. practice mindfulness and wellbeing strategies to sustain productivity in high-pressure environments
vii. exhibit civic responsibility by aligning technological solutions with Bhutan's national priorities and values
viii. demonstrate confidence in independent decision-making and proactive leadership.

Interpersonal and Communication Skills

i. collaborate effectively across disciplines and cultures with accountability and leadership
ii. communicate technical concepts effectively through user-centred and accessible design approaches
iii. present and articulate technical and non-technical ideas clearly to diverse audiences.
iv. negotiate effectively with stakeholders to balance technical feasibility and business needs

v. demonstrate cross-cultural communication skills in global collaborative projects
vi. use storytelling and visualisation tools (dashboards, infographics, prototypes) to communicate technical insights clearly
vii. mentor peers and junior colleagues, fostering a collaborative and inclusive learning environment
viii. demonstrate transparent, empathetic, and professional communication when engaging with diverse stakeholders.

1.3. Career-related Opportunities

Graduates of the Bachelor of Science in Applied Computing and Information Technology (ACIT) will possess advanced technical, analytical, ethical, and professional competencies that prepare them for meaningful contributions to Bhutan's digital transformation and the evolving global information technology landscape. The programme equips graduates with the knowledge, skills, and attributes required to apply computing solutions to real-world challenges across a range of public, private, and community sectors, while demonstrating professionalism, innovation, and responsible use of technology.

The programme also prepares graduates to engage in entrepreneurship, innovation, and lifelong learning by fostering problem-solving, creativity, research capability, and human-centred design. Through experiential learning, industry engagement, and authentic projects, graduates will develop the ability to address emerging technological challenges within diverse organisational and societal contexts while contributing to sustainable and inclusive digital development.

In addition, the programme provides a strong foundation for further study and professional development in computing and related disciplines. Graduates will demonstrate transferable capabilities, including effective communication, teamwork, leadership, adaptability, ethical decision-making, and critical thinking, enabling them to work collaboratively across disciplines and respond to rapidly changing technological environments. These attributes position graduates to contribute responsibly to national development priorities and the broader advancement of the digital economy.

1.4. Justification of the Level of Award and Title

1.4.1. Level of Award

The proposed Programme is designed as a Bachelor’s degree (undergraduate level), justified by the intellectual complexity, graduate autonomy, and professional competencies it develops.

Learning complexity
Students engage with advanced computing domains such as AI, machine learning, cybersecurity, blockchain, and data science. These are taught not as isolated modules but as integrated challenges requiring analysis, synthesis, and application across technical and interdisciplinary contexts.

Graduate autonomy
The programme emphasises independent inquiry through capstone projects, industry collaborations, and applied problem-solving. Students are expected to plan, execute, and critically evaluate their own work, demonstrating autonomy in learning and professional practice.

Intellectual depth
Beyond technical mastery, students must integrate ethical reasoning, sustainability, entrepreneurship, and leadership into their solutions. This requires higher-order thinking, balancing innovation with social responsibility, and preparing graduates for complex decision-making in real-world contexts.

Graduate outcomes
The curriculum develops transferable competencies such as critical thinking, communication, teamwork, adaptability, and research capability alongside specialist knowledge. Graduates are prepared for professional roles, national digital transformation initiatives, and further academic progression.

Alignment with National Frameworks
The Bhutan Qualifications Framework (BQF) recognises the bachelor's level (Level 6) as appropriate for preparing graduates with both theoretical foundations and applied competencies to enter professional practice. The programme outcomes align directly with Level 6 descriptors:

- *Analytical thinking***:** Critical evaluation of advanced computing domains and synthesis of interdisciplinary perspectives.
- *Independent learning:* Capstone projects and industry collaborations requiring self-directed inquiry.
- *Application in unfamiliar contexts:* Experiential learning and case studies addressing novel problems in Bhutanese and global IT environments.
- *Professional judgement:* Decision-making grounded in institutional policies, ethical reasoning, and mindful technology practices.

Distinction from Diploma-Level Qualifications
Unlike diploma programmes, which emphasise foundational skills and guided learning, this bachelor's programme requires:

- **Cognitive complexity:** Engagement with advanced, integrated domains of computing and interdisciplinary problem-solving.
- **Research components:** Independent projects, industry exposure, and applied inquiry that demand critical evaluation and synthesis.
- **Student autonomy:** Responsibility for planning, executing, and reflecting on work, demonstrating readiness for professional practice and leadership.

These expectations exceed diploma-level descriptors and align with bachelor-level outcomes of independent learning, problem-solving, and leadership.

Comparative Benchmarking:
When benchmarking IT programmes, two frameworks stand out: ACM and IEEE. The ACM/IEEE Computing Curricula 2020 provides globally recognised guidelines, emphasising competency-based learning across core computing knowledge areas such as algorithms, systems, networking, AI/ML, cyber security, and cloud computing. These standards ensure international relevance, adaptability, and competitiveness, positioning programmes to meet the expectations of the global digital economy. In addition, alignment with frameworks such as ABET and the European e-Competence Framework (ECF) further strengthens employability, continuous improvement, and links to digital economy roles, while UNESCO/ITU digital skills frameworks add an inclusion and sustainability lens.

The proposed BSc ACIT is well-aligned with ACM/IEEE standards, ensuring a robust and comprehensive education for students. Comparable bachelor's programmes are offered internationally, such as Tribhuvan University's BSc in Computer Science and IT, University of Delhi's BSc (Hons) Computer Science, University of Colombo's BSc in Computer Science,

and Israel Institute of Technology's BSc in Information Systems Engineering. These examples demonstrate parity with regional and global standards, ensuring Bhutanese graduates are recognised and competitive worldwide. This benchmarking confirms that Bhutanese graduates will be recognised and competitive worldwide. Thus, the level of award is justified as a bachelor's degree because it requires sustained engagement with advanced knowledge, independent learning, and intellectual depth, aligned with BQF Level 6 descriptors and benchmarked against comparable international programmes.

1.4.2. Title of Award

The proposed title "Bachelor of Science in Applied Computing and Information Technology (BSc ACIT)" is justified as follows:

Accuracy of discipline representation
The curriculum integrates computing science foundations (programming, algorithms, systems design, data structures) with applied IT components (cloud solutions, web applications, emerging technologies such as AI and blockchain). The title reflects this dual emphasis: *Applied Computing* signals the scientific and problem-solving foundations, while *Information Technology* highlights practical applications in systems, networks, and digital innovation.

Clarity for stakeholders
The designation "Applied Computing and Information Technology" communicates clearly to employers, students, and academic institutions the graduate's area of expertise. It signals both technical depth and applied relevance, ensuring transparency in graduate competencies.

Comparative Benchmarking
The title is consistent with regional and international practice, as seen in the examples shared above. This demonstrates parity with comparable bachelor's programmes across South Asia and beyond, reinforcing the credibility and recognition of the proposed award. These examples demonstrate that the proposed title is aligned with established disciplinary conventions, ensuring recognition and mobility for Bhutanese graduates.

Global recognition
The "BSc" designation follows international conventions, facilitating graduate mobility, employability, and opportunities for further study. The combined focus on computing and IT ensures graduates are recognised as professionals capable of contributing to both technical innovation and applied digital transformation.

The title Bachelor of Science in Applied Computing and Information Technology (BSc ACIT) accurately represents the programme's disciplinary scope, aligns with international naming conventions, and ensures clarity for stakeholders. It is therefore appropriate to the curriculum and graduate competencies.

1.5. Justification and Demand for the Programme

1.5.1. Justification

The proposed Bachelor of Science in Applied Computing and Information Technology (ACIT) has been developed in response to Bhutan's increasing demand for graduates with contemporary computing knowledge, digital competencies, and ethical professional practices. The programme is aligned with national development priorities that recognise

information and communication technology (ICT) as a key driver of socio-economic development, innovation, digital transformation, and economic diversification.

The programme is informed by major national policy frameworks, including the Digital Drukyul Flagship Programme of the 12th Five Year Plan (2018 – 2023), the ICT Master Plan (iSherig 2), and the Bhutan Education Blueprint 2014 – 2024. Collectively, these frameworks emphasise the importance of strengthening digital capabilities, expanding ICT integration across education and public services, and developing a skilled workforce capable of supporting Bhutan's transition towards a knowledge-based and digitally enabled economy.

The programme also responds to His Majesty's vision of ensuring equitable access to knowledge, technology, and opportunities for all Bhutanese. As digital technologies continue to transform government, industry, education, and society, there is an increasing need for graduates who can develop, implement, and manage technology-driven solutions while demonstrating ethical responsibility, innovation, and professional competence.

The programme has been designed to reflect contemporary developments in applied computing through the integration of software development, cloud computing, cybersecurity, networking, data analytics, artificial intelligence, and human-centred computing. These areas are complemented by experiential learning, industry engagement, project-based learning, and professional practice to ensure that graduates are equipped with both technical expertise and transferable skills required in dynamic professional environments.

The programme also builds upon Royal Thimphu College's established commitment to academic quality, applied learning, and continuous improvement. By integrating technical competence with ethical reasoning, sustainability, innovation, and professional practice, the programme provides students with a relevant and future-oriented education that supports national development priorities while preparing graduates to contribute effectively within Bhutan and the wider global digital economy.

1.5.2 Evidence of Demand

The proposed programme is supported by evidence gathered through a comprehensive market needs assessment involving more than 400 participants, including employers, industry professionals, educators, graduates, and prospective students. The findings indicate strong demand for graduates with contemporary computing knowledge and applied technical skills. Approximately 97% of stakeholders identified the development of ICT professionals as essential to supporting Bhutan's digital transformation and emphasised the growing need for graduates with competencies in software development, cybersecurity, cloud computing, networking, data management, and emerging digital technologies. Many respondents also highlighted difficulties in recruiting suitably qualified local ICT professionals. Prospective student responses similarly demonstrated strong interest in computing education, with 85% expressing a preference for IT-related programmes, citing employment opportunities, interest in digital technologies, innovation, and entrepreneurship as key motivations. Feedback from graduates and industry representatives further emphasised the importance of practical learning experiences, industry engagement, project-based learning, and exposure to emerging technologies. These findings have directly informed the design of the proposed curriculum to ensure its relevance to current industry and national workforce requirements.

Alignment with National and Local Needs

The programme is designed to address both national and local development needs by preparing graduates with the knowledge, skills, and professional competencies required to support Bhutan's expanding digital economy and technology-enabled services. It

strengthens national capacity in applied computing and emerging digital technologies while responding to the increasing demand for ICT expertise across government, education, industry, and community sectors. Through its emphasis on applied learning, ethical practice, innovation, and industry engagement, the programme contributes to the development of a digitally competent workforce capable of supporting sustainable national development and responding effectively to evolving technological and societal needs.

Planned Intake and Workforce Gap

Labour market projections indicate that Bhutan will require a substantial increase in qualified ICT professionals over the coming years to support national development priorities and the continued expansion of the digital economy. Current graduate output from existing higher education institutions remains insufficient to meet projected workforce demand, resulting in an ongoing shortage of skilled computing professionals. The proposed programme will contribute to addressing this workforce gap through a planned and sustainable student intake that aligns with institutional capacity and available resources while maintaining academic quality. As the programme matures, graduate output will increase progressively to support national workforce requirements. By producing graduates with strong technical competence, professional skills, ethical awareness, and practical industry experience, the programme will make a meaningful contribution to strengthening Bhutan's ICT workforce and enhancing the country's capacity for digital innovation and sustainable development.

1.6. Programme Structure

All modules are 12-credit modules except ITP402 ACIT Capstone Project (60 credits), culminating in a total of 480 credits achieved over four years of full-time study. Each semester comprises approximately 15 – 16 weeks of learning and teaching, with around 40 hours of student effort per week, followed by two weeks of examinations.

The Programme is structured to ensure students' progress from foundational competence to advanced applied capability, while remaining rooted in mindful technology practice across the degree:

Foundational Knowledge
Year 1 focuses on building core competencies necessary for all computing disciplines. Students will acquire programming fundamentals, basic web development skills, understanding of digital logic and operating systems, and mathematical reasoning, alongside professional awareness through GNH principles. The emphasis is on foundational understanding, preparing students for more complex computational and applied topics in later years.

Core Computing Competencies
In Year 2, students will strengthen their core computing competencies. Programming knowledge progresses to data structures, algorithms, and software engineering principles. Database skills advance from theory to practical programming and students develop system administration and networking understanding, essential for managing IT infrastructure. Mathematics and communication skills continue to support analytical and professional capabilities. This year bridges foundational knowledge and applied competence, preparing students for specialised and emerging areas.

Applied and Specialised Knowledge
Year 3 introduces applied and specialised topics, where students integrate core knowledge into realistic projects and domain-specific applications. Students apply programming, software and database skills to AI, mobile, back-end, and IoT systems. Data analytics and cybersecurity modules enable students to handle complex, data-driven and secure

computing environments. HCI and project management modules prepare students for collaborative, human-centred and professional practice, while entrepreneurship develops business and innovation awareness. This year transitions students from core competence to specialised applied knowledge.

Integration, Professional Practice, and Capstone
Year 4 focuses on integration of knowledge and professional competence. Students tackle emerging technologies and enterprise-level systems, applying both technical and managerial skills. Research Methods prepare them to conduct structured investigations, while IT Laws and Ethics ensure responsible and ethical practice. The capstone project enables students to demonstrate comprehensive competence, integrating programming, data, networking, cybersecurity and HCI principles into a single applied project. By the end of this year, students have progressed from foundational understanding to advanced applied competence, ready for professional practice or higher study.

This structure ensures graduates progress from foundational knowledge to advanced applied skills, preparing them for diverse roles in Bhutan's digital economy and beyond while remaining rooted in mindful technology and GNH values.

Yr	Sem	Modules				
1	I	**CCS101** Digital Logic and Design	**PGL101** Fundamentals of Programming	**WEB101** Foundations of Web Development	**MTH101** Discrete Mathematics	**HCI101** Mindful Technology and Digital Wellbeing
	II	**CCS102** Operating System	**DMS101** Database Management System	**PGL102** Object-Oriented Programming	**MTH102** Calculus	**EAP101** Intermediate English for Academic Purposes
2	I	**CCS203** Computer Organisation and Architecture	**DMS202** Database Programming	**NET201** Networking Fundamentals	**PGL203** Data Structure and Algorithm	**EAP102** Upper-Intermediate English for Academic Purposes
	II	**NET202** System Administration	**SDP201** Systems Analysis and Design	**WEB202** Dynamic Front-end Development	**MTH203** Statistics and Probability	**DZG101** རྫོང་ཁ་བརྡ་དོན་སྒྲོད་ལེན།
3	I	**EMT301** Introduction to Artificial Intelligence and Machine Learning	**PGL304** Mobile App Development	**SDP302** Software Engineering	**WEB303** Back-end Development	**HCI302** Human Computer Interaction
	II	**CYS301** Introduction to Cyber Security	**BIS301** Data Analytics and Business	**EMT302** Internet of Things	**RIT301** Research Methodology	**EDP101** Entrepreneurship

			Intelligence			
4	**I**	**EMT403** Blockchain Technology and Applications	**BIS402** Enterprise Systems and Business Context	**HCI403** IT Project Management	**HCI404** IT Laws and AI Ethics	**ITP401** Project
	II	**ITP402** ACIT Capstone Project				

1.6.1. Modules Categories and Coding

The Thematic Categorisation of the Modules and Modules borrowed from other programmes and University Wide Modules are shown in the tables below:

Table1: *Thematic Categorisation of the Modules*

Theme/ Category	**Modules**
Human-Centred Computing, GNH, Ethics, Law and IT Governance	HCI101 Mindful Technology and GNH HCI302 Human Computer Interaction HCI404 IT Laws and AI Ethics
Programming and Software Development	PGL101 Fundamentals of Programming PGL102 Object-Oriented Programming PGL203 Data Structures and Algorithms PGL204 Mobile App Development SDP201 System Analysis and Design SDP302 Software Engineering
Web Technologies	WEB101 Foundations of Web Development WEB202 Dynamic Front-end Development WEB303 Back-end Development
Data Management and Analytics	DMS101 Database Management System DMS202 Database Programming BIS301 Data Analytics and Business Intelligence
Computer System and Infrastructure	CCS101 Digital Logic and Design CCS102 Operating System CCS203 Computer Organisation and Architecture
Networking and Cybersecurity	NET201 Networking Fundamentals NET202 System Administration

	CYS301 Introduction to Cyber Security
Emerging Technologies	EMT301 Introduction to Artificial Intelligence and Machine Learning EMT302 Internet of Things EMT403 Blockchain Technology and Applications
Business Enterprise	BIS402 Enterprise Systems and Business Context
Mathematics and Foundations	MTH101 Discrete Mathematics MTH102 Calculus MTH203 Statistics and Probability
Research and Projects	RIT301 Research Methodology ITP401 Project HCI403 IT Project Management ITP402 ACIT Capstone Project

Table 2: *Modules borrowed from other programmes*

Sl	**Modules**	**Borrowed from**
1	EAP101 Intermediate English for Academic Purposes	BA English Studies
2	EAP102 Upper-Intermediate English for Academic Purposes	BA English Studies

Table 3: *University-wide Modules*

Sl	**Modules**
1	DZG101 རྫོང་ཁ་བརྡ་དོན་སྒྲུབ་ལེན།
2	EDP101 Entrepreneurship

1.7. Learning and Teaching Approach

The learning and teaching approach of this Programme is progressive and developmental, evolving across the four years of study. Student-centred learning remains the core pedagogical principle, with each academic year introducing new focal components that gradually enhance students' independence, technical competence, and professional readiness.

Year I: Foundation and Guided Learning
In the first year, teaching is student-centred but primarily teacher-led, with close guidance from tutors. Emphasis is placed on building strong theoretical foundations through structured lectures, guided laboratory sessions, and tutor-supported problem-solving activities. These

methods will help students to develop core conceptual understanding and essential technical skills that would be useful in the coming semesters.

Year II: Active and Independent Learning
In the second year, teacher-led would be reduced to give opportunity to promote students' engagement. This would be achieved through hands-on lab work, group problem-solving, coding exercises, system design discussions, student presentations, and guest lectures from industry professionals.

Year III and IV: Student-Led Learning and Professional Practice
By the third year, students take a more active role by leading discussions, presenting solutions, and collaborating on technical projects. Courses include hands-on labs and group work that simulate real-world tasks such as building systems, testing tools, and solving practical problems.

In the final year, the focus shifts to independent, professional-level work. Students complete capstone projects addressing real or industry-relevant problems, managing all stages from planning to delivery. By graduation, they are expected to demonstrate job-ready skills in problem-solving, communication, and full project execution.

Placements / Work-based Learning
The module ITP402 ACIT Project provides students a structured work-based learning opportunity. Students spend enough time in their assigned organisations or innovation projects, gaining hands-on experience in software development, system administration, networking, and data management or IT support. Detailed requirements, supervision mechanisms, and assessment criteria are outlined in the module descriptor.

1.8. Assessment approach

Assessments are structured into two components: Continuous Assessment (CA) and the Semester-End Examination (SEE). While mid-semester and SEE remain part of most modules, the primary emphasis is on CA to support ongoing learning and skill development.

1.8.1. Continuous Assessment (CA)

CA assessments: assignments, projects, presentations, and class activities would provide an ongoing evaluation of their learning. These assessments would encourage continuous engagement, reinforce learning, and allow students to progressively develop knowledge, skills, and competencies aligned with the programme outcomes. CA also enables timely feedback, helping students identify gaps and improve their performance throughout the learning process.

In the lower years, where the focus is on building foundational knowledge, assignments enable students to assess their understanding, while tests help reinforce and consolidate key concepts. As student progress to the upper years, projects provide opportunities to apply theoretical knowledge in practical contexts, and viva assessments help develop confidence and the ability to articulate ideas clearly. Advanced-level tests further evaluate students' creativity, critical thinking, and problem-solving skills. Finally, modules such as the capstone project allow students to integrate and demonstrate their overall learning, preparing them effectively for professional practice and the demands of the job market.

1.8.2. Semester-End Examination (SEE)

The SE Examination, on the other hand, offers a comprehensive and standardised measure of students' overall understanding at the end of the semester. It assesses the integration,

retention, and application of knowledge under controlled conditions, ensuring that learning outcomes are achieved at the required level.

Together, CA and SEE ensure a balanced assessment approach, where CA supports continuous learning and skill development, and SE validates the attainment of programme learning outcomes in a holistic and consistent manner.

1.8.3. Assessment Tools and Rubrics

Most CA components are assessed using grading rubrics tailored to the nature of the task and module requirements. Written tests primarily assess foundational knowledge, while projects, presentations, and applied tasks evaluate higher-order skills and professional readiness.

1.9. Regulations

1.9.1. Entrance Requirements

Student Background:	**Min. Entrance Requirements / Eligibility Criteria**
Bhutanese Student*	BHSEC Class XII pass (or equivalent for Bhutanese studying outside Bhutan) For Science Students ● 50% aggregate in best 4 subjects including Mathematics ● 45% in Mathematics ● Pass in Dzongkha ● 50% marks in English For Arts and Commerce Students ● 50% aggregate in best 4 subjects including Business Mathematics ● 50 % in Business Mathematics ● Pass in Dzongkha ● 50% marks in English
Non-Bhutanese**	Passing score on ISCE/BHSEC, or equivalent secondary education certificate from home country For Science Students ● 50% aggregate in best 4 subjects including Mathematics ● 45% in Mathematics ● 50% marks in English For Arts and Commerce Students ● 50% aggregate in best 4 subjects including Business Mathematics ● 50 % in Business Mathematics ● 50% marks in English

**Applicants who have not studied Dzongkha in class XII should have passed it in class X. A proficiency test in Dzongkha will determine the eligibility of Bhutanese applicants who have not studied Dzongkha in either class X or XII.*

*** Dzongkha Communication will be substituted with a module from a collection of approved alternative modules for foreign students.*

1.9.2. Assessment and Progression Requirement

The assessment regulation and criteria for progression from one semester to the next, including the final award criteria are as per the guidelines given under section D1 in The Wheel of Academic Law, RUB (latest version available at http://www.rub.edu.bt/), subject to any amendment or revision as made by the Academic Board of the University.

1.10. Programme Management, Quality Assurance, and Enhancement

Programme governance and quality assurance are structured in accordance with Section F6 of the RUB Wheel of Academic Law and the Governance Manual. Oversight is provided at multiple levels to ensure academic standards, effective delivery, and continuous improvement.

President
At the institutional level, the College Academic Committee (CAC), chaired by the President as the Head of the College (at RTC), is the principal authority for academic standards and quality. It includes senior management, academic and student representatives, staff representatives, and an external member. All Programme committees and Boards of Examiners report to the CAC, which approves minor Programme changes and endorses Annual Programme Monitoring Reports.

Dean of Academic Affairs (DAA)
The Dean of Academic Affairs will:

- provide strategic leadership and direction for all academic programmes to ensure alignment with institutional goals and standards.
- oversee the planning, development, implementation, and evaluation of academic policies and curricula.
- ensure quality assurance and continuous improvement in teaching, learning, and assessment practices.
- coordinate with Programme Leaders, Module Coordinators, and faculty to promote academic excellence and innovation.
- facilitate professional development opportunities for academic staff to enhance instructional effectiveness.
- monitor student academic performance and implement interventions to support student success.
- represent academic affairs in institutional decision-making processes and liaise with external stakeholders where necessary.
- oversee academic scheduling, resource allocation, and compliance with regulatory requirements.
- ensure proper documentation and reporting of academic activities to relevant authorities.

Programme Leader
The BSc in ACIT Programme is managed and coordinated by the Programme Leader under the guidance of the Dean.

The Programme Leader will:

- organise and coordinate all Programme related activities with highest level of efficiency and competence.
- lead the academic development of the Programme.

- liaise and maintain proper/regular linkages with the Programme Tutors and students for the effective operation of the Programme.
- submits Programme reports, plans and Programmes to concerned authorities.
- maintain proper records of the Programme activities for future reference.
- mobilise teaching learning resources to facilitate effective instructional delivery and optimal learning in the students.
- plan and prepare the budget required for the activities of the different years and accordingly use the budget approved judiciously.
- monitor the operation of the Programme on an ongoing basis, and to co-ordinate its annual monitoring.
- allocate appropriate staff for teaching and other duties required by the Programme.

Module Coordinator
The Module Coordinator will:

- plan, organise, and coordinate all activities related to the assigned module with efficiency and effectiveness.
- develop and review module content, learning outcomes, and assessment strategies in alignment with programme objectives.
- liaise regularly with Module Tutors to ensure consistency and quality in module delivery.
- monitor the progress of students within the module and address academic concerns in a timely manner.
- ensure that teaching and learning resources for the module are adequate, relevant, and up to date.
- coordinate the preparation, moderation, and submission of assessments and results.
- maintain proper records of module activities, student performance, and feedback.
- contribute to module evaluation and continuous improvement processes.
- report module-related matters to the Programme Leader as required.

Module Tutor
The Module Tutor will:

- deliver teaching and learning sessions effectively in accordance with the module plan and schedule.
- prepare lesson plans, instructional materials, and assessments to support student learning.
- facilitate an engaging and inclusive learning environment that promotes active participation.
- assess and provide timely, constructive feedback on student performance.
- support students academically and provide guidance to enhance their learning outcomes.
- maintain accurate records of attendance, assessments, and student progress.
- collaborate with the Module Coordinator to ensure alignment with module objectives and standards.
- participate in meetings, training, and professional development activities as required.
- contribute to continuous improvement by providing feedback on module delivery and student learning experiences.

College Academic Committee (CAC)
The College Academic Committee (CAC) is the highest academic decision-making body at the College level and operates in accordance with Section F6 of the Wheel of Academic Law of the Royal University of Bhutan. The CAC members will comprise;

- President - Chair
- Dean of Academic Affairs – Member Secretary
- Dean of Research and Industrial Linkages

- Dean of Student Affairs
- An elected staff representative
- Representative of other groups of staff including up to 4 Programme Leaders. *Rest of the PLs can be invited as and when the need is felt.
- An elected student representative
- An external member (Optional)

The CAC is the overarching authority on all academic issues and ultimate guarantor of standards and quality at the college-wide level and for the University. All Programme management committees and examiners report to the CAC. The CAC should be consulted at the beginning of each semester to approve minor changes to modules in the Programme under guidelines specified by the University on allowable changes.

Programme Committee (PC)
The Programme Committee will comprise:

- The Programme Leader (Chair)
- One module tutor from amongst those teaching major modules
- One module tutor from amongst those teaching the professional modules
- Two student representatives
- One external member

The Programme Committee will:

- ensure the effective conduct, organisation and development of the Programme.
- ensure the overall academic quality of the Programme.
- formally submit the necessary documents for approval, accreditation or assessment of the Programme endorsed by the College Academic Committee.
- develop human resources.

Programme Board of Examiners (PBE)
Assessment and progression decisions are delegated to the Programme Board of Examiners (PBoE), which includes an independent Chair, the PL, Programme tutors, and an External Examiner.

PBoE is responsible for ensuring the quality of assessment. The board will follow the general assessment regulations of the Royal University of Bhutan.

The members of the Programme Board of Examiners will comprise:

- An independent Chair
- The Programme Leader
- All module tutors
- One module tutor from amongst those teaching the professional modules
- External examiner

The Programme Board of Examiners will:

- ensure and establish a sound academic structure of quality and standard by closely monitoring the Programme design, evaluation and assessment mechanism for judging the quality of the Programme.
- conduct module evaluation at the end of each module to look at the quality of teaching and learning. Based on the feedback from the students each tutor makes changes for the next teaching and submits these to the Programme Leader, where required.
- coordinate analysis of the module evaluation with respective supervisors and devise improvement strategies.
- decide on the award to be granted to each student on completion of, or decision to exit from the Programme.

- decide on the eligibility and suitability of transfer of student's academic credit from other universities.

Moderation of Assessment
All assessments contributing 20% or more of a module's marks are formally moderated in line with the Wheel of Academic Law.

External Examiner
Quality monitoring is supported through structured reporting and external oversight. Module tutors submit semesterly reflective reports to the PL, who compiles an Annual Programme Monitoring Report (APMR) with input from tutors and an External Examiner (EE). The EE reviews student performance, assessment quality, resources, and staffing, and their report forms a key part of the APMR submitted to the University.

Student Consultative Meeting
To maintain continuous dialogue between students and the College administration, student consultative meetings are held every semester by the Dean and Associate Dean. These meetings provide a formal platform for students to share feedback, raise concerns, and suggest improvements related to academic Programmes, campus facilities, and student support services.

The Dean consolidates the findings and reports them to the College Management Team and relevant Programme leaders, who take responsibility for addressing the issues raised and communicating decisions or actions to students.

Such meetings enhance transparency, strengthen student engagement in institutional development, and ensure that the learning environment remains responsive to student needs.

Student Module Evaluation
At the end of every semester, students complete a module evaluation survey to provide feedback on teaching quality, learning resources, assessment methods, and overall module delivery. These evaluations are systematically collected and reviewed by the Dean's Office, which consolidates the findings and identifies areas requiring improvement.

Any issues or concerns raised are discussed with the respective Programme leaders and faculty members, and appropriate actions are taken to address them. Outcomes and follow-up measures are communicated back to students where relevant, ensuring transparency and continuous enhancement of teaching and learning practices.

Additional quality assurance mechanisms include regular tutor performance evaluations, classroom observations, self-appraisals, and structured Continuous Professional Development (CPD). New tutors receive enhanced orientation and observation support to ensure effective integration into teaching practices.

Module coordination promotes consistency across multiple sections through shared assessments, peer review, moderation, and, where appropriate, cross-teaching.

Student-focused quality systems include the RTC Gateway for transparency, semesterly module and tutor evaluations using university-approved instruments, formal student consultative meetings, and academic advising. Clear procedures are also in place for module repeats, with structured academic support to ensure equivalence with regular module delivery.

Together, these governance and quality assurance arrangements provide strong academic oversight, foster meaningful student engagement, ensure fairness and consistency in assessment, and drive the continuous enhancement of teaching and learning.

2. Module Descriptors

Year 1 Semester I

Module Code and Title: **CCS101 Digital Logic and Design**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Arun Kumar Singh

General Objective
The module aims to provide students with a comprehensive understanding of digital logic and computer organisation, enabling them to build a foundation for designing and analysing digital systems.

Learning Outcomes
On completion of the module, students will be able to:

i. use binary systems, decimal codes, and number conversions
ii. apply Boolean algebra laws and theorems to simplify logic expressions
iii. design combinational logic circuits using standard design procedures
iv. analyse combinational circuits
v. implement digital circuits using MSI and LSI components
vi. design sequential circuits using flip-flops, registers, and counters
vii. use simulation tools to test and verify digital circuit functionality
viii. interpret timing diagrams in digital systems
ix. explain the basic organisation of memory in digital systems.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent study	Practical assignments and simulations	3	60
	Self-study	1	
Total		**8**	**120**

Assessment Approach

A. Class Test (10%)
This will assess students' practical understanding of fundamental concepts from Unit I and Unit II. The duration of the test will be 40 minutes and it will be conducted out of 30 marks.

B. Laboratory Practical Works (20%)
Exercises will evaluate students' hands-on skills in simplifying Boolean functions, designing combinational logic, and implementing circuits, as covered in Units III-VI. Practical exercises will include lab exercises, K-map simplification, logic gate design tasks, MSI/LSI circuit modelling, and simulation using digital logic software tools.

6 Marks Accuracy of solutions (Correct simplification of Boolean functions, correctness in logic design and circuit implementation)
5 Marks Completeness of tasks (Coverage of all required exercises: K-map, logic gate design, MSI/LSI tasks, and simulations)
5 Marks Practical application (Ability to apply theoretical concepts effectively in practical exercises)
4 Marks Presentation (Well-documented steps, clear diagrams, and organised submission)

C. Practical Examination (15%)

This assessment will evaluate students' ability to design, analyse, and implement sequential and combinational circuits, including flip-flops, counters, and registers, as covered in Units IV-VII.

6 Marks Accuracy (Correct design and implementation; circuit operates correctly and produces the expected outputs)

5 Marks Simulation and verification (Proper use of simulation and verification tools to test circuit functionality; correct interpretation of simulation results and timing behaviour)

4 Marks Application of theoretical concepts. (Demonstrates understanding and correct application of relevant theoretical concepts such as flip-flops, counters, registers, and circuit design principles.)

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours.

E. Semester-End Examination (35%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours and converted to 35%.

Overview of Assessment Approaches and Weighting

Areas of Assessment	Quantity	Weighting (%)
A. Class Test	1	10
B. Laboratory Practical Works	1	20
C. Practical Examination	1	15
D. Mid-Semester Examination	1	20
E. Semester-End Examination (SEE)	1	35
Total		100

Pre-requisites: None

Subject Matter

Unit I: Number Systems

1.1. Digital Systems
1.2. Binary Numbers
1.3. Number Base Conversion
1.4. Octal and Hexadecimal Numbers
1.5. Complements
1.6. Signed Binary Numbers
1.7. Decimal codes (BCD, 2 4 2 1,8 4 -2 -1, Excess 3, Gray Code)
1.8. Binary Storage and Registers
1.9. Binary Logic

Unit II: Boolean algebra and Logic Gates

2.1. Basic and Axiomatic definitions of Boolean Algebra
2.2. Basic Theorems and properties of Boolean Algebra
2.3. Boolean Functions
2.4. Logic Operations
2.5. Logic Gates
2.6. Integrated Circuits

Unit III: Simplification of Boolean Functions

3.1. K-map
3.2. Two and Three variable maps
3.3. Four variable maps

3.4. Product of sum simplification
3.5. NAND and NOR implementation
3.6. Don't Care conditions
3.7. Determinant and selection of Prime Implicants

Unit IV: Combinational Logic

4.1. Design Procedure
4.2. Adders
4.3. Subtractors
4.4. Code Conversions
4.5. Analysis Procedure
4.6. Multilevel NAND and NOR Circuits
4.7. Exclusive-OR Circuits

Unit V: Combinational Logic with MSI and LSI

5.1. Binary Parallel Adder and Subtractor
5.2. Decimal Adder
5.3. Magnitude Comparator
5.4. Decoders and Encoders
5.5. Multiplexers
5.6. Read-only-Memory (ROM)
5.7. Programmable Logic Array (PLA)
5.8. Programmable Array Logic (PAL)

Unit VI: Synchronous and Asynchronous Sequential Logic

6.1. Flip-Flops
6.2. Triggering of Flip-flops
6.3. Analysis of clocked sequential circuits
6.4. Design Using State Equations and State Reduction Tables
6.5. Introduction to Asynchronous Circuits
6.6. Circuits with Latches.

Unit VII: Registers and Counters

7.1. Registers
7.2. Shift registers
7.3. Ripple Counters
7.4. Synchronous Counters
7.5. Timing Sequences
7.6. Memory Organisation

List of Practicals

i. Implementation and Verification of Basic Logic Gates (AND, OR, NOT, NAND, NOR, XOR)
ii. Design and Implementation of Combinational Circuits using Boolean Expressions
iii. Design and Implementation of Binary Code Converters (Binary–Gray, Gray–Binary, BCD–Excess-3)
iv. Design and Implementation of Combinational Logic Circuits using Multiplexers
v. Design and Implementation of Half Adder, Full Adder, Half Subtractor, and Full Subtractor Circuits
vi. Implementation and Analysis of Flip-Flops (SR, JK, D, and T Flip-Flops)
vii. Design and Implementation of Sequential Circuits using Flip-Flops
viii. Design and Verification of Counters (Asynchronous and Synchronous Counters)
ix. Design and Testing of a Clock Pulse Generator using 555 Timer

Reading List

Essential Reading

Mano, M. M. (2019). *Digital Logic and Computer Design* (6th ed.). Pearson India.
Roth, C. H. (2013). *Fundamentals of Logic Design* (7th ed.). Cengage Learning.

Additional Reading

Hayes, J. P. (1993). *Introduction to digital logic design* (1st ed.). Addison-Wesley.
Holdsworth, B., & Woods, C. (2002). *Digital Logic Design* (4th ed.). Newnes/Elsevier.
Mano, M. M., Kime, C. R., & Martin, T. (2015). *Logic & Computer Design Fundamentals* (5th ed.). Pearson.
Wakerly, J. F. (2018). *Digital Design: Principles and Practices* (5th ed.). Pearson.

Open-Source Reading

All About Circuits. (Open Textbook). *Digital Circuits*. Available at: https://www.allaboutcircuits.com/textbook/digital/
Kuphaldt, T. R. (Open Textbook). *Lessons in Electric Circuits – Digital*. Available at: https://www.ibiblio.org/kuphaldt/electricCircuits/Digital/

Date: May 2026

Module Code and Title: **PGL101 Fundamentals of Programming**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Ngawang Tenzin

General Objective

This module aims to introduce students to programming though Python, fostering computational thinking, problem-solving and algorithmic design. It seeks to build confidence in applying programming concepts to real-world problems while establishing disciplined coding practices and a strong foundation for advanced study in computing.

Learning Outcomes

On completion of the module, students will be able to:

i. develop algorithms and flowcharts to plan solutions to computing problems
ii. design functional and modular Python programs
iii. apply decision-making and iterative processes to solve problems efficiently
iv. create reusable programme components to enhance code maintainability
v. manipulate data to support computational tasks
vi. implement robust error handling and debugging strategies
vii. utilise libraries and tools to improve programme functionality and workflow
viii. demonstrate good programming practices, including readability, testing and version control.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent Study	Assignment	2	60
	Self-study	2	
Total		**8**	**120**

Assessment Approach

A. VLE Quiz (10%)

One VLE quiz will be conducted after completion of the first three units and will consist of multiple-choice questions (MCQs) and short-answer type question, focused on interpreting simple code and identifying correct outputs. The quiz will be conducted for a duration of 40 minutes with the maximum marks of 30.

B. Laboratory Practical Work (30%)

The laboratory practical work will evaluate students' ability to implement programming solutions correctly and consistently. It will be divided into three parts: Laboratory Assignments (15%), Practical Examination (10%) and Viva Voce (5%).

i. **Laboratory Assignment (15%)**

Laboratory Assignments (a total of 8) will be based on the concepts taught in the theory class. Students will be provided a set of programming questions every week covering the concepts taught in each unit. The students will have to submit the solutions at the end of the lab session through the VLE.

The following criteria will be used to assess the set of programming questions. Each set will be assessed out of 20 marks.

4 Marks	Problem Solving approach (Logical approach and efficiency of the approach)
5 Marks	Use of Concepts (Application of concepts such as oops, functions, data handling)
8 Marks	Coding (Correct syntax and function)
3 Marks	Output (Correct output and meeting all the requirements)

ii. **Practical Examination (10%)**

A practical examination will be conducted after the completion of the module and before the semester-end examination and will cover all the units to assess students' ability to solve programming problems. The exam will be marked out of 50 and converted to 10%. The grading will focus on:

2 Marks	Problem-solving approach (clarity and justification of the approach) Accuracy of program output (correctness of solutions)
6 Marks	Coding (coding style, logical flow, functions, efficiency and syntax)
2 Marks	Output (correct outputs and meeting all the requirements)

iii. **Viva Voce (5%)**

An oral examination will be conducted to evaluate students' understanding of programming concepts and their ability to explain solutions for the laboratory examination. The viva will be (maximum 5 minutes per student) marked out of 15 and converted to 5%. The grading will focus on:

6 Marks	Conceptual understanding (clarity of Python fundamentals, operators, flow control and syntax)
4 Marks	Ability to explain solutions (logical reasoning, articulation and clarity of thought)

3 Marks Application to practical scenarios (linking theory to practice, examples from assignments)
2 Marks Confidence and communication skills (clarity, conciseness, professional presentation)

C. Mid-Semester Examination (20%)
The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours and converted to 20%.

D. Semester-End Examination (40%)
The Semester-end examination will be conducted for 100 marks over a duration of three hours and will be converted to 40%.

Overview of the Assessment Approaches and Weighting

Areas of Assignments		**Quantity**	**Weighting**
A. VLE Quiz		1	10%
B. Laboratory Practical Works	Laboratory assignments	8	15%
	Practical examination	1	10%
	Viva-voce	1	5%
C. Mid-Semester Examination		1	20%
D. Semester-End Examination (SEE)		1	40%
Total			100%

Pre-requisites: None

Subject Matter

Unit I: Introduction to Programming and Python Basics
1.1. Problem-solving and computational thinking
1.2. Programming Paradigm - Structural Programming and Object-Oriented Programming
1.3. Algorithms, pseudocode and flowcharts
1.4. Python environment, running scripts, basic input/output
1.5. Debugging mindset (reading errors, using print/debug tools)

Unit II: Variables, Data Types, and Operators
2.1. Variables and naming conventions
2.2. Numeric types, strings, booleans
2.3. Operators and expressions
2.4. Type conversion and formatted output

Unit III: Control Flow
3.1. Conditionals (if, elif, else)
3.2. Loops (for, while) and common loop patterns
3.3. Nested loops and structured iteration
3.4. Basic validation patterns

Unit IV: Functions and Modularity
4.1. Defining functions, parameters, return values
4.2. Scope and local/global concepts
4.3. Decomposition: breaking problems into small functions

Unit V: Data Structures in Python
5.1. Strings and common operations
5.2. Lists and list methods
5.3. Tuples and immutability

5.4. Sets and basic set operations
5.5. Dictionaries (key/value modelling)

Unit VI: File Handling and Exceptions

6.1. Reading and writing text files
6.2. Working with CSV files
6.3. JSON reading/writing
6.4. Exception handling for robustness

Unit VII: Python Standard Libraries and External Libraries

7.1. Selected standard libraries: math, random, datetime and os
7.2. Introduction to code style (readability, naming, functions)
7.3. Introduction to Git (repositories, commits, basic workflow)
7.4. Testing basics

List of Practicals

i. Getting started: I/O, expressions, simple scripts
ii. Conditionals: decision problems
iii. Loops: counters, accumulators, patterns, simple analytics
iv. Functions: decomposition and reusable solutions
v. Lists/dictionaries: lookup, counting, grouping
vi. File handling: parse and summarise small data files
vii. CSV/JSON: basic reading/writing for structured data
viii. Debugging practice: interpreting and fixing common errors

Reading List

Essential Reading

Downey, A. (2015). *Think Python: How to think like a computer scientist* (2nd ed.), Green Tea Press.

Heinold, B. (2012). *A practical Introduction to Python programming*. Mount St. Mary's University

Zelle, J. M. (2017). *Python programming: An introduction to computer science* (3rd ed.). Franklin, Beedle & Associates.

Additional Reading

Deitel, P., Deitel, J. (2019). *Introduction to Python for computer science and data science learning to program with AI, Big Data and the Cloud*. Pearson

Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.

Rao, R. N. (2017). *Core Python programming*. Dreamtech Press.

Sweigart, A. (2019). *Automate the boring stuff with Python* (2nd ed.). No Starch Press.

Date: May 2026

Module Code and Title: **WEB101 Foundations of Web Development**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Phuntsho Wangdi

General Objective

This module aims to build foundational front-end web skills and the mental model of how the web works. Students will learn and publish small, accessible, responsive, and

standards-compliant websites, providing the foundational basis for software development modules.

Learning Outcomes
On completion of the module, students will be able to:

i. explain the fundamental concepts of the World Wide Web
ii. describe the HTTP protocol and the request–response cycle
iii. identify client and server roles in web communication
iv. explain the structure and components of URLs
v. describe the steps involved in HTML parsing by web browsers
vi. create semantic HTML5 documents using appropriate elements, forms, and media
vii. apply CSS3 to style web pages
viii. implement layout using Flexbox and Grid
ix. write basic JavaScript to manipulate the Document Object Model
x. apply accessibility guidelines and localisation principles in web design
xi. use browser DevTools, Git, and deployment techniques to manage and publish websites.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent Study	Assignment and Research	3	60
	VLE Discussion	1	
Total		**8**	**120**

Assessment Approach

A. VLE Quiz (10%)
Two quizzes, comprising of multiple-choice and short-answer questions will be conducted. The quizzes would cover topics from Unit I and Unit IV respectively. Each quiz will be for 35 minutes and conducted out of 30 marks**,** the total marks will be converted to 10%.

B. Practical Assignment I – Semantic HTML and Accessible Forms (10%)
Individual student will build a multi-section HTML page with semantic structure, media and accessible forms, enabling them to learn best practices for building standards-compliant and accessible web content. This will be assessed based on the following criteria:

3 Marks	Semantic structure and validity (Headings, sections, landmarks
3 Marks	Accessible forms (Labels, input types, error messaging)
2 Marks	Media (Descriptive alt text, captions/transcripts, correctly sized media.)
2 Marks	Organisation and documentation (Logical file structure, README, commented code.)

C. Practical Assignment II – CSS Layout and Responsive Design (10%)
Individual students will implement a responsive layout with attention to typography, spacing, colour contrast, and mobile-first design, allowing them to develop skills in applying effective design principles that would improve usability and accessibility across different devices. This will be assessed based on the following criteria:

5 Marks	Responsive Layout (Layout adapts to all screen sizes, elements aligned and proportioned correctly)

3 Marks Typography (Consistent fonts, readable text, clear hierarchy of headings and body text)
2 Marks Spacing and Alignment (Proper margins/padding, consistent alignment, balanced visual spacing)

D. Practical Examination (10%)

Students will individually complete an in-lab task where they will add basic inter-activity to a web page using JavaScript DOM events and simple form validation. They will also use browser Developer Tools (DevTools) to identify and fix minor HTML, CSS, or JavaScript issues, developing hands-on skills in interactive web development and debugging. This will be assessed based on the following criteria:

4 Marks Correctness of Code (Code runs correctly and performs the required functionality without errors)

3 Marks Documentation (Includes appropriate comments or brief descriptions explaining the code)
3 Marks Readability (Code is clear, properly indented, and uses meaningful variable/function names)

E. Project (20%)

This project is assigned at the start of the first week and must be submitted by the end of the fifth week. Students will work individually to complete a project by incorporating topics from units I to VII. This project is to be implemented using a web development framework to assess their understanding of web development concepts in order to develop functional web applications for a given requirement. The project will be evaluated based on the following criteria:

6 Marks Content (Application of relevant web development concepts,
Technical accuracy and correctness of implementation,
Relevance of features to project requirements,
Logical structure and organization of components)

6 Marks Completeness (Inclusion of all required features and functionalities, Proper functioning of each feature, Fulfilment of all project requirements, handling of basic user interactions and inputs)
4 Marks Documentation (Clear description of project objectives and requirements, Explanation of implemented features, Alignment between requirements and implementation, Clarity and organization of documentation)
4 Marks Consistency (Uniform design, coding standards, and functionality)

F. Semester-End Examination (40%)

The Semester-End Examination will be conducted for 100 marks over a duration of three hours and converted to 40%.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Quizzes	2	10
B. Practical Assignment I	1	10
C. Practical Assignment II	1	10
D. Practical Examination	1	10
E. Project	1	20
F. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: None

Subject Matter

Unit I: Web Foundations

- 1.1. Web Model
  - 1.1.1. Introduction to the World Wide Web
  - 1.1.2. Hypertext Transfer Protocol (HTTP)
  - 1.1.3. HTTP request–response cycle
  - 1.1.4. Uniform Resource Locators (URLs) and their structure
  - 1.1.5. Client–server architecture and roles
  - 1.1.6. Web assets (HTML, CSS, JavaScript, images, fonts, media)
  - 1.1.7. Browser rendering process
- 1.2 DOM Concept
  - 1.2.1. Introduction to the Document Object Model (DOM)
  - 1.2.2. HTML parsing process
  - 1.2.3. Building the DOM tree
  - 1.2.4. Introduction to CSS Object Model (CSSOM)
  - 1.2.5. Relationship between DOM and CSSOM
  - 1.2.6. Rendering tree and page display
- 1.3 Static vs Dynamic Websites
  - 1.3.1. Static websites and their characteristics
  - 1.3.2. Dynamic websites and server-side interaction
  - 1.3.3. Difference between static and dynamic content
  - 1.3.4. Separation of content, structure, and behaviour
  - 1.3.5. Progressive enhancement and graceful degradation
- 1.4 Evolution of Web Technology
  - 1.4.1. Web 1.0 – Read-only web
  - 1.4.2. Web 2.0 – Interactive and social web
  - 1.4.3. Web 3.0 – Semantic and intelligent web
  - 1.4.4. Evolution of web standards (HTML, CSS, JavaScript)
  - 1.4.5. Modern web technologies and APIs

Unit II: HTML5 Structure and Semantics

- 2.1 Document Structure
  - 2.1.1 HTML5 document skeleton
  - 2.1.2 Doctype declaration
  - 2.1.3 <head> element and metadata
  - 2.1.4 Viewport configuration
  - 2.1.5 <body> structure and page content
- 2.2 Semantic Elements
  - 2.2.1 Headings and document outline
  - 2.2.2 Sectioning elements (section, article)
  - 2.2.3 Navigation elements (nav)
  - 2.2.4 Structural elements (header, footer)
  - 2.2.5 Lists and tables
- 2.3 Forms and Media
  - 2.3.1 HTML form structure
  - 2.3.2 Input types and labels
  - 2.3.3 Form validation attributes
  - 2.3.4 Embedding images
  - 2.3.5 Audio and video elements
  - 2.3.6 Hyperlinks and navigation
- 2.4 Valid Markup and Accessibility
  - 2.4.1 Importance of valid HTML markup
  - 2.4.2 HTML landmarks for accessibility

2.4.3 Introduction to ARIA (Accessible Rich Internet Applications)
2.4.4 Basic accessibility practices

Unit III: CSS3 Styling and Layout

3.1 Core CSS Concepts
3.1.1 CSS syntax and rules
3.1.2 Cascade and inheritance
3.1.3 Specificity rules
3.1.4 Selectors and combinators
3.1.5 Units and colour systems
3.2 CSS Box Model
3.2.1 Margin, border, padding, and content
3.2.2 Display properties
3.2.3 Positioning (static, relative, absolute, fixed)
3.2.4 Stacking context and z-index
3.3 Layout Systems
3.3.1 Introduction to Flexbox
3.3.2 Flex container and flex items
3.3.3 CSS Grid layout system
3.3.4 Grid container and grid items
3.3.5 Common page layout patterns
3.4 Responsive Design
3.4.1 Mobile-first design approach
3.4.2 Media queries
3.4.3 Fluid typography
3.4.4 Responsive spacing and layout adjustments
3.5 Design Hygiene
3.5.1 Typography best practices
3.5.2 Spacing scale and visual rhythm
3.5.3 Colour contrast and readability
3.5.4 Design consistency

Unit IV: Accessibility and Inclusive Design

4.1 WCAG-Aligned Practices
4.1.1 Web Content Accessibility Guidelines overview
4.1.2 Text alternatives for images
4.1.3 Labels and accessible forms
4.1.4 Keyboard navigation and focus management
4.2 Inclusive Design Principles
4.2.1 Colour and contrast considerations
4.2.2 Proper heading hierarchy
4.2.3 Semantic HTML vs ARIA roles
4.2.4 Accessible form design
4.3 Accessibility Evaluation
4.3.1 Basic accessibility audit tools
4.3.2 Accessibility checklists
4.3.3 Ethical and cultural considerations in design

Unit V: JavaScript (ES6) Fundamentals

5.1 Core JavaScript Concepts
5.1.1 Introduction to JavaScript
5.1.2 Variables (let, const)
5.1.3 Data types
5.1.4 Operators and expressions
5.1.5 Control flow statements

5.2 Functions and DOM Interaction
- 5.2.1 JavaScript functions
- 5.2.2 Event handling
- 5.2.3 Accessing DOM elements
- 5.2.4 Updating content dynamically
- 5.2.5 Simple form validation

5.3 Separation of Concerns
- 5.3.1 Unobtrusive JavaScript principles
- 5.3.2 Keeping HTML, CSS, and JS separate
- 5.3.3 Common JavaScript anti-patterns to avoid

Unit VI: DevTools, Git and Deployment

6.1 Browser Developer Tools
- 6.1.1 Elements panel
- 6.1.2 Console panel
- 6.1.3 Network panel
- 6.1.4 Debugging layout and JavaScript issues

6.2 Git Version Control
- 6.2.1 Introduction to version control
- 6.2.2 Git workflow (init, add, commit, push)
- 6.2.3 Pull requests and collaboration
- 6.2.4 Writing a Markdown README file

6.3 Deployment
- 6.3.1 Static website hosting options
- 6.3.2 Mapping assets and file structure
- 6.3.3 Cache Busting Basics

Unit VII: Localisation Basics

7.1 Language and Directionality
- 7.1.1 Importance of localisation in web design
- 7.1.2 Language and cultural considerations
- 7.1.3 Directionality (LTR and RTL text)
- 7.1.4 Dzongkha labels and content integration

7.2 Content Localisation Workflow
- 7.2.1 Managing text strings and labels
- 7.2.2 Formatting dates and times for different regions
- 7.2.3 Basic localisation workflow for websites

List of Practicals

i. Semantics and Validity: Build a multi-section page with landmarks; validate markup.
ii. Accessible Forms: Implement labelled inputs, error messages, and helpful hints.
iii. Flexbox Layout: Create a responsive header/nav/content/footer pattern.
iv. Grid Gallery: Design a responsive grid for cards/images with hover states.
v. Intro JS: Add simple interactivity—toggles, form checks, and DOM updates.
vi. Accessibility Audit: Run a contrast/focus/keyboard check; remediate issues.
vii. DevTools Clinic: Fix layout bugs; inspect network requests for assets.
viii. Git and Deploy: Publish the site; submit repo link, deployment URL.

Reading List

Essential Reading

Dipierro, M. (2013). *WEB2PY* (5th ed.). Experts4solutions.
Duckett, J. (2010). *Beginning HTML, XHTML, CSS, and JavaScript.*
Kogent Learning Solutions Inc. (2015). *HTML 5 in simple step*. Dreamtech Press.

Robson, E., & Freeman, E.T. (2014). *Head First JavaScript Programming.* O'Reilly Media.
Wang, P., & Katila, S. (2003). *An introduction to Web Design and Programming* (1st ed.). Prentice Hall.

Additional Reading

Chacon, S., & Straub, B. (2014). Pro Git (2nd ed.). Apress. https://git-scm.com/book/en/v2
Kantor, I. (n.d.). The modern JavaScript tutorial. https://javascript.info/ W3C Web Accessibility Initiative. (2024). WAI tutorials. https://www.w3.org/WAI/tutorials/ Google. (n.d.). web.dev – Learn forms. https://web.dev/learn/forms/
Mozilla Developer Network. (n.d.). *HTML: HyperText markup language.* MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/HTML
Mozilla Developer Network. (n.d.). *CSS: Cascading style sheets.* MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/CSS
Mozilla Developer Network. (n.d.). *JavaScript guide.* MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide
World Wide Web Consortium (W3C). (2023). *Web content accessibility guidelines (WCAG) 2.2.* https://www.w3.org/TR/WCAG22/
WHATWG. (n.d.). *HTML living standard.* https://html.spec.whatwg.org/Google. (n.d.). *web.dev – Learn responsive design.* https://web.dev/learn/design/

Date: May 2026

Module Code and Title: **MTH101 Discrete Mathematics**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor(s): Hari Kumar Tiwari and Namgay Tshering
Module Coordinator: Hari Kumar Tiwari

General Objective
The module aims to equip students with a solid foundation in the principles of discrete mathematics, enabling them to analyse and solve problems which are essential for Information Technology programme.

Learning Outcomes
On completion of the module, students will be able to:
i. distinguish different types of functions and their graphs
ii. solve the equations up to degree 3
iii. use suitable Graph Theory to solve specific computational problems
iv. identify different types of sets and their applications
v. solve the problems on relations and logics
vi. apply different types of matrices in problem solving
vii. determine the Eigen roots and Eigen vectors of a given square matrix
viii. use matrix method and determinants method to solve linear equations.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	**60**
	Class Activities	2	
Independent Study	Self-study	2	**60**
	Written Assignment	1	
	VLE Discussion	1	

Total	**8**	**120**

Assessment Approach

A. Matrices in IT (15%)

Students in groups of 4-5 will work and present their findings on the tools related to *Matrices and Determinants*. Their work should highlight on the connections of the tools applied in other modules. Each group is expected to compile their work into a clear and concise report, written within 350–˜400 words. In addition to the written submission, each student will participate in a viva or class presentation, which will be used to evaluate the originality and depth of their work. From this assignment, students should be able to convert word problems into matrix form and solve using matrix methods

Task I: Report Writing (10%)

3 Marks	Content Quality (accuracy, relevance, balance of fact and opinion and evidence-based)
2 Marks	Language (formal tone, clarity, conciseness, audience awareness)
2 Marks	Formatting and Presentation (consistency, visual, numbering and heading and proofreading)
2 Marks	Structure (title page, table of contents, introduction, body, conclusion, recommendation and reference)

Task II: VIVA-VOCE (5%)

The viva-voce will be for 10 marks and converted to 5%.

5 Marks	Subject Knowledge (Precise, comprehensive, well-supported answers)
3 Marks	Communication (Clear, structured, fluent responses)
2 Marks	Confidence and defence (Calm, assertive, effective handling of questions)

Task III: Presentation (5%)

The presentation will be for 10 marks and converted to 10%.

2 Marks	Visual Design (consistency, clarity, visual aids and cognitive ease)
5 Marks	Delivery and style (Confident and body language, voice modulation, timing)
3 Marks	Audience engagement (know your audience, interactive elements and relatable language)

B. Problem Solving (10%)

Students will individually apply the problem-solving techniques taught in class from Unit I and Unit III. The specific dates for these problem-solving sessions will not be announced in advance. Each student will complete two problem-solving exercises, worth 15 marks each and the marks obtained will be converted into 5%. The problem solving will have a duration of 45 minutes per exercise.

C. Class Tests (15%)

Each student will individually complete two class tests, consisting of application-based problems drawn from the theoretical concepts taught in class from Unit II and Unit IV. Each test will carry 20 marks and the marks obtained will be converted into 7.5%, and students will be notified in advance of the date and time. The written test will have a duration of 45 minutes.

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours and converted to 20%.

E. Semester-End Examination (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours and converted to 40%.

Overview of the assessment approaches and weighting:

Areas of Assessments	Quantity	Weighting (%)
A. Group Assignment	1	15
B. Problem Solving	2	10
C. Class tests	2	15
D. Mid-Semester Examination	1	20
E. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: None

Subject Matter

Unit I: Functions and Graphs

1.1 Introduction to the concept of Functions and Equations
1.2 Types of Functions and Equations
- 1.2.1 Linear Functions
- 1.2.2 Non-Linear Functions
- 1.2.3 Linear Equations
- 1.2.4 Quadratic Equations
- 1.2.5 Polynomial of Degree 3
- 1.2.6 Rational Equations

1.3 The Graph Theory
- 1.3.1 Basic terminology
- 1.3.2 Multi graphs
- 1.3.3 Weighted graphs
- 1.3.4 Paths and circuits
- 1.3.5 Shortest paths in weighted graphs
- 1.3.6 Eulerian and Hamiltonian paths and circuits
- 1.3.7 The traveling salesman problem
- 1.3.8 Factors of a graph and Planar graphs

Unit II: Sets

1.1 Introduction to Sets
1.2 Types of Sets
- 1.2.1 Set, Element, Membership
- 1.2.2 Subsets, Proper Subsets
- 1.2.3 Power Set (Set of all subsets)
- 1.2.4 Universal Set and Complement Set

1.3 Set Operations
- 1.3.1 Union and Intersection

1.3.2 Differences and Symmetric Difference
1.3.3 Cartesian Product
1.3.4 Disjoint Sets
1.3.5 Venn Diagrams

1.4 Functions as Sets
1.4.1 Function definition as a set of order pairs
1.4.2 Domain, Range, Codomain, Image, Preimage
1.4.3 Injective (One-to-One), Subjective (Onto), Bijective
1.4.4 Inverse Functions

Unit III: Relations and Logics

a. Introduction to Relations
3.1.1 Basic Concepts
3.1.2 Relation on a set A (subset of AxA) or between Set A and B
3.1.3 Graph interpretation: direct edges

b. Special kind of Relations
i. Reflexive, Symmetric, Antisymmetric, Transitive
ii. Equivalence Relations: Partition of the set into equivalence classes
iii. Partial Orders and Total Orders
iv. Functional Relations (functions as special relations)

c. Introduction to mathematical logic
i. Statements and notations
ii. Connectives
iii. Normal forms
iv. Theory of inference for the statement calculus
v. Inference theory of the Predicate calculus

Unit IV: Matrices and Determinants

a. Introduction to Matrices and Determinants
b. Types of Matrices
4.2.1 Symmetric and Skew symmetric
4.2.2 Hermitian and Skew Hermitian
4.2.3 Orthogonal and Unitary matrices
c. Determinants
d. Characteristic Equation and Cayley-Hamilton theorem (Proof not required)
e. Rank of a matrix
f. Eigen values and Eigen vectors
g. Minor, Cofactor, Adjoint and Inverse of Matrix
h. Solution of a system of linear equations
4.8.1 Matrix Method and
4.8.2 Determinant Method up to 3 variables

Reading List

Essential Reading

Kreyszig, E. (2011). *Advanced engineering mathematics.* Wiley.

Stallings, W. (n.d.). *Computer organization and architecture* (7th ed.). Pearson Education Asia.

Tremblay, J. P., & Manohar, R. (2005). *Discrete mathematical structures with applications to computer science* (23rd reprint). Tata McGraw-Hill.

West, D. B. (2002). *Introduction to graph theory* (2nd ed.). PHI Learning Pvt. Ltd.

Additional Reading

Balakrishnan, V. K. (2012). *Introductory discrete mathematics*. Courier Corporation.
Bradley, T. (2013). *Essential mathematics for economics and business* (4th ed.). Wiley.
Halmos, P. (2009). *Naive set theory*. Springer.
Lial, M. L., Greenwell, R. N., & Ritchey, C. D. (2021-2022). *Finite mathematics and calculus with applications* (11th ed.). Pearson.
Rosen, K. H. (2007). *Discrete mathematics and its applications* (6th ed.). Tata McGraw-Hill Companies.

Date: May 2026

Module Code and Title: **HCI101 Mindful Technology and Digital Wellbeing**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Dr. Kezang Sherab and Dr. Sonam Chuki
Module Coordinator: Dr. Kezang Sherab

General Objective
The module aims to develop students' critical awareness of the social, psychological, and ethical implications of digital technologies, while cultivating mindful and responsible engagement with technology through the principles of Gross National Happiness (GNH), mindfulness practices, and the Four Immeasurables of Buddhism. It seeks to equip students with practical strategies, reflective skills, and ethical frameworks that enable them to use, design, and promote digital technologies in ways that enhance individual wellbeing, community harmony, and sustainable societal development.

Learning Outcomes
On the completion of the module, students will be able to:
i. explain the philosophical foundations of Gross National Happiness and its relevance to technology use and development
ii. analyse the psychological, social, and ethical impacts of digital technologies on individuals, education systems, and society
iii. identify the risks and negative consequences of excessive or unmindful technology use
iv. apply the Four Immeasurables to guide ethical and responsible behaviour in digital environments
v. apply practical strategies for mindful and balanced technology use in personal, academic, and professional contexts
vi. evaluate digital technologies and platforms using ethical and wellbeing-oriented frameworks consistent with GNH values
vii. design technology solutions or initiatives that promote digital wellbeing, ethical engagement, and societal harmony
viii. demonstrate reflective awareness of personal digital habits and implement strategies to cultivate healthier relationships with technology
ix. model responsible digital citizenship and mindful technology practices within academic, professional, and community settings.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lecture	2	60
	Class activities	2	

Independent study	Applied Assignments and Reflective Practice	3	45
	Research, Reading, and Data Analysis	1	15
Total		**8**	**120**

Assessment Approach

A. Individual Reflective Journal (30%)

Students will individually maintain a monthly reflective journal throughout the semester. At the end of each month, they are expected to write one entry (minimum 200–300 words) and submit it for the tutor's comments (these entries will not be assessed). These monthly entries will be compiled and synthesised into a final reflective report at the end of the semester.

In their journal, students' will reflect on their personal experiences and insights related to:

- Mindfulness practices (e.g., breath awareness, gratitude, presence)
- Emotional intelligence (e.g., self-awareness, empathy, regulation)
- Digital habits (e.g., screen time, online behaviour, tech-life balance)

Their reflections should demonstrate how these practices help them manage the challenges of digital technology and support their growth as an ethical digital citizen. The journal is not just descriptive; it should show how they connect these practices to their learning and personal development over time, culminating in a comprehensive final report.

Grading Criteria and Parameters

8 Marks	Depth of reflection (Insightful, analytical, deep self-awareness, and critical thinking supported by examples or evidence)
8 Marks	Connection to module themes (Explicit references to concepts, theories, or case studies from the module)
5 Marks	Consistency and timeliness (All entries timely, consistent effort)
5 Marks	Writing quality and structure (Excellent clarity, structure, and mechanics)
4 Marks	Growth over time (Mindfulness practice, change of bad habits, progression, clarity of writing)

B. Case Study Analysis (30%)
Task Description

Students will work in groups of 4 to 5 to analyse a case study of a city project that integrates mindfulness, wellbeing, or sustainable digital futures. Options include, but not limited to:

i. Bhutan's Gelephu Mindfulness City (GMC)
ii. Songdo International Business District (South Korea) – smart sustainable city
iii. Masdar City (UAE) – eco-friendly innovation hub
iv. Auroville (India) – intentional community focused on wellbeing and sustainability
v. Singapore Smart Nation initiative – digital wellbeing and governance focus

Deliverables:

a. Written Report of 2000–2500 words (30%) and
b. Group presentation (20%).

Note: This assignment carries 50% of the overall module assessment, which will be converted to 30% of the final grade.

Key Points to Cover:

- Background and vision of the city/project.
- How mindfulness, wellbeing, or sustainability are embedded in its design.
- Connections to broader values (e.g., GNH pillars, cultural identity, environment, governance, community vitality).
- Implications for mindful technology and sustainable digital futures.

Grading Criteria and Parameters for Write-up

6 Marks	Relevance and understanding (Excellent grasp of project background and vision; supported by contextual examples)
8 Marks	Analysis and insight (Deep, originality of analysis, clear insights into wellbeing/mindfulness/sustainability)
8 Marks	Integration of module themes (Strong, explicit connections to multiple module themes)
3 Marks	Collaboration and presentation (Excellent teamwork; clear and engaging presentation)
3 Marks	Use of evidence and reflection (Examples, references, or reflective insights, and acknowledges data gaps or methodological constraints)
2 Marks	Language (Clear, coherent, and well-organised presentation; accurate grammar, appropriate academic vocabulary, and effective oral communication.)

Grading Criteria and Parameters for 15min. Group Presentation

4 Marks	Relevance and understanding (Accurate background explanation, contextualisation, vision articulation, relevance links)
5 Marks	Analysis and insight (Original, thoughtful analysis; sharp interpretation)
4 Marks	Integration of module themes (Explicit links to wellbeing, ethics, futures, etc.)
3 Marks	Collaboration and presentation (Balanced teamwork; engaging, professional delivery)
2 Marks	Use of evidence and reflection (Strong references; critical reflection on limits)
2 Marks	Language (Clear and fluent verbal communication; accurate grammar, appropriate vocabulary, correct pronunciation, and effective expression of ideas.)

C. Designing a Framework for Mindful and Responsible Technology Use (40%)

Task Description

Students will work in groups of 4-5 to design a framework or set of guidelines for the mindful and responsible use of technology in society. The assignment should focus on how technology can be harnessed for human-centred, sustainable, and ethical futures.

While GNH, Gelephu Mindfulness City, mindfulness practices, wellbeing, and emotional intelligence may be used as *approaches or lenses*, the central focus must remain on technology— its risks, negative impacts, and strategies for responsible use.

Deliverable:

- Group Report (2500 – 3000 words). This will be out of 40 marks and converted to 20%.
- Group presentation (20%). This will be out of 30 marks and converted to 20%.

Note: This assignment carries 70% of the overall module assessment, which will be converted to 40% of the final grade.

Key Points to Cover:

a. Technology Challenges: Identify key risks and negative impacts of technology (e.g., digital addiction, misinformation, surveillance, erosion of empathy).
b. Human-Centred Approaches: Show how mindfulness, wellbeing, emotional intelligence, and GNH principles can guide responsible technology use.
c. Framework Design: Propose actionable guidelines, policies, or practices for individuals, institutions, and governments.
d. Case References: Draw examples from Bhutan (e.g., GMC, GNH) and global initiatives (e.g., smart cities, digital wellbeing projects).
e. Future Implications: Discuss how mindful technology use can shape sustainable, harmonious societies.

Grading Criteria and Parameters for Written Report

9 Marks	Focus on technology (Technology stays central; risks and solutions clear)
10 Marks	Analytical depth and insight (Original, sharp analysis; meaningful insights)
9 Marks	Framework design and practicality (Framework is innovative, comprehensive, and workable)
5 Marks	Integration of module themes (Module themes woven in without overshadowing technology)
4 Marks	Collaboration and presentation (Balanced teamwork; polished, engaging delivery)
3 Marks	Language (Clear, coherent, and well-organised presentation; accurate grammar, appropriate academic vocabulary, and effective oral communication.)

Grading Criteria and Parameters for a 30 min. Group Presentation

7 Marks	Relevance and understanding (Explains project story with clear context & vision)
8 Marks	Analysis and insight (Deliver sharp, original analysis with meaningful insights)
7 Marks	Integration of module themes (Connects explicitly to multiple module)
3 Marks	Collaboration and presentation (Show balanced teamwork; engaging, professional delivery)
3 Marks	Use of evidence and reflection (Uses strong references; reflects critically on data limits)
2 Marks	Language (Clear and fluent verbal communication; accurate grammar, appropriate vocabulary, correct pronunciation, and effective expression of ideas.)

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Individual Reflective Journal	1	30
B. Case Study Analysis- write-up and group presentation	2	30
C. Designing a Framework for Mindful and Responsible Technology Use- write-up and group presentation	2	40
Total		**100**

Pre-requisites: None

Subject Matter

Unit I: Introduction to Technology, Wellbeing, and Gross National Happiness

1.1 Concepts of wellbeing in the digital age
1.2 Overview of Gross National Happiness and its four pillars
1.3 Relationship between technology, happiness, and societal progress

1.4 Emerging concepts of happiness in digital societies
1.5 Digital transformation and its impacts on Bhutanese society

Unit II: Digital Technology and the Human Mind

2.1 Psychology of digital technology use
2.2 Attention economy and persuasive technology
2.3 Cognitive overload and information fatigue
2.4 Social media addiction and behavioural design
2.5 Neuroscience of digital engagement

Unit III: Influence of Digital Advancement

3.1 Digital distraction and reduced concentration
3.2 Mental health issues: anxiety, depression, digital burnout
3.3 Online harassment, cyberbullying, and digital toxicity
3.4 Impacts on relationships, culture, and community
3.5 Misinformation, technology dependence, and digital imbalance
3.6 Strategies to overcome ill-effects of technology

Unit IV: Foundations of Mindfulness and Contemplative Practices

4.1 Concepts of mindfulness and awareness
4.2 Mindfulness in Buddhist philosophy and modern psychology
4.3 Basic meditation techniques for focus and clarity
4.4 Emotional intelligence: self-awareness, empathy, regulation, social skills.
4.5 Mindful attention in learning and digital use
4.6 Contemplative practices for mental health and productivity
4.7 Journaling Mindfulness and Contemplative Practices

Unit V: The Four Immeasurables and Ethical Technology Engagement

5.1 Loving kindness (*jampa*) in digital interactions
5.2 Compassion (*nying jey*) and ethical responsibility in technology design
5.3 Empathetic joy (*gawa*) in collaborative digital communities
5.4 Equanimity (*tang nyom*) and emotional regulation in online environments
5.5 Integrating moral values into technology practices
5.6 Guided practices and journaling

Unit VI: Mindful Technology Practices for Personal Wellbeing

6.1 Digital minimalism and intentional technology use
6.2 Managing screen time and digital habits
6.3 Mindful communication in digital spaces
6.4 Growth mindset and academic resilience with technology
6.5 Digital tools for enhanced productivity and focus
6.6 Digital detox strategies and reflective journaling
6.7 Developing personal digital wellbeing plans

Unit VII: Technology, Ethics, and GNH-oriented Innovation

7.1 Ethical frameworks for technology development
7.2 Ethical responsibilities and responsible computing
7.3 Technology for social good and wellbeing
7.4 Designing digital tools aligned with GNH principles
7.5 Case studies of ethical and mindful technological innovations

Unit VIII: Building a Culture of Mindful Technology

8.1 Managing time and boundaries in study and personal life
8.2 Promoting digital wellbeing in educational institutions
8.3 Technology policies that support wellbeing

8.4 Role of technologists in shaping humane digital futures
8.5 Community awareness and responsible digital citizenship
8.6 Designing mindful technology initiatives
8.7 Building sustainable digital habits
8.8 Conscious technology and mindful societies (E.g. Gelephu Mindfulness City)

Reading List

Essential Reading

Al-Mansoori, R. S., Al-Thani, D., & Ali, R. (2023). Designing for digital wellbeing: From theory to practice a scoping review. *Human Behaviour and Emerging Technologies*, *2023*(1), 9924029.

Cecchinato, M. E., Rooksby, J., Hiniker, A., Munson, S., Lukoff, K., Ciolfi, L., ... & Harrison, D. (2019, May). Designing for digital wellbeing: A research & practice agenda. In *Extended abstracts of the 2019 CHI conference on human factors in computing systems* (pp. 1-8).

Newport, C. (2019). *Digital minimalism: Choosing a focused life in a noisy world*. Portfolio/Penguin.

Salamida, M. (2023). *Designing meaningful future digital interactions: Fostering wellbeing to reduce techno-induced stress* (Doctoral dissertation). Politecnico di Milano.

Vinh, T.H. (2022). *The culture of happiness: How to scale up happiness from people to organisations.* Parallel Press.

Additional Reading

Broom, D. (2022, February 15). *6 trends that define the future of health and wellness* [Web article]. World Economic Forum. https://www.weforum.org/stories/2022/02/megatrends-future-health-wellness-covid19/

Cecchinato, M., E., Munson, S., Thieme, A., Rooksby, J., Lukoff, K., Harrison, D., Hiniker, A., & Ciolfi, L. (2019). Designing for digital wellbeing: A research & practice agenda. In *Proceedings of CHI '19: CHI Conference on Human Factors in Computing Systems (CHI '19), May 04, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, https://doi.org/10.1145/3290607.3298998

Dalai Lama, & Cutler, H. C. (2009). *The art of happiness: A handbook for living*. Riverhead Books.

World Economic Forum. (2025). *Thriving workplaces: How to improve workplace wellbeing—and why it matters* [Insight report]. World Economic Forum. https://www.weforum.org/publications/thriving-workplaces-how-employers-can-improve-productivity-and-change-lives/

World Economic Forum. (n.d.). *The future of personalized well-being* [Project brief]. World Economic Forum. https://www.weforum.org/projects/precision-consumer-2030/

Ura, K., & Galay, K. (2004). *Gross national happiness and development.* The Centre for Bhutan Studies.

Date: March 2026

Year 1 Semester II

Module Code and Title: **CCS102 Operating Systems**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Phuntsho Wangdi

General Objective

This module intends to build students' understanding of operating systems by introducing core design concepts and resource management strategies. It will enable them to evaluate different system types and appreciate how operating systems shape software behaviour and performance, preparing them to develop applications that interact efficiently with underlying systems.

Learning Outcomes

On completion of the module, students will be able to:

i. explain the role and responsibilities of an operating system, including its services and interfaces
ii. evaluate CPU scheduling strategies and effects on system performance
iii. assess concurrency issues and its appropriate synchronisation techniques
iv. apply strategies to prevent, avoid, or resolve the potential deadlocks
v. assess the impact of different memory allocation and replacement strategies
vi. explain file system organisation, including different approaches to managing file storage
vii. assess I/O and storage management mechanisms, including their impact on system performance
viii. interpret OS behaviour using Linux tools and relate observations to system operations.

Learning and Teaching Approach

Type	**Approach**	**Hours per week**	**Total credit hours**
Contact	Interactive Lectures	3	60
	Computer Laboratory Session	1	
Independent Study	Assignment and Research	3	60
	VLE Discussion	1	
Total		**8**	**120**

Assessment Approach

A. Class Test (10%)

This is a written test conducted in class, with a duration of 40 minutes. Two such tests will be held, the first before the midterm, covering topics from Units I-III, and the second after the midterm, covering topics from Units IV-VII. Each test will be out of 20 marks.

B. Assignment – Operating System Process Management (10%)

Individual students are required to submit one assignment (800-1000 words), weights of 10%, related to the Process Management units, to earn this 10% of the final mark. The assignment, due before the midterm, will focus on a topic from Process Management.

Activity**:** Students are expected to write about a particular process management mechanism implemented in a popular operating system. Assignments will be evaluated based on the following criteria:

3 Marks Description of techniques (Clarity and accuracy in explaining the technique)

3 Marks Identification of Practical applications (How the technique is applied in real operating systems)
4 Marks Completeness (Clarity, depth of the work and coverage)

C. Linux Case Study Report (20%)

Students in groups (4–5 students) will complete a written case study analysis (1,200–1,500 words). Students will investigate system behaviour, processes, memory, file systems, I/O, or other relevant topics using appropriate commands or tools to observe, measure, and interpret system activity. And accordingly prepare a report on their investigation. The marking will be based on the following criteria;

6 Marks Description of Techniques (Clarity and accuracy in explaining methods used)
6 Marks Application of Techniques (Correct use of tools and relevance of collected evidence)
4 Marks Completeness (Coverage of all required sections and content)
2 Marks Analysis and Evidence (Insightful discussion and correct interpretation of results)
2 Marks Submission (Clear structure, readability, and inclusion of supporting screenshots/outputs)

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours.

E. Semester-End Examination (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Class Test	2	10
B. Assignment	1	10
C. Linux Case Study Report	1	20
D. Mid-Semester Examination	1	20
E. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: PGL102 Object-Oriented Programming, CCS101 Digital Logic and Computer Organisation

Subject Matter

Unit I: Operating System Overview

1.1. Definition and Goals of Operating Systems
1.2. Two Views of an Operating System
    1.2.1. User view
    1.2.1. System view
1.3. Evolution and Types of Operating Systems
    1.3.1. Batch systems
    1.3.2. Time-sharing systems
    1.3.3. Real-time systems
    1.3.4. Distributed systems
1.4. Operating System Structures
    1.4.1. Monolithic structure
    1.4.2. Layered structure
    1.4.3. Microkernel architecture

- 1.5. System Calls
  - 1.5.1. Purpose of system calls
  - 1.5.2. Examples of system calls
- 1.6. Shell and System Programmes
  - 1.6.1. Role of the shell
  - 1.6.2. Examples of system programmes

Unit II: Process and Thread Management

- 2.1 Processes
  - 2.1.1. Process concept
  - 2.1.2. Process states
  - 2.1.3. Process Control Block (PCB)
  - 2.1.4. Context switching
- 2.2. Threads
  - 2.2.1. User-level threads
  - 2.2.2. Kernel-level threads
  - 2.2.3. Benefits and trade-offs
- 2.3. CPU Scheduling Concepts
  - 2.3.1. Scheduling criteria
  - 2.3.2. Scheduling principles
- 2.4. Scheduling Algorithms
  - 2.4.1. First-Come First-Served (FCFS)
  - 2.4.2. Shortest Job First (SJF)
  - 2.4.3. Priority scheduling
  - 2.4.4. Round Robin scheduling
  - 2.4.5. Comparative reasoning of algorithms

Unit III: Concurrency and Synchronisation

- 3.1. Critical Section Problem
  - 3.1.1. Race conditions
  - 3.1.2. Critical section requirements
- 3.2. Synchronisation Mechanisms
  - 3.2.1. Mutex locks
  - 3.2.2. Semaphores
  - 3.2.3. Monitors
- 3.3. Inter-Process Communication (IPC)
  - 3.3.1. IPC concepts
  - 3.3.2. Producer-consumer problem
  - 3.3.3. Readers-writers problem

Unit IV: Deadlocks

- 4.1. System Model
  - 4.1.1. Resource allocation graphs
- 4.2. Deadlock Concepts
  - 4.2.1. Deadlock conditions
- 4.3. Deadlock Handling Strategies
  - 4.3.1. Deadlock prevention
  - 4.3.2. Deadlock avoidance
  - 4.3.3. Deadlock detection

Unit V: Memory Management

- 5.1. Memory Management Goals
- 5.2. Virtual Memory Concepts
  - 5.2.1. Paging fundamentals
  - 5.2.2. Virtual memory concepts.

- 5.3. Address Translation
  - 5.3.1. Page tables
  - 5.3.2. Translation Lookaside Buffer (TLB)
  - 5.3.3. Page faults
- 5.4. Page Replacement Algorithms
  - 5.4.1. FIFO
  - 5.4.2. LRU
  - 5.4.3. Optimal
  - 5.4.4. Clock / Second Chance
  - 5.4.5. Comparative reasoning
- 5.5. Locality Concepts
  - 5.5.1. Locality of reference
  - 5.5.2. Belady's anomaly

Unit VI: File Management

- 6.1. File Concepts
  - 6.1.1. File naming
  - 6.1.2. File attributes
  - 6.1.3. File operations
- 6.2. Directory Structures
  - 6.2.1. Directory organisation
  - 6.2.2. Implementation concepts
- 6.3. File Allocation Methods
  - 6.3.1. Contiguous allocation
  - 6.3.2. Linked allocation
  - 6.3.3. Indexed allocation/inodes
- 6.4. Free Space Management
  - 6.4.1. Bitmap approach
  - 6.4.2. Linked list approach

Unit VII: Device and Storage Management

- 7.1. I/O Hardware Concepts
  - 7.1.1. I/O devices and controllers
  - 7.1.2. Memory-mapped I/O
- 7.2. Interrupts and Direct Memory Access (DMA)
- 7.3. I/O Software Layers
  - 7.3.1. Device drivers
  - 7.3.2. Interrupt handlers
- 7.4. Disk Scheduling Algorithms
  - 7.4.1. FCFS
  - 7.4.2. SSTF
  - 7.4.3. SCAN / C-SCAN
  - 7.4.4. LOOK / C-LOOK
- 7.5. Disk Organisation
  - 7.5.1. Disk formatting concepts
  - 7.5.2. RAID overview

Unit VIII: Linux

- 8.1. Linux Overview
  - 8.1.1. Linux kernel
  - 8.1.2. Kernel modules
  - 8.1.3. User space
- 8.2. Linux Process Model
  - 8.2.1. Process management
  - 8.2.2. Scheduling concepts

8.3. Linux Memory and File Systems
    8.3.1. Memory management observations
    8.3.2. File system structure
8.4. Linux Tools for Observing OS Behaviour
    8.4.1. Process monitoring tools
    8.4.2. Memory monitoring tools
    8.4.3. File and I/O monitoring tools
8.5. Basic Shell Scripting

Reading List

Essential Reading

Deitel, H. M., Deitel, P. J., & Choffnes, D. R. (2003). *Operating systems* (3rd ed.). Prentice Hall.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2005). *Operating system concepts* (7th ed.). John Wiley & Sons.

Tanenbaum, A. S. (2008). *Modern operating systems* (3rd ed.). PHI Learning Pvt. Ltd.

Additional Books

Bovet, D. P., & Cesati, M. (2005). *Understanding the Linux kernel* (3rd ed.). O'Reilly Media.

Love, R. (2010). *Linux kernel development* (3rd ed.). Addison-Wesley Professional.

McKusick, M. K., Neville-Neil, G. V., & Watson, R. N. M. (2014). *The design and implementation of the FreeBSD operating system* (2nd ed.). Addison-Wesley Professional.

Stallings, W. (2018). *Operating systems: Internals and design principles* (9th ed.). Pearson.

Date: March 2026

Module Code and Title: **DMS101 Database Management System**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Ngawang Tenzin

General Objective

This module aims to introduce the principles of Database Management Systems, emerging technologies and cloud databases. Students will gain hands-on experience with MySQL/PostgreSQL and complete a mini project to apply concepts in real-world database applications.

Learning Outcomes

On completion of the module, students will be able to:

i. explain the concepts of data, databases and DBMS architecture
ii. describe the roles of database administrators, designers and end-users
iii. design Entity–Relationship (ER) models
iv. transform ER models into relational schemas
v. apply normalization techniques to improve database design
vi. demonstrate proficiency in SQL for data definition, manipulation and control
vii. implement views, triggers, stored procedures and transactions using SQL
viii. apply principles of transaction management, concurrency control and recovery.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory session	2	
Independent Study	Self-study	4	60
Total		**8**	**120**

Assessment Approach

A. **VLE Quiz (10%)**

A quiz will be conducted on the VLE platform after covering the first three units. The quiz will carry a total of 30 marks, have a duration of 30 minutes, and will consist of multiple-choice questions (MCQs) and short-answer type questions.

B. **Practical Examinations (20%)**

Two practical examination, one before the mid-semester and one after the mid-semester, of 10% each will be conducted to evaluate students' practical skills. Each exam will be out of 30 marks and convert to 10%. The grading will focus on:

3 Marks	Accuracy of solutions (Correct execution of tasks, valid outputs and proper use of commands/techniques)
3 Marks	Application of concepts (Ability to apply subject-specific principles)
4 Marks	Problem-solving and reasoning (Logical approach, clarity of steps, efficiency in designing solutions and structured methodology)

C. **Project (30%)**

Students will work in groups of 4–5 members to carry out a database project by following the database design process. It will be divided into three parts: Progress Report Presentation (10%), Final Report Presentation (10%) and Report Evaluation (10%).

i. **Progress Report Presentation (10%)**

Students are expected to identify a problem that requires a database system and present the initial design of the database. The presentation should briefly explain the problem, system requirements and a preliminary database design. Students should demonstrate their understanding of the entities, relationships and overall structure of the proposed database. Marks will be awarded based on following criteria

2 Marks	Problem Identification and Scope (Clarity of problem statement, relevance of the selected system, clear objectives and scope)
3 Marks	Requirement Analysis (Identification of users, entities and attributes, relationships between entities)
3 Marks	Initial Database Design (Preliminary ER diagram, correctness of entities and relationships, logical representation of the system)
2 Marks	Language (clarity of writing and grammar)

ii. **Final Report Presentation (10%)**

Students are expected to present the completed database design and demonstrate the implemented database system. The presentation should show the final design, implementation and how the system manages data effectively. Marks will be awarded based on following criteria:

4 Marks	Content (Database design, system implementation, functionality and queries)
2 Marks	Demonstration (System demonstration, explanation and outputs)
2 Marks	Delivery and Presentation (Confidence, clarity, visual organization and communication skills)
2 Marks	Presentation and Teamwork (Clarity of presentation, logical explanation, equal participation of members)

iii. **Final Report Evaluation (10%)**

Students are expected to submit a final project report that documents the database design and implementation process. The report should clearly describe the problem, design approach and implementation of the database system. Marks will be awarded based on following criteria:

2 Marks	Problem Description and Requirements (Clear background of the problem; system objectives; requirement description)
3 Marks	Database Design Documentation (ER diagram; relational schema; normalization explanation)
3 Marks	Implementation Details (Tables and keys; constraints; SQL queries and outputs)
2 Marks	Report Organization and Clarity (Logical structure; clarity of writing; formatting and referencing)

D. **Semester-End Examination (40%)**

The Semester-end Examination will be conducted for 100 marks over a duration of three hours.

Overview of the Assessment Approaches and Weighting

Areas of Assignments		**Quantity**	**Weighting**
A. VLE Quiz		1	10%
B. Practical Examinations		2	20%
C. Project	Progress Report Presentation	1	10%
	Final Report Presentation	1	10%
	Report Evaluation	1	10%
D. Semester-End Examination (SEE)		1	40%
Total			**100%**

Pre-requisites: None

Subject Matter

Unit I: Introduction and DBMS Architecture

1.1. Data, databases, DBMS, Database system, characteristics of DBMS
1.2. Roles: DBA, designers, end-users
1.3. Advantages and limitations of DBMS
1.4. DBMS architecture: three-schema model
1.5. Data independence, data models, schemas, instances
1.6. DBMS languages and interfaces
1.7. Client Server Architecture of Database system
1.8. Database Design Process

Unit II: Conceptual Data Modelling

2.1. Conceptual data models
2.2. Entity types, attributes, keys and weak entities
2.3. Relationships and constraints

2.4. ER diagrams and case studies
2.5. EERD model - specialisation and generalisation concepts

Unit III: Relational Model & Database Design

3.1. Relational data model
3.2. Keys (super, candidate, primary)
3.3. Relational algebra operations
3.4. Codd's rules
3.5. Integrity constraints
3.6. Mapping ER to relational schema

Unit III: Database Design Theory and Methodology

4.1. Functional dependencies (concept and types)
4.2. Anomalies in database design
4.3. Normalization (1NF, 2NF, 3NF)
4.4. Advanced normalization (BCNF, 4NF)

Unit IV: SQL and Implementation

5.1. SQL basics: CREATE, ALTER, DROP
5.2. DML: INSERT, UPDATE, DELETE, SELECT
5.3. Querying with WHERE, ORDER BY, GROUP BY, HAVING
5.4. Joins: equi, natural, outer, self-joins
5.5. Subqueries and set operations (UNION, INTERSECT, MINUS)
5.6. Views, indexes, permissions
5.7. Advanced SQL: triggers, stored procedures, functions
5.8. PL/SQL extensions

Unit V: Transactions & Concurrency Control

6.1. Transaction concepts & ACID properties
6.2. Concurrency issues: schedules, serialisability, recoverability
6.3. Lock-based protocols: shared, exclusive, two-phase locking
6.4. Deadlock handling
6.5. Backup and recovery mechanisms
6.6. Introduction to MapReduce

List of Practicals

i. Installing MySQL/PostgreSQL and creating a database.
ii. Using DDL to create tables and constraints.
iii. Using DML to insert, update, delete records.
iv. Querying data using SELECT, WHERE, ORDER BY, GROUP BY, HAVING.
v. Performing joins, subqueries and set operations.
vi. Creating views and indexes.
vii. Writing triggers, stored procedures and functions.
viii. Implementing transaction management (COMMIT, ROLLBACK).

Reading List

Essential Reading

Connolly, T., & Begg, C. (2015). *Database systems: A practical approach to design, implementation and management*. Pearson.

Elmasri, R., & Navathe, S. B. (2026). *Fundamentals of database systems* (8th ed.). Pearson.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database system concepts* (7th ed.). McGraw-Hill.

Additional Reading

Chaudhri, A. B., Rashid, A., & Zicari, R. (2003). *XML data management.* Addison-Wesley.

Date, C. J. (2004). An introduction to database systems. Addison-Wesley.

Ramakrishnan, R., & Gehrke, J. (2000). *Database management systems*. McGraw Hill.

Williams, K. et al. (2000). *Professional XML databases*. Wrox Press.

Date: May 2026

Module Code and Title: **PGL102 Object-Oriented Programming**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Arun Kumar Singh

General Objective

This module aims at developing students' ability to design and implement structured, modular, and maintainable software using object-oriented programming principles. Building on the programming knowledge gained in PGL101 Fundamentals of Programming, students will learn to model real-world problems.

Learning Outcomes

On completion of the module, students will be able to:

i. analyse the characteristics of object-oriented programming in comparison with procedural programming
ii. design classes and objects to represent real-world entities using C++
iii. apply encapsulation and abstraction to create modular program structures
iv. implement inheritance in order to extend existing classes and promote code reuse
v. implement polymorphism using virtual functions and abstract classes
vi. develop generic functions and classes using templates
vii. apply exception handling techniques to manage program errors
viii. implement file input and output operations to store and retrieve program data
ix. demonstrate professional programming practices including readable code structure and documentation.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory session	2	
Independent Study	Self-study	4	60
Total		**8**	**120**

Assessment Approach

A. Class Test (10%)

This is a written test conducted in class with a duration of 40 minutes. Two such tests will be administered during the semester from Unit I - III, and Unit VI and VII respectively. Each test will be marked out of 20 marks and will require students to answer short questions, perform code tracing, or solve programming-related problems.

B. Laboratory Practical Work (25%)

The laboratory assessment evaluates students' practical coding competence and consists of: Laboratory Report (5%), Practical Examination (15%) and Viva Voce (5%).

1. Laboratory Report (5%)

Students will complete a series of Ten programming tasks aligned to the units. Assignments will emphasise class design, modularity, safe programming habits, and correctness. The reports will be submitted via VLE every week.

2 Marks	Accuracy and correctness (compiles, runs, correct outputs, handles common edge cases)
3 Marks	Content (Application of OOP concepts, Problem-solving approach, coding)

2. Practical Examination (15%)

The Semester-End practical examination will be conducted in the laboratory where students solve one or more programming problems using classes, functions, and core OOP techniques. Mindset. It focuses on correctness and program structure.

6 Marks	Correctness of program output (functional correctness)
4 Marks	Use of appropriate techniques (classes, modular functions, correct control flow)
3 Marks	Structured reasoning (clean program structure, clear approach)
2 Marks	Code clarity and presentation (readable formatting and naming)

3. Viva Voce (5%)

An oral examination will be conducted during Practical Examination where students explain their code, design choices, and core OOP concepts.

2 Marks	Conceptual understanding (OOP concepts, class responsibilities, key C++ concepts used)
3 Marks	Clarity (confidence, clear reasoning and detail explanation to the solution)

C. Mid-Semester Examination (10%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours and converted to 10%.

D. Project (15%)

Students will develop a small OOP-based application in pairs. The project must demonstrate: class modelling and encapsulation, modularity, file I/O or exception handling (at least one), documentation and a short demonstration, readable, maintainable code with evidence of iterative development and basic testing mindset, for pair projects, each submission must include a brief contribution log. Individual viva may be used to confirm authorship and understanding, and team members may receive differing marks where contribution differs.

Assessment criteria:

5 Marks	Functionality and correctness (Features implemented correctly; stable behaviour; handles common cases)
4 Marks	Use of OOP concepts (Appropriate classes; encapsulation; sensible design decisions; coherent responsibilities)
3 Marks	Code quality (modularity, readability and maintainability)
3 Marks	Documentation and presentation (Clarity, appropriateness and completeness)

E. Semester-End Examination (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours and converted to 40%.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Class Test	1	10
B. Laboratory Report	10	5

Laboratory Examination Laboratory Viva Voce	1 1	15 5
C. Mid-Semester Examination	1	10
D. Project	1	15
E. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: PGL101 Fundamentals of Programming

Subject Matter

Unit I: Object-Oriented Programming Foundations and Design Thinking

1.1. Structured vs object-oriented approach
1.2. Encapsulation, abstraction, inheritance, polymorphism
1.3. Modelling real-world entities as classes
1.4. Basic design hygiene: cohesion, responsibility, naming

Unit II: C++ Fundamentals for Software Construction

2.1 Programme structure, tokens, data types, operators
2.2 Input/output streams (cin, cout)
2.3 Control statements and functions
2.4 Modularity, functions and types of functions
2.5 Intro to standard library containers

Unit III: Classes and Objects

3.1 Class definition, access specifiers
3.2 Constructors/destructors
3.3 Static members, passing/returning objects
3.4 Introductory “Ownership and lifecycle” concepts

Unit IV: Basics of Operator Overloading

4.1 Purpose and restrictions
4.2 Unary/binary overloading (selected operators)
4.3 Readability and appropriate use

Unit V: Inheritance

5.1 Base and derived classes, access modes
5.2 Types of inheritance and code reuse
5.3 Inheritance vs Composition

Unit VI: Polymorphism and Abstract Classes

6.1 Virtual functions and late binding
6.2 Abstract classes and pure virtual functions (interfaces)

Unit VII: Templates and Exception Handling

7.1 Function templates and class templates (foundations)
7.2 Importance of generic programming
7.3 Exception handling: try/throw/catch
7.4 Defensive coding habits and predictable failure behaviour

Unit VIII: File Handling

viii.1 Stream hierarchy
viii.2 Text file operations and structured storage basics
viii.3 Reading/writing records
viii.4 Error handling patterns for file operations

Unit IX: Project Development

9.1 Problem definition and class model
9.2 Implementation and iterative refinement
9.3 Documentation and demonstration preparation

List of Practicals

i. Writing a Basic C++ Program for Input and Output Operations
ii. Implementing Conditional and Iterative Statements in a C++ Program
iii. Developing Modular Programs Using Functions in C++
iv. Implementing Encapsulation Using Access Specifiers and Member Functions
v. Developing C++ Programs Using Operator Overloading
vi. Implementing Inheritance in a C++ Program
vii. Demonstrating Polymorphism Using Virtual Functions
viii. Developing Generic Programs Using Function and Class Templates
ix. Implementing Exception Handling in a C++ Program
x. Reading and Writing Data Using File Handling in C++ Programs

Reading List

Essential Reading

Lafore, R. (2014). *Object-Oriented Programming in C++* (4th ed.). SAMS Publications.

Schildt, H. (2013). *C++: The complete reference* (4th ed.). Tata McGraw Hill.

Stroustrup, B. (2014). *Programming: Principles and practice using C++* (2nd ed.). Addison-Wesley.

Additional Reading

Deitel, P., & Deitel, H. (2013). *C++ how to program* (3rd ed.). Pearson.

Lippman, S., Lajoie, J., & Moo, B. (2013). *C++ primer* (5th ed.). Addison-Wesley.

Savitch, W. (2015). *Problem solving with C++* (10th ed.). Pearson.

Date: May 2026

Module Code and Title: **MTH102 Calculus**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Hari Kumar Tiwari and Urmila Gurung
Module Coordinator: Hari Kumar Tiwari

General Objective

This module will enable the students to apply different concepts of calculus to problem solving and cultivates the mathematical thinking required for pursuing an information technology programme.

Learning Outcomes

On completion of the module, students will be able to:

i. solve cubic and bi-quadratic equations under certain conditions
ii. apply the concept of limit to check the continuity of functions
iii. determine derivatives up to third order
iv. distinguish between differentiation and partial differentiation
v. find the maximum and minimum values of a function involving up to two independent variables
vi. state and prove different theorems
vii. use different integration methods to various types of functions
viii. find out the values of a given definite integral
ix. apply different rules to solve numerical integrals

x. use different methods to find the solutions of numerical algebraic and transcendental equations.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	**60**
	Class activities	2	
Independent Study	Self-study	2	**60**
	Written Assignment	1	
	VLE Discussion	1	
Total		**8**	**120**

Assessment Approach

A. Assignment – Application of differential calculus (15%)

Students will work in groups of 4-5 to research and present their findings on the tools used in Differential Calculus. Their research outcomes must be compiled into a well-structured report, written clearly within a word limit of 350-400 words. In addition to submitting the report, each member from the group will participate in a viva or class presentation, which will serve to assess the originality and depth of their work. From this assignment, students should be able to analyse the problems and solve them using appropriate tools/ methods.

Task 1: Report writing (10%)

3 Marks	Content Quality (accuracy, relevance, balance of fact and opinion and evidence-based)
2 Marks	Language (formal tone, clarity, conciseness, audience awareness)
2 Marks	Formatting and Presentation (consistency, visual, numbering and heading and proofreading)
3 Marks	Structure (title page, table of contents, introduction, body, conclusion, recommendation and reference)

Task 2: VIVA (5%)

2 Marks	Subject Knowledge (Precise, comprehensive, well-supported answers)
3 Marks	Confidence and Communication (Clear, structured, fluent responses)

Task 3: Presentation (5%)

2 Marks	Visual Design (consistency, clarity, visual aids and cognitive ease)
3 Marks	Delivery and style (Confident and body language, voice modulation, timing)

B. Problem Solving (10%)

Two problem solving exercises where students will individually apply the problem-solving techniques taught in class from Unit I and Unit III. The specific dates for these problem-solving sessions will not be announced in advance. Each problem-solving exercises will be worth 15 marks and the marks obtained will be converted into 5%. The problem solving will have a duration of 45 minutes per exercise.

C. Class tests (15%)

Each student will individually complete two class tests, consisting of application-based problems drawn from the theoretical concepts taught from Unit II and Unit IV. Each test will

carry 20 marks and the marks obtained will be converted into 7.5%, and students will be notified in advance of the date and time. The written test will have a duration of 45 minutes.

D. Mid-Semester Examination (20%)
The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours.

E. Semester-End Examination (40%)
The Semester-end examination will be conducted for 100 marks over a duration of three hours.

Overview of the assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Group Assignment	1	15
B. Problem Solving	2	10
C. Class tests	2	15
E. Mid-Semester Examination	1	20
F. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites:

Subject Matter

Unit-I: Theory of Equations

1.1. Relation between the roots and coefficients
1.2. Solving cubic and bi-quadratic equations when some condition on the roots is given
1.3. Reciprocal equations
1.4. Transformation of equations

Unit-II: Functions, Limits and Continuity

2.1 Introduction to Functions
- 2.1.1 Types of Functions
- 2.1.2 Linear Functions
- 2.1.3 Non-Linear Functions
- 2.1.4 The Graph of a Linear and Non-Linear Functions

2.2 Limits and continuity,
- 2.2.1 Introduction to Limits
- 2.2.2 One-Sided Limits and Continuity
- 2.2.3 Fundamental properties of continuous function
- 2.2.4 Appling limit to check the continuity of functions

Unit-III: Differential Calculus

3.1 Introduction to Differentiation
- 3.1.1 Geometric meaning of derivative
- 3.1.2 Rules of differentiation
- 3.1.3 Differentiation of Algebraic Functions
- 3.1.4 Differentiation of Logarithmic and Exponential Functions
- 3.1.5 Successive differentiation up to third derivative
- 3.1.6 Local Maxima and Local Minima

3.2 Function of several variables
- 3.2.1 First order partial derivatives
- 3.2.2 Euler's theorem on homogeneous functions of two variables
- 3.2.3 Second order partial derivative
- 3.2.4 Maxima and Minima of functions of two independent variables

3.3 Theorems

3.3.1 Rolle's theorem
3.3.2 Mean value theorem
3.3.3 Maclaurin's theorems with Cauchy's and Lagrange's forms of reminder
3.3.4 Taylor's series

Unit-IV: Integral calculus

3.1 Introduction to Integration
- 3.1.1 Rules of integration of indefinite integrals
- 3.1.2 Integration of algebraic functions
- 3.1.3 Integration of Logarithmic and Exponential Functions
- 3.1.4 Solution of definite integrals.

4.2 Introduction to Numerical Analysis

4.3 Numerical Integration
- 4.3.1 Midpoint Rule
- 4.3.2 Trapezoidal Rule
- 4.3.3 Simpon's Rule
- 4.3.4 Runge-Kutta method

4.4 Solution of simultaneous linear algebraic equations
- 4.4.1 Gaussian elimination method
- 4.4.2 Gauss – Seidal method

4.5 Solution of numerical algebraic and transcendental equations
- 4.5.1 Regula – Falsi method
- 4.5.2 Newton – Raphson's method

Reading List

Essential Reading

Atkinson, K., & Kendall, E. (2005). *An introduction to numerical analysis*. John Wiley & Sons.

Kreyszig, E. (2011). *Advanced engineering mathematics* (Wiley India ed.). Wiley India.

Lial, M. L., Greenwell, R. N., & Miller, C. (1998). *Finite mathematics and calculus with applications* (5th ed.). Addison Wesley Longman.

Additional Reading

Apostol, T. M. (2005). *Calculus: Volume I*. John Wiley & Sons.

James, S. (2010). *Calculus: Concepts and contexts* (4th ed.). Richard Stratton.

Uspensky, J. V. (1948). *Theory of equations*. McGraw-Hill.

Date: May 2026

Module Code and Title: **EAP101 Intermediate English for Academic Purposes**
Programme: BA in English Studies
Credit: 12
Module Tutors: Mohan Rai, Ruma Tamang, Tenzin Wangmo
Module Coordinator: Mohan Rai

General Objective

EAP101 is the first part of a two-semester series that aims to develop abilities in reading, writing, listening, and speaking in an academic context to support students' learning through their degree studies. The module addresses basic language skills and builds on them to fit the academic context.

Learning Outcomes

On completion of the module, students will be able to:

i. identify grammatical features in context
ii. describe appropriate practices and standards in the academic context
iii. use appropriate grammatical structures to express academic ideas
iv. extract information from complex academic lectures, presentations, and discussions using various listening strategies
v. extract information from academic readings using reading strategies such as skimming, scanning, and in-depth reading
vi. compose a well-structured and coherent academic essay following appropriate conventions and styles
vii. use the current APA referencing style for common sources
viii. present academic content and ideas to an audience in formal and informal settings
ix. use academic vocabulary in written and spoken settings.

Learning and Teaching Approach

Type	Approach	Hours per Week	Total Credit Hours
Contact	Interactive Lecture	2	60
	Exercises, workshops, practice.	2	
Independent study	Writing assignments, VLE discussions	4	60
	Reading and review of class materials		
Total			**120**

Assessment Approach

A. Listening test: 15%

In a single sitting, 3 audio recordings of 3, 5, and 7 minutes respectively, will be played to the students. The 3 recordings may be of conversations, short lectures, or other appropriate material, and will be of increasing difficulty and complexity. Students will answer questions on a question paper simultaneously as they listen to the audio. The audio will not be replayed. The questions may be of multiple choice or short answer type, and will rely solely on information provided in the audio. Students will be given time in between sets of questions to check their answers and prepare for the next set.

B. Reading test: 15%

In a single sitting, students will read three passages of increasing difficulty and complexity. Each passage will be followed by a set of objective-type, short answer questions, or long answer questions on the respective passages, which will assess students' comprehension, critical thinking skills, grammar, and paraphrasing/summarising skills.

C. Grammar Test: 10%

The test will check students' fundamental understanding of grammar rules and their ability to apply them in context. It will cover topics such as sentence structure, verb tenses, subject-verb agreement, punctuation, and common grammatical errors. The test will provide enough context within the questions to avoid ambiguity, and to emphasise functional usage.

D. Presentation: 30%

Each student will make a 5-7 minutes presentation on a subject of their interest with relevant examples. The presentations will be assessed based on the following criteria:

Content: 20 marks
Use of academic standards: 30 marks
Use of visual aids: 15 marks
Language and delivery: 35 marks

E. Academic essay: 30%

Students will write a 600–750-word academic essay. The essay will follow basic academic standards discussed in the module and use current APA referencing for the source types taught. The assignment will be written in two drafts: the first draft will be worth 10; and the final draft will be worth 20, including 5 for improvement.

Both drafts will be evaluated using the following criteria:

Content: 20 marks Critical thinking: 20 marks Use of academic standards: 30 marks
Language and grammar: 30 marks
Improvement on feedback will be evaluated using the following rubric:
Marginal improvement: 0 – 49 marks
Satisfactory improvement: 50 – 59 marks
Significant and appropriate improvement: 60 – 74 marks
Significant improvement beyond feedback given: 75 – 100 marks

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Listening test	1	15
B. Reading test	1	15
C. Grammar Test	1	10
D. Presentation	1	30
E. Academic essay	1	30
Total		**100**

Pre-requisites: None

Subject Matter

The Subject Matter will be taught within the context of standard usage scenarios and examples from relevant texts. The skills, although divided amongst the different units, will not be taught in isolation. The EAP teaching team will maintain a Course Pack which is an Essential Reading Compilation of readings and practice materials, which they will revise and update regularly as per need. The Course Pack used for the semester should be endorsed by the Programme Management Committee.

Unit I: Grammar in context

1.1. Sentence construction: simple and compound sentences on academic topics
1.2. Subject-verb agreement
1.3. Complex sentences
1.4. Tenses review: identifying and using correct tenses in different situations
1.5. Prepositions and prepositional phrases
1.6. Articles

Unit II: Academic Language

2.1. Difference between general and academic English
2.2. Dos and don’ts of Academic English
2.3. Developing an academic vocabulary
2.4. Academic integrity and plagiarism

Unit III: Listening

3.1. Introduction to Academic Listening
3.2. Note-taking: outlining, mind-mapping, organizing notes
3.3. Listening for gist and detail
3.4. Active listening

Unit IV: Reading

4.1. Introduction to Academic Reading
4.2. Types of academic materials
4.3. Reading skills: skimming, scanning, close reading

4.4. Paraphrasing and summarising

Unit V: Writing

5.1. Understanding essay questions: instruction words and content words; BUG method
5.2. Essay structure
5.3. Building paragraphs
5.4. Writing process: pre-writing, writing, revising
5.5. Basic APA referencing: referencing books, journals, and websites
5.6. Paper layout and format

Unit VI: Speaking

6.1. Asking and answering questions
6.2. Giving opinions
6.3. Pronunciation
6.4. Presentation skills
6.5. Signposting language

Reading List

Essential Reading

(An Essential Reading compilation of the shorter reading materials will be made available to students).

American Psychological Association. (2020). *Publication manual of the American Psychological Association 2020: The official guide to APA style* (7th ed.). American Psychological Association.

Thaine, C. & McCarthy, M. (2014). *Cambridge academic English – An integrated skills course for EAP: B1+ (Upper Intermediate) student's book*. Cambridge University Press.

Wallwork, A. (2013). *English for academic research: Grammar, usage and style.* Springer.

Wallwork, A. (2022). *Giving an academic presentation in English.* Springer.

Date: July, 2025

Year 2 Semester I

Module Code and Title: Computer Organisation and Architecture
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Phuntsho Wangdi

General Objective

This module aims to provide students with a holistic understanding of the fundamental principles of computer architecture, enabling them to explain how major hardware components operate and contribute to overall system performance.

Learning outcomes

On completion of the module, students will be able to:

i. explain the fundamentals of computer architecture principles
ii. describe the structure of major computer system components
iii. analyse the execution of machine-level operations
iv. explain how data is stored in computer systems
v. describe input and output mechanisms in computer systems

vi. explain system interconnections in computer architecture
vii. analyse processing techniques that improve system performance
viii. apply quantitative methods to evaluate system efficiency
ix. discuss trends in modern computer system design.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Lecture Interactives	4	60
Independent Study	Assignments, Reading, reviewing materials and VLE Discussion	4	60
Total		**8**	**120**

Assessment Approach

A. Class Test (20%)

This is a written test conducted in class, for a duration of 40 minutes. Two such tests will be held, the first before the, covering unit I and Unit II. And the second after the, covering Unit III and Unit IV and the total will be converted to 20%.

B. Assignment (20%)

Students are required to submit one individual assignment consisting of two compulsory tasks, designed to assess both conceptual understanding and authenticity of work in Computer Architecture. The assignment will cover topics including computer architecture fundamentals, instruction execution, memory performance, CPU organisation, pipelining, and processor evaluation.

Task 1: Individual Analytical and Theoretical Report (15%)

Each student must individually prepare a 500–600 word written report that combines one analytical or numerical problem and one theoretical discussion question based on the core topics of Computer Architecture. The report should demonstrate clear problem-solving methodology, sound theoretical understanding and the ability to relate analytical results to architectural concepts. Each assignment will be assessed as per the criteria set below:

4 Marks	Conceptual Understanding (demonstrate accurate understanding of core architecture concept, show ability to explain concept clearly)
5 Marks	Analytical / Numerical Problem Solving (Correctness and clarity, including logical step and reasoning)
4 Marks	Theoretical Discussion (depth and relevance and theoretical discussion, demonstrate critical thinking)
2 Marks	Application and Integration (ability to relate analytical result to architectural concept)

Task 2: Individual Viva-Voce (5%)

Each student must individually attend a viva-voce based on the submitted report. This task is intended to verify the authenticity and originality of the work and to assess the student's conceptual understanding, clarity of explanation, and ability to justify solutions and arguments presented in the report.

4 Marks	Understanding of Analytical Solution (ability to clearly explain the numerical problem, solution step and result)
4 Marks	Conceptual Clarity (demonstrated sound understanding of theoretical concept discussed in the report)

4 Marks Authenticity and Ownership (confident, consistency with submitted report, and evidence of individual work)

C. Midterm Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours.

D. Semester-End exam (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours.

Overview of the assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Class Tests	2	20
B. Assignments	2	20
C. Midterm Examination	1	20
D. Semester-end Examination (SEE)	1	40
Total		**100**

Pre-requisites: CCS101 Digital Logic and Design

Subject Matter

Unit I: Introduction to Computer Architecture

1.1 Overview of Computer Architecture and Organization
1.2 Evolution of Computer Systems
1.3 Basic Functional Units of a Computer
1.4 Von Neumann Architecture
1.5 Performance Metrics and Benchmarking

Unit II: CPU Architecture

2.1 Structure of the CPU
2.2 Arithmetic Logic Unit (ALU)
2.3 Control Unit
2.4 Register Organization
2.5 Instruction Cycle
  2.5.1 Fetch Cycle
  2.5.2 Decode Cycle
  2.5.3 Execute Cycle
2.6 Micro-operations and Control Signals

Unit III: Instruction Set Architecture

3.1 Introduction to Instruction Set Architecture
3.2 Instruction Types
3.3 Instruction Formats
3.4 Addressing Modes
  3.4.1 Immediate Addressing
  3.4.2 Direct Addressing
  3.4.3 Indirect Addressing
  3.4.4 Register Addressing
3.5 Assembly Language Basics
3.6 RISC vs CISC Architecture

Unit IV: Memory Organization

4.1 Memory Hierarchy
4.2 Cache Memory
  4.2.1 Cache Mapping Techniques
  4.2.2 Cache Performance
4.3 Main Memory
4.4 Virtual Memory

4.5 Memory Management Techniques

Unit V: Input / Output Systems

5.1 Input and Output Devices
5.2 I/O Interface
5.3 Programmed I/O
5.4 Interrupt-Driven I/O
5.5 Direct Memory Access (DMA)

Unit VI: System Bus and Interconnection

6.1 Bus Structure
6.2 Types of Buses
6.2.1 Data Bus
6.2.2 Address Bus
6.2.3 Control Bus
6.3 Bus Arbitration Techniques
6.4 Interconnection Structures

Unit VII: Pipelining and Parallel Processing

7.1 Instruction Pipelining
7.2 Pipeline Stages
7.3 Pipeline Hazards
7.4 Superscalar Architecture
7.5 Parallel Processing Techni

Unit VIII: Performance Evaluation

8.1 CPU Performance Metrics
8.2 Speedup and Efficiency
8.3 Amdahl's Law
8.4 Performance Improvement Techniques

Unit IX: Modern Computer Architecture Trends

9.1 Multicore Processors
9.2 GPU Architecture
9.3 Cloud Computing Hardware
9.4 Energy Efficient Computing

Reading List

Essential Reading

Hwang, K. (2010). *Advanced computer architecture* (2nd ed.). McGraw-Hill Education.

Mano, M. M. (n.d.). *Computer system architecture*. Pearson Education.

Patterson, D. A., & Hennessy, J. L. (2012). *Digital design and computer architecture* (2nd ed.). Elsevier.

Stallings, W. (n.d.). Computer organization and architecture (7th ed.). Pearson Education Asia.

Additional Reading:

Englander, I. (n.d). *The architecture of computer hardware and systems software: An information technology approach* (3rd Edition). Wiley.

Godse, A.P., & Godse, D.A. (2010), *Computer organization and architecture. Technical.*

Hayes, J.P. (n.d). McGraw-Hill

Morris M.M., & Charles, K. (n.d). *Computer system architecture*. Pearson New International.

Date: May 2026

Module Code and Title: **DMS202 Database Programming**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Arun Kumar Singh

General Objective
This module aims at equipping students with advanced knowledge and practical skills in SQL programming and database-centric application development, while fostering expertise in performance tuning, transaction management and emerging database technologies.

Learning Outcomes
On completion of the module, students will be able to:

i. formulate complex SQL queries using advance SQL concepts
ii. perform data analysis using window functions and common table expressions
iii. develop stored procedures to implement business logic
iv. develop user-defined functions for database applications
v. implement triggers to enforce data integrity and audit changes
vi. apply transaction control and concurrency management techniques
vii. demonstrate query execution plans for database performance tuning
viii. implement structured error handling in database programs
ix. build a secure and functional database application for real-world scenarios
x. analyse XML and semi-structured data.

Learning and Teaching Approach

Type	Approach	Hours/Week	Total Hours
Contact	Interactive Lectures	2	30
	Laboratory Sessions	2	30
Independent Study	Assignment/Project	2	30
	Self-Study	2	30
Total		**8**	**120**

Assessment Approach

A. Class Tests (10%)
Two tests, one before midterm and one after midterm, of 20 marks each with a duration of 40 minutes will be conducted on the VLE. Test before midterm will be conducted after completing Unit II (Analytical SQL and Recursive Queries). The test after the midterm will include short, structured questions and applied scenario-based questions requiring students to draft, interpret, or correct SQL code fragments.

B. Mid-Semester Examination (15%)
The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours.

C. Practical Examination (15%)
The examination will be conducted in the laboratory for a duration of two hours at the end of the semester to assess students' practical competence in database programming and advanced SQL techniques covered in the module. The examination will require students to implement database objects and solve applied problems using SQL in a controlled laboratory environment.

The examination will evaluate the correctness of SQL statements, logical structuring of database code, appropriate use of database programmability features, and the ability to produce correct and efficient outputs from database queries. This will be out of 50 Marks and converted to 15%.

Assessment Criteria:

30 Marks	Coding (clarity, organsation, commenting, SQL queries, functions and outputs)
10 Marks	Application of database programming techniques (use of procedures, functions, triggers, and transaction control)
10 Marks	Problem-solving approach (ability to interpret the given scenario and implement suitable database solutions)

D. Project (30%)

Students will complete an individual database programming project that demonstrates a complete, secure, and documented database solution for a real-world scenario. The project should demonstrate the following components;

- a relational schema with constraints and appropriate indexing,
- advanced SQL queries (including at least one CTE and one window function),
- at least 2 stored procedures and 1 function,
- at least 2 triggers (e.g., audit logging + integrity enforcement),
- transaction scripts demonstrating consistent outcomes,
- brief performance tuning notes (indexing choices + query plan observations).

i. **Presentation/Demonstration and Viva (10%)**

The assessment consists of two components: Presentation/Demonstration (5%) and Viva Voce (5%). In the presentation and demonstration, each student will demonstrate the developed database system and explain key aspects of the project, including design choices, database programmability logic, integrity enforcement mechanisms, transaction behaviour, performance considerations, and secure practices.

3 Marks	Demonstration of functionality (clear walkthrough of the system and correct results)
2 Marks	Organisation and professionalism (well organized workflow, clarity in the materials and all requirements met)

The **viva voce** will assess the student's individual understanding of the submitted work. Students will answer questions related to the database design, SQL implementation, programmability objects, and system behaviour. The viva also supports academic integrity by confirming the student's authorship and understanding beyond the submitted code.

3 Marks	Conceptual understanding (knowledge of database design, procedures, ability to explain implementation decisions and system behaviour)
2 Marks	Confidence in responses (Audibility, Eye contact and Speech)

ii. **Report and Code Submission (20%)**

Students must submit a report and a complete, runnable SQL codebase

7 Marks	Technical completeness and correctness (Meet all required components; correct ɜviour of schema/queries/objects/transactions)
5 Marks	Code quality and robustness (Readable scripts, modular organisation, error ɖling where appropriate, consistent standards)
4 Marks	Performance evidence (Index rationale; basic execution plan interpretation; ningful improvements documented)
2 Marks	Documentation quality (clear structure; comments; setup instructions; professional ɹatting)
2 Marks	Security and integrity awareness (principle of least privilege; safe patterns; integrity rcement)

E. Semester-End Examination (30%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Class Tests	2	10
B. Mid-Semester Practical Examination	1	15
C. Final Practical Examination	1	15
D. Project	1	30
E. Semester-End Examination (SEE)	1	30
Total		**100**

Pre-requisites: DMS101 Database Management System

Subject Matter

Unit I: Advanced Querying Techniques

1.1. Deep dive into JOIN operations: CROSS APPLY, OUTER APPLY
1.2. Advanced Subqueries: Correlated subqueries, using EXISTS/ NOT EXISTS
1.3. Set Operations: UNION, INTERSECT, EXCEPT and their behaviour with duplicates
1.4. Advanced filtering with CASE statements within queries

Unit II: Analytical SQL and Recursive Queries

2.1. Common Table Expressions (CTEs): Non-recursive and Recursive CTEs
2.2. Window Functions: ROW_NUMBER(), RANK(), DENSE_RANK(), NTILE()
2.3. Analytical Functions: LAG(), LEAD(), FIRST_VALUE(), LAST_VALUE()
2.4. Advanced Aggregation: GROUPING SETS, ROLLUP and CUBE

Unit III: Stored Procedures and Programmable Logic

3.1. Stored Procedure fundamentals: Parameters (input, output, default values)
3.2. Control-of-Flow language: BEGIN...END, IF...ELSE, WHILE loops
3.3. Error Handling: Implementing TRY...CATCH blocks
3.4. Dynamic SQL: Construction and execution, with security considerations

Unit IV: Functions and Triggers

4.1. User-Defined Functions (UDFs): Scalar, Inline Table-Valued, Multi-Statement Table-Valued
4.2. Comparison of Functions vs. Stored ProceduresDML Triggers: AFTER Triggers (FOR), INSTEAD OF Trigger
4.3. Working with the inserted and deleted virtual tables
4.4. DDL Triggers for auditing schema changes
4.5. Managing trigger nesting and recursion

Unit V: Advanced Transaction Management and Concurrency

5.1. Transaction isolation levels (Read Uncommitted, Read Committed, Repeatable Read, Serializable, Snapshot)
5.2. Understanding concurrency phenomena: Dirty Reads, Non-Repeatable Reads, Phantom Reads
5.3. Implementing explicit locking hints (UPDLOCK, HOLDLOCK, etc.)
5.4. Advanced deadlock prevention and resolution strategies

Unit VI: Performance Tuning and Optimisation

6.1. Understanding and interpreting Execution Plans
6.2. Index architecture: Clustered vs. Non-Clustered Indexes
6.3. Advanced indexing strategies: Covering indexes, Filtered indexes

6.4. Query optimisation best practices and identifying performance bottlenecks
6.5. Database profiling and monitoring tools

Unit VII: XML and Semi-Structured Data

7.1. Introduction to XML
7.2. Key components of XML: DTD and XML Schema
7.3. Storing and querying XML data
7.4. Applications of XML in data transport and storage

Unit VIII: Distributed and Modern Databases

8.1. Distributed DBMS concepts
8.2. Data fragmentation, replication, and allocation
Types of distributed DBMS
8.3. Introduction to cloud databases
8.4. NoSQL database models: key-value stores, document databases, column-family databases, and graph databases

List of Practicals

i. Laboratory setup and exploration of advanced JOINs and subqueries.
ii. Implementing and using Common Table Expressions (CTEs).
iii. Solving analytical problems using Window Functions.
iv. Creating and executing Stored Procedures with parameters and control-flow logic.
v. Implementing robust error handling in Stored Procedures.
vi. Developing and testing different types of User-Defined Functions (UDFs).
vii. Creating DML Triggers for data auditing and integrity.
viii. Implementing INSTEAD OF Triggers on complex views.
ix. Experimenting with transaction isolation levels and concurrency.
x. Analysing and optimizing slow queries using Execution Plans.
xi. Designing and implementing effective indexes.

Reading List

Essential Reading

Ben-Gan, I. (2020). *T-SQL querying (Developer Reference).* Microsoft Press. (Or equivalent for PostgreSQL, e.g., "Mastering PostgreSQL 15" by Hans-Jürgen Schönig).

Celko, J. (2018). *Joe Celko's SQL for smarties: Advanced SQL programming (5th ed.).* Morgan Kaufmann.

Faroult, S., & Robson, P. (2006). *The art of SQL.* O'Reilly Media.

Additional Reading

Dewson, R. (2012). *Beginning SQL Server for developers*. Apress.

Kline, K. (2021). *SQL in a nutshell* (4th ed.). O'Reilly Media.

PostgreSQL Global Development Group. (2024). PostgreSQL documentation. https://www.postgresql.org/docs/

Microsoft. (2024). SQL Server documentation. Microsoft Learn. https://learn.microsoft.com/sql/

Date: May 2026

Module Code and Title: **NET201 Networking Fundamentals**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Ngawang Tenzin

General Objective

The primary objective of this module is to develop students' understanding of computer networking principles and their practical application. This module will build the ability of the students to analyse and apply networking concepts in different computing environments. It will also prepare students to understand how network systems support communication and services in modern organisations and technologies.

Learning Outcomes

On completion of this module, students will be able to:

i. explain the fundamental concepts of computer networks
ii. classify different types of networks and network topologies
iii. compare the osi and tcp/ip models
iv. identify and configure common network devices
v. analyse the functions of the data link layer
vi. apply ipv4 addressing and perform subnetting calculations
vii. implement basic routing using static and dynamic routing protocols
viii. analyse network traffic using appropriate tools
ix. evaluate transport layer protocols in data communication
x. describe application layer protocols used in networking
xi. assess emerging networking technologies.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent Study	Assignment	2	60
	Self-study	2	
Total		**8**	**120**

Assessment Approach

A. Class tests (10%)

This is an in-class written test with a duration of 40 minutes. Two such tests will be conducted, the first before the midterm, covering topics from Units I-III, and the second after the midterm, covering topics from Units IV-VI. Each test is worth 20 marks and will be converted to 5%.

B. Assignment I – Foundations on Computer Networking (10%)

Students will work in groups of 4–5 members on this assignment, which must be submitted by the end of the second week. This assignment covers Unit I and Unit II and is designed to assess students' understanding on the fundamental computer networking concepts, including basic networking terminology, type of networks, network components and the functions of different network layers. The grading will be based on the following criteria:

4 Marks Concepts (Demonstrates clear grasp of network configuration, transport layer, and network services)
2 Marks Accuracy and completeness (Correctness of explanations and coverage of all required topics)
2 Marks Organisation (Logical structure, clarity, and readability of the assignment)
2 Marks Originality (Show critical thinking and original interpretation of concepts)

C. Assignment II - Core Networking Layers (10%)

Students will work in groups of 4–5 members on Assignment 2, which must be submitted before the final examination. This assignment focuses on Unit III, Unit IV and Unit V. The assignment is designed to assess students' understanding of both theoretical and

conceptual aspects of computer networking, including data link layer mechanisms, network addressing and routing, and transport layer services. Students are expected to demonstrate knowledge of framing, error control, channel allocation techniques, Ethernet and wireless LAN standards, as well as IP addressing, routing algorithms and protocols, network security concepts, and transport layer protocols and congestion control mechanisms. The grading will be based on the following criteria:

4 Marks	Concepts (Accurate explanation of data link layer, Network layer and Transport layer)
2 Marks	Accuracy and completeness (Correctness of explanations and coverage of all required topics)
2 Marks	Organisation (Logical structure, clarity, and readability of the assignment)
2 Marks	Originality (Shows critical thinking and original interpretation of concepts)

D. Practical Examination (15%)

A practical examination will be conducted after the completion of the module and before the semester-end examination to assess students' ability to configure and troubleshoot network devices. The exam will be marked out of 50 and converted to 15%, and the duration of exam is one and half hours. The grading will be based on the following criteria:

15 Marks	Network setup and configuration (Correct topology, device setup and proper connectivity)
10 Marks	Addressing and network communication (Accurate IP addressing, subnetting and configuration of communication)
15 Marks	Connectivity verification and troubleshooting (Identification and resolution of network issues)
10 Marks	Practical analysis and use of simulation tools (Correct use of simulation tools, packet flow analysis and evidence of testing)

E. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours.

F. Semester-End Examination (35%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours.

Overview of the assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Class Test	1	10
B. Assignment I	1	10
C. Assignment II	1	10
D. Practical Examination	2	15
E. Mid-Semester Examination	1	20
F. Semester-End Examination (SEE)		35
Total		**100**

Pre-requisites: None

Subject Matter

Unit I: Introduction to Computer Network

1.1. Definitions, Uses, Benefits
1.2. Overview of Network Topologies
1.3. Overview of Network Types
1.4. Networking Types (Client/Server, P2P)

1.5. Overview of Protocols and Standards
1.6. OSI Reference Model
1.7. TCP/IP Models and its comparison with OSI.
1.8. Connection and Connection-Oriented Network Services
1.9. Internet, ISPs, Backbone Network Overview

Unit II: Physical Layer and Network Media

2.1. Network Devices: Repeater, Hub, Switch, Bridge, Router
2.2. Different types of transmission medias (wired: twisted pair, coaxial, fiber optic, Wireless: Radio waves, micro waves, infrared)
2.3. Ethernet Cable Standards (UTP & Fiber cable standards)
2.4. Circuit, Message & Packet Switching
2.5. ISDN: Interface and Standards

Unit III: Data Link Layer

3.1. Function of Data Link Layer (DLL)
3.2. Overview of Logical Link Control (LLC) and Media Access Control (MAC)
3.3. Framing and Flow Control Mechanisms
3.4. Error Detection and Correction techniques
3.5. Channel Allocation Techniques (ALOHA, Slotted ALOHA)
3.6. Ethernet Standards (802.3 CSMA/CD, 802.4 Token Bus, 802.5 Token Ring)
3.7. Wireless LAN: Spread Spectrum, Bluetooth, Wi-Fi
3.8. Overview Virtual Circuit Switching, Frame Relay& ATM
3.9. DLL Protocol: HDLC, PPP

Unit IV: Network Layer

4.1. Introduction and Functions
4.2. IPv4 Addressing & Sub-netting
4.3. Class-full and Classless Addressing
4.4. IPv6 Addressing and its Features
4.5. IPv4 and IPv6 Datagram Formats
4.6. Comparison of IPv4 and IPv6 Addressing
4.7. Example Addresses: Unicast, Multicast and Broadcast
4.8. Routing
    4.8.1. Introduction and Definition
    4.8.2. Types of Routing (Static vs Dynamic, Unicast vs Multicast, Link State vs Distance Vector, Interior vs Exterior)
    4.8.3. Path Computation Algorithms: Bellman Ford, Dijkstra's
    4.8.4. Routing Protocols: RIP, OSPF & BGP
4.9. Overview of IPv4 to IPv6 Transition Mechanisms
4.10. Overview of ICMP/ICMPv6&NATing
4.11. Overview of Network Traffic Analysis
4.12. Security Concepts: Firewall & Router Access Control

Unit V: Transport Layer

5.1. Introduction, Functions and Services
5.2. Transport Protocols: TCP, UDP and Their Comparisons
5.3. Connection Oriented and Connectionless Services
5.4. Congestion Control: Open Loop & Closed Loop, TCP Congestion Control
5.5. Traffic Shaping Algorithms: Leaky Bucket & Token Bucket
5.6. Queuing Techniques for Scheduling
5.7. Introduction to Ports and Sockets, Socket Programming

Unit VI: Application Layer

6.1. Introduction and Functions
6.2. Web &HTTP
6.3. DNS and the Query Types
6.4. File Transfer and Email Protocols: FTP, SFTP, SMTP, IMAP, POP3
6.5. Overview of Application Server Concepts: Proxy, Web, Mail
6.6. Network Management: SNMP

Unit VII: Multimedia and Future Networking

7.1. Overview Multimedia Streaming Protocols: SCTP
7.2. Overview of SDN and its Features, Data and Control Plane
7.3. Overview of NFV
7.4. Overview of NGN

List of Practicals

i. Understanding of Network equipment, wiring in details
ii. OS (Ubuntu/CentOS/Windows) installation, practice on basic Networking commands (ifconfig/ipconfig, tcpdump, netstat, dnsip, hostname, route...)
iii. Overview of IP Addressing and sub-netting, static ip setting on Linux/windows machine, testing
iv. Introduction to Packet Tracer, creation of a LAN and connectivity test in the LAN, creation of VLAN and VLAN trunking.
v. Basic Router Configuration, Static Routing Implementation
vi. Implementation of Dynamic/interior/exterior routing (RIP, OSPF, BGP)
vii. Firewall Implementation, Router Access Control List (ACL)
viii. Packet capture and header analysis by wire-shark (TCP, UDP, IP)
ix. DNS, Web, FTP server configuration (shall use packet tracer, GNS3)

Reading List

Essential Reading

Forouzan, B. A. (2007). *Data communications and networking* (4th ed.). McGraw-Hill.

Kurose, J. F., & Ross, K. W. (2003). *Computer networking: A top-down approach featuring the Internet* (2nd ed.). Pearson Education Asia.

Tanenbaum, A. S., Feamster, N., & Wetherall, D. J. (2021). *Computer networks* (6th ed.). Pearson Education.

Additional Reading

Black, U. D. (2000). *TCP/IP and related protocols*. McGraw-Hill.

Comer, D. E. (2018). *Internetworking with TCP/IP: Principles, protocols, and architecture* (6th ed.). Pearson.

Peterson, L. L., & Davie, B. S. (2012). *Computer networks: A systems approach* (5th ed.). Morgan Kaufmann.

Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer networks* (5th ed.). Pearson.

Stallings, W. (2014). *Foundations of modern networking: SDN, NFV, QoE, IoT, and cloud*. Addison-Wesley Professional.

Date: May 2026

Module Code and Title: **PGL203 Data Structures and Algorithms**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Phuntsho Wangdi

General Objective

This module aims to develop students' core competence in designing and implementing efficient Programmes by selecting appropriate data structures and algorithms. The module is designed to help students think computationally and implement solutions to programming problems.

Learning Outcomes

On completion of the module, students will be able to:

i. explain the concepts of data types, data structures, and Abstract Data Types (ADTs)
ii. describe algorithm analysis using asymptotic notation for time and space complexity
iii. implement fundamental data structures in C++ such
iv. implement basic tree structures such as Binary Search Trees (BST) to perform core operations
v. represent graphs using basic data structures
vi. use recursion to solve computational problems with appropriate base and termination conditions
vii. apply linear search and binary search
viii. use sorting algorithms to organize data efficiently
ix. apply hashing techniques, including basic collision-handling strategies
x. analyse the efficiency of algorithms and select suitable data structures or algorithms for a given problem scenario.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent Study	Assignment and Research	3	60
	VLE Discussion	1	
Total		**8**	**120**

Assessment Approach

A. Class Test (15%)

This is an in-class written test with a duration of 40 minutes. Two such tests will be conducted, the first before the midterm, covering topics from the start of the course up to the first quarter, and the second after the midterm, covering topics from immediately after the midterm to the next quarter. Each test is worth of 30 marks and the total will be converted to 15%.

B. Practical Examination (20%)

The Practical Examination will evaluate students' practical competence under time constraints. Tasks will require implementing or completing data structure and algorithm solutions in C++. Students will also need to note their implementation decisions briefly or annotate complexity at a basic level. The practical examination will be conducted after the completion of the module and before the semester-end examination, and the practical exam is worth 50 marks and converted into 20% over a duration of two hours.

9 Marks	Implementation (Program complies and run without errors, correct output, passes test cases)
6 Marks	Use of data structures and Algorithms (Correct selection of data structures, appropriateness of algorithms applied, logically structured solution design)
5 Marks	Reasoning and debugging approach (clear step, identifies and fixes errors, explains behaviour)

C. Mid-Semester Examination (25%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours.

D. Semester-End Examination (40%)
The Semester-end examination will be conducted for 100 marks over a duration of three hours.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Class Test	2	15
B. Practical Examination	1	20
C. Mid-Semester Examination	1	25
D. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: PGL102 Object-Oriented Programming

Subject Matter

Unit I: Introduction to Data Structures and Algorithms

1.1. Data types, data structures, and Abstract Data Types (ADTs)
1.2. Algorithm fundamentals: inputs/outputs, correctness, and basic design approaches
1.3. Asymptotic notation: Big-O, Big-Omega, and Big-Theta (time and space), common growth rates
1.4. C++ implementation foundations for DSA (arrays/vectors, basic pointers/references as needed, std: vector usage)
1.5. Recursion
    1.5.1. Recursion principles, base cases, and termination
    1.5.2. Recursion vs iteration (trade-offs)
    1.5.3. Simple recursion problems and efficiency discussion

Unit II: Lists: Array and Linked Lists

2.1. Array-based lists vs linked lists
2.2. Singly linked lists (core operations: insert, delete, traverse)
2.3. Conceptual awareness of doubly/circular lists

Unit III: Queues

3.1. Queue as an ADT: operations and applications
3.2. Linear and circular queues
3.3. Priority queue concept

Unit IV: Searching and Sorting

4.1. Searching: linear search, binary search, correctness, and complexity
4.2. Sorting: bubble/selection/insertion, merge sort, quicksort, and heap sort
4.3. Efficiency and suitability trade-offs

Unit V: Hashing

5.1. Hash tables and hash functions
5.2. Collision handling: chaining and open addressing
5.3. Performance reasoning and limitations

Unit VI: Trees

6.1. Trees: basic concepts, traversals, Binary Search Tree (BST) insert/search/delete, complexity discussion
6.2. Types of Binary Trees: full, complete, perfect, balanced, degenerate
6.3. Applications of Binary Search Trees: efficient searching, sorting, priority queues, symbol tables
6.4. Advanced Search Trees: introduction to AVL trees, Red-Black trees, and their use case

List of Practicals

i. Complexity practice: annotate Big-O for small code snippets and DS operations
ii. Stack operations and one application
iii. Queue operations (linear/circular) and one application
iv. Linked list implementation (insert/delete/traverse)
v. Recursion tasks (at least two) with correct base cases and testing
vi. Sorting implementations (at least two: one simple, one divide-and-conquer) with basic comparison of behaviour
vii. Searching tasks (linear and binary) with correct preconditions and testing
viii. Hash table implementation (chaining or open addressing) with collision handling demonstration
ix. BST operations (insert/search/traversal) and basic deletion
x. Basic graph representation and traversal (BFS or DFS). Usimr morer t thtr

Reading List

Essential Reading

Kruse, R. L., Tondo, C. L., & Leung, B. P. (2007). *Data structures and programme design in C* (2nd ed.). Pearson Education India.

Langsam, Y., Augenstein, M. J., & Tenenbaum, A. M. (2015). *Data structures using C and C++* (2nd ed.). Prentice-Hall of India.

Additional Reading

Ammeraal, L. (1992). *Programmes and data structures in C: Based on ANSI C and C++* (2nd ed.). Wiley.

Rowe, G. (1997). *Introduction to data structures and algorithms with C++*. Prentice Hall.

Date: May 2026

Module Code and Title: **EAP102 Upper Intermediate English for Academic Purposes**
Programme: BA in English Studies
Credit Value: 12
Module Tutor: Mohan Rai, Ruma Tamang, Tenzin Wangmo
Module Coordinator: Mohan Rai

General Objective

EAP102 is the second part of a two-semester series that aims to develop abilities in reading, writing, and speaking in an academic context to support students' learning through their degree studies. The module builds from the basics established in EAP101 and immerses the students fully in academic materials and tasks.

Learning Outcomes

On completion of the module, students will be able to:

i. describe the purpose of different sections of academic articles.
ii. interpret a variety of academic texts.
iii. state opinions on academic materials.
iv. produce research essays that synthesise information from a variety of sources.
v. use refined grammatical structures in academic writing.
vi. use the current APA referencing style for all types of academic sources.
vii. express cogent ideas appropriately in group and individual settings.
viii. speak fluently and coherently in academic settings.

ix. use contextually appropriate vocabulary in written and spoken settings.

Learning and Teaching Approach

Type	Approach	Hours per Week	Total Credit Hours
Contact	Interactive Lectures	2	60
	Exercise, workshops, practice	2	
Independent Study	Writing assignments, VLE discussions	4	60
	Reading and review of class materials		
Total		8	120

Assessment Approach

A. Reading Test: 15%

In a single sitting, students will read three passages of increasing difficulty and complexity. Each passage will be followed by a set of objective-type, short answer questions, or long answer questions on the respective passages, which will assess students' comprehension, critical thinking skills, grammar, and paraphrasing/summarising skills.

F. Charts and figures Class Test: 15%

In a single sitting, students will be given 2 charts or figures of increasing difficulty and complexity. Students will summarize the information in the charts by selecting and reporting the main features and making comparisons where relevant. Students will be graded according to the following criteria:

30 Content
20 Organisation and structure
20 Lexical resource
30 Language and Grammar

G. Interview: 20%

Students will sit for an interview to assess their speaking and communication skills. Each student will be given a period of 5-6 minutes to answer a set of questions prepared by the tutor. Students will initially be asked general questions about themselves and their interests in the first part. This part will last approximately 3 minutes. In the second part, students will be given a prompt on which they will speak for 1-2 minutes. They will have 1 minute to prepare before they speak on the prompt. Students will be graded according to the following criteria:

30 Fluency and coherence
10 Relevance
30 Lexical resource and word choice
30 Use of grammar

H. Fishbowl discussion: 15%

Students will be divided into groups of 6-7. Each group will be assigned one topic before their discussion and given 5 minutes for preparation. Students within each group will then be expected to take an individual stand on the topic and hold an impromptu discussion amongst themselves for 20-30 minutes. The tutor will observe the discussion and assess the students individually. The discussion will be assessed on the following criteria:

10 Relevance
20 Arguments and analysis
35 Comportment within the group
35 Language and speaking skills

I. Research Essay: 35%

The student will write a 700-850-word research essay on topics assigned by the tutor. The essay will follow academic standards discussed in the module and use APA referencing. The assignment will be written in two drafts: the first draft will be worth 15; and the final draft will be worth 20, including 5 for improvement.

Both drafts will be evaluated using the following criteria:

15 Content
15 Critical thinking
40 Use of academic standards
30 Language and grammar

Improvement on feedback will be evaluated using the following rubric:

0 – 49 Marginal improvement
50 – 59 Satisfactory improvement
60 – 74 Significant and appropriate improvement
75 – 100 Significant improvement beyond feedback given

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Reading Test	1	15
B. Charts and figures Class Test	1	15
C. Interview	1	20
D. Fishbowl discussion	1	15
E. Research Essay	1	35
Total		100

Pre-requisites: EAP101 Intermediate English for Academic Purposes

Subject Matter

The Subject Matter will be taught within the context of standard usage scenarios and examples from relevant texts. The skills, although divided amongst the different units, will not be taught in isolation. The EAP teaching team will maintain a Course Pack which is an Essential Reading Compilation of readings and practice materials, which they will revise and update regularly as per need. The Course Pack used for the semester should be endorsed by the Programme Management Committee.

Unit I: Academic Reading

1.1. Text features and Organisational elements
1.2. Components of an academic article
1.3. Interpreting and critically analysing academic texts
1.4. Reading skills: predicting, summarising, making text-to-text, text-to-world, and text-to- self connections
1.5. Summarising data in charts and figures

Unit II: Academic Writing

1.1 Features of Academic Writing
1.2 Using reporting verbs
1.3 Using linking words and transitional phrases
1.4 Paraphrasing and quoting sources
1.5 Intermediate APA referencing
1.6 Referring to figures and tables
1.7 Presenting contrasting information

Unit III: Grammar for Academic Writing

3.1 Word families and collocations

3.2 Drafting and building arguments
3.3 Noun phrases, verb phrases and prepositional phrases
3.4 Comparing and contrasting
3.5 Hedging expressions
3.6 Building contextual vocabularies

Unit IV: Academic Speaking

4.1 Discussing ideas with colleagues
4.2 Agreeing and disagreeing
4.3 Reaching a consensus
4.4 Outlining ideas and expressing opinions
4.5 Planning for oral tasks
4.6 Intonation and enunciation
4.7 Answering questions

Reading List

Essential Reading

(An Essential Reading compilation of the shorter reading materials will be made available to students).

American Psychological Association. (2020). *Publication manual of the American Psychological Association 2020: The official guide to APA style* (7th ed.). American Psychological Association.

Thaine, C. & McCarthy, M. (2014*). Cambridge academic English – An integrated skills course for EAP: B1+ (Upper Intermediate) student's book*. Cambridge University Press.

Wallwork, A. (2013). *English for academic research: Grammar, usage and style.* Springer.

Wallwork, A. (2022). *Giving an academic presentation in English.* Springer.

Date: July, 2025

Year 2 Semester I

Module Code and Title:	**NET202 System Administration**
Programme:	BSc in Applied Computing and Information Technology
Credit:	12
Module Tutor:	Phuntsho Wangdi

General Objective

This module will provide students with the knowledge and practical skills required for effective system administration. It focuses on developing the ability to manage and maintain computing systems, ensuring system reliability, and support organizational IT operations.

Learning Outcomes

On completion of this module, students will be able to:

i. automate routine system tasks using appropriate scripting techniques
ii. apply access control to manage users and permissions
iii. monitor system processes effectively to ensure optimal system performance
iv. identify issues promptly to prevent potential system failures
v. maintain file systems to ensure data integrity and organized storage of information
vi. manage storage system to enable efficient access and reliable data management
vii. schedule recurring tasks using system automation tools

viii. configure essential network services for system administration
ix. troubleshoot system and network issues to ensure system security, stability, and performance.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent Study	Assignment	2	60
	Self-study	2	
Total		**8**	**120**

Assessment Approach

A. Class Test (10%)

This will be an in-class written test with a duration of 40 minutes. Two such tests will be conducted, the first before the midterm, covering Unit I to Unit III, and the second after the midterm, covering Unit IV to Unit VII. Each test is worth 20 marks and the total marks will be converted to 10%.

B. Assignment: Computational Problem solving (15%)

Students will work in groups of 4–5 members and complete both an assignment and a presentation. Each group will be assigned a set of problem-solving questions covering topics from Unit I to Unit IV. Group members are expected to collaborate effectively and ensure equal contribution from all members. Depending on the available time and class size, each group will be required to deliver a 10–12-minute presentation focusing on the underlying and relevant theoretical concepts related to the problems solved in the assignment. Each assignment will be assessed per the criteria set.

Task 1: Assignment (10%)

3 Marks	Accuracy of Methods and Answers (Correct methods used and accurate final answers)
2 Marks	Logical Explanation (Clarity and coherence in presenting steps and reasoning)
2 Marks	Originality (Evidence of independent thinking and avoidance of duplication)
3 Marks	Individual Contribution and Teamwork (Equal participation, collaboration, and documented contribution of each member; may include peer or tutor evaluation

Task 2: Presentation (5%)

Each presentation will be assessed as per the following criteria.

2 Marks	Content and Completeness (Coverage of key concepts and relevance to the assigned problems)
3 Marks	Presentation skills (confidence, clarity of speech, structure and respond to questions)

C. Practical Examinations (15%)

The Practical Examination will be conducted for 1 hour towards the end of the semester, before the Semester-End examination. This assessment will test the ability of students to analyse, design, and implement modular programs based on all 12 practicals included in the syllabus. The tutor will select a subset (5–6 practicals) for the examination, so students must be prepared to apply concepts from all listed practicals. The practical examination grading will focus on the following criteria:

Task 1: Practical Programming (10%)

- 3 Marks Program Analysis (Clear understanding of the program and requirements)
- 3 Marks Program Design (Proper logic, algorithm and structure)
- 4 Marks Coding (Correct syntax, efficiency code and successful execution)

Task 2: VIVA (5%). This will be out of 10 marks and converted to 5%.

- 5 Marks Explanation of program (Ability to clearly explain the logic, flow, and structure of the written code)
- 3 Marks Conceptual understanding (Knowledge of programming concepts, modularity, and problem-solving approach)
- 2 Marks Response (Accuracy and clarity in answering follow-up question)

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and a half hours.

E. Semester-end Examination (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours, and the marks obtained will be converted to 40%.

Overview of assessment approaches and weighting

Areas of Assessment	Quantity	Weighting (%)
A. Class Test	2	10
B. Assignment	1	15
C. Practical Examination	1	15
D. Mid-Semester Examination	1	20
E. Semester-end Examination (SEE)	1	40
Total		**100**

Pre-requisites: CCS202 Operating System, NET201 Network Fundamentals and Service

Subject Matter

Unit I: Introduction to System Administration

1.1. Roles and responsibilities
1.2. System administrator
1.3. Linux distributions and administration features
1.4. System-specific administration tools and utilities
1.5. Using *man* command for documentation

Unit II: Scripting and the Shell

2.1. Shell basics and command-line usage
2.2. Bash scripting fundamentals
2.3. Regular expressions for system tasks
2.4. Python scripting for automation
2.5. Difference between Bash and python Scripting

Unit III: Access Control and Privileges

3.1. Traditional UNIX file permissions
3.2. Modern access control (ACLs)

3.3. Managing users, groups, and root privileges
3.4. Using sudo and role-based access

Unit IV: Controlling Processes

4.1. Components and life cycle of processes
4.2. Process states and scheduling
4.3. Signals and process control (*kill, ps, top*)

Unit V: File System Administration

5.1. File hierarchy, pathnames, and organization
5.2. File types, attributes, and permissions
5.3. Mounting and unmounting file systems
5.4. Access Control Lists (ACLs)

Unit VI: Task Scheduling and Automation

6.1. Automating tasks with *cron* command
6.2. Crontab file format and management
6.3. Scheduling one-time tasks using *at* Command

Unit VII: System-Level Networking for Administrators

7.1. DHCP configuration and working
7.2. Point-to-Point Protocol (PPP) setup and troubleshooting
7.3. Network configuration tasks for Linux systems
7.4. Monitoring system-level network services

Unit VIII: Domain Name System (DNS) Administration

8.1. Managing name servers and DNS namespaces
8.2. Delegation, caching, and extended DNS protocol
8.3. Creating subdomains and managing domains
8.4. Key DNS resource records: SOA, NS, A, PTR

Unit IX: Network File System (NFS) Administration

9.1. Introduction to network file services
9.2. Server-side NFS configuration (export file, nfsd)
9.3. Client-side NFS setup (mounting, restricting exports, map files)
9.4. Monitoring NFS (nfsstat) and automatic mounting

Unit X: System and Network Troubleshooting

10.1. Monitoring system resources (CPU, memory, disk)
10.2. Log management and basic security monitoring
10.3. Tools for troubleshooting: top, htop, journalctl
10.4. Network troubleshooting tools from admin perspective
10.4.1. SmokePing, SNMP, NetFlow and packet sniffers

List of Practicals

i. Installation of a Linux operating system using a virtual machine
ii. Configure a web server and virtual hosts
iii. Creating, modifying, and deleting users
iv. Creating and managing user groups
v. Changing file and directory ownership
vi. Creating and managing scheduled tasks using crontab
vii. Explore various file/directory handling commands: pwd, cd, ls, rm, mv, cat, cp, echo, mkdir, rmdir
viii. Shell script to check whether the given number is even or odd

ix. Shell script to find GCD and LCM of two numbers
x. Shell script to assign file permissions using (a) symbolic mode and (b) absolute mode
xi. Shell script to display a greeting according to the time of day
xii. Shell program to compress and decompress files using gzip and gunzip

Reading List

Essential Reading

Nemeth, E., Hein, T. R., Whaley, B., & Snyder, G. (2010). *Unix And Linux System Administration Handbook*. Upper Saddle River, N.J: Prentice Hall.

Negus, C. (2015). *Linux Bible*. Somerset: Wiley.

Carling, M., Degler, S., & Dennis, J. (2000). *Linux System Administration*. Indianapolis, IN: New Riders.

Additional Reading

Frisch, A. (2009). *Essential System Administration*.

Limoncelli, T. A., Hogan, C. J., & Chalup, S. R. (2007). *The Practice of System and Network Administration*. Addison-Wesley

Date: May 2026

Module Code and Title: **SDP201 Systems Analysis and Design**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Sonam

General Objective

This module intends to equip students with the knowledge and skills to analyse, design, and document information systems that meet organisational requirements. Students will learn to apply methodologies and tools to produce professional artefacts that support effective system development and integration.

Learning Outcomes

On completion of the module, students will be able to:

i. conduct system analysis and capture requirements.
ii. explain the role of system analysis and design in the software development lifecycle
iii. apply structured and object-oriented techniques to model system requirements
iv. use various tools to represent processes and data
v. apply modelling techniques to document and integrate system processes.
vi. evaluate alternative system solutions against business and technical requirements
vii. design logical and physical system models
viii. present system analysis and design artefacts clearly and professionally
ix. develop professional specifications of system design principles.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	3	60
	Computer Laboratory Session	1	
Independent study	Assignments, Project work, Reading and review of class materials	4	60

Total	**8**	**120**

Assessment Approach

A. VLE Quiz (10%)

A quiz will be conducted via the Virtual Learning Environment (VLE) during class hours for 35 minutes of maximum 25 marks and the marks obtained will be converted into 10%. It will include multiple-choice, true/false, and short-answer questions covering Units I and II. The quiz is designed to test recall, comprehension, and application of foundational concepts in system analysis and design

B. Case Study Analysis (15%)

Students in groups of 4–5 students will complete a written case study analysis within 1200–1500 words focusing on evaluating system requirements for a given organisational scenario and proposing design solutions. The assessment includes process/data modelling and structured recommendations. The assessment will be graded out of 20 marks and will be converted to 15% based on the following marking criteria:

4 Marks	Understanding (Clear grasp of system needs and solution options)
5 Marks	Modelling and analysis (Effective use of UML/DFD to identify improvements)
3 Marks	Risk and compliance (Addresses risks, ethics and governance issues)
2 Marks	Clarity of recommendation (Provide actionable, well-structured conclusions)
2 Marks	Individual Contribution (Evidence of meaningful participation and input from each group member)
2 Marks	Language and Presentation (Uses clear, professional, and coherent academic writing style with proper grammar and structure)
2 Marks	References and Citation (Appropriate use of credible sources, correctly cited and referenced)

C. System Design Specification Project: 20%

Students in groups of 4–5 members will submit a written report (1600-1800 words) on developing a System Design Specification. Also, students will prepare a 15 minutes group presentation consisting of 10 to 12 slides. The project includes UML diagrams, ER models, and interface mock-ups, demonstrating integration of methodologies and tools. Both the assessment will be graded out of 25 marks and will be converted to 10% each based on the following marking criteria:

Report (10%)

5 Marks	Requirements alignment (Specification aligns with organisational needs)
4 Marks	Architecture and modelling (Appropriate logical/physical models)
3 Marks	Benefits and risk management (Identifies benefits, risks, and mitigation)
4 Marks	Implementation plan (Outlines realistic steps and phases)
3 Marks	Structure and quality (Well organised, clear, professional formatting)
2 Marks	Individual Contribution (Evidence of meaningful participation and input from each group member)
2 Marks	Language and Presentation (Clear, professional academic writing style with proper grammar and flow)

2 Marks References and Citation (Use of credible sources, correctly cited and referenced in line with academic standards)

Presentation (10%)

5 Marks Clarity of content (Key ideas communicated logically)
4 Marks Engagement and delivery (Confident, engaging presentation)
3 Marks Visual support (Clear, professional slides, number of slides)
3 Marks Team collaboration (Effective teamwork and coordination)
3 Marks Communication (Persuasive and audience-aware)
3 Marks Individual Contribution (Each member demonstrates active participation in the presentation)
2 Marks Language and Expression(Fluent, professional language with clear articulation and grammar)
2 Marks References (Proper acknowledgement of sources used in slides and presentation)

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours.

E. Semester-End Examination (35%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours.

Overview of assessment approaches and weighting

Areas of Assessments	**Quantity**	**Weighting**
A. VLE Quiz	1	10
B. Case Study Analysis	1	15
C. System Design Specification Project	1	20
D. Mid-Semester Examination	1	20
E. Semester-End Examination (SEE)	1	35
Total		**100**

Pre-requisites: None

Subject Matter

Unit I: Foundations of System Analysis and Design

1.1. Introduction to information systems in organisations
1.2. Systems Development Life Cycle (SDLC) phases:
  1.2.1. Planning
  1.2.2. Analysis
  1.2.3. Design
  1.2.4. Implementation
  1.2.5. Maintenance
1.3. Feasibility Analysis in System Development
  1.3.1. Technical feasibility
  1.3.2. Economic feasibility (cost–benefit analysis)
  1.3.3. Operational feasibility
  1.3.4. Schedule feasibility
  1.3.5. Legal and regulatory feasibility
1.4. Importance of system analysis and design in project success
1.5. Relationship between business needs and system solutions

1.6. Historical evolution of system development approaches
1.7. Key stakeholders in system projects

Unit II: Role of the System Analyst

2.1. Responsibilities of a system analyst in organisations
2.2. Required skills
    2.2.1. Technical
    2.2.2. Analytical
    2.2.3. Communication
    2.2.4. Interpersonal
    2.2.5. Leadership
2.3. Ethical responsibilities and professional conduct
2.4. Career paths and certifications for system analysts

Unit III: Requirements Gathering and Analysis

3.1. Techniques:
    3.1.1. Interviews
    3.1.2. Questionnaires
    3.1.3. Observation
    3.1.4. Document review
3.2. Functional vs. non-functional requirements
3.3. Requirements validation and prioritisation
3.4. Common challenges in requirements elicitation
3.5. Tools for capturing requirements
    3.5.1. User stories
    3.5.2. Use cases
    3.5.3. Requirement matrices
    3.5.4. Prototyping for requirement validation
3.6. Requirements traceability and change management
3.7. Stakeholder analysis and communication strategies

Unit IV: Modelling Techniques

4.1. Data Flow Diagrams (DFD)
    4.1.1. Context
    4.1.2. Level-0
    4.1.3. Level-1 diagrams
4.2. Use Case diagrams and scenarios
4.3. UML diagrams
    4.3.1. Class
    4.3.2. Sequence
    4.3.3. Activity
    4.3.4. State diagrams
4.4. Process modelling basics
    4.4.1. Workflow charts
    4.4.2. Swimlanes
    4.4.3. SIPOC diagrams
4.5. Business rules modelling
4.6. Integration of models into system documentation

Unit V: System Design Principles

5.1. Logical vs. physical design
5.2. Interface design and usability principles (UI/UX basics)

5.3. Data design and database schema development
5.4. Security and compliance considerations in design
5.5. Documentation standards for design artefacts
5.6. Designing for scalability and performance
5.7. Prototyping and iterative design approaches

Unit VI: Introduction to Project Management in System Development

5.1. Work breakdown structures and Gantt charts
5.2. Resource allocation and scheduling
5.3. Risk management in system projects
5.4. Quality assurance and testing strategies
5.5. Change management and stakeholder communication
5.6. Cost estimation models
5.7. Project estimation techniques
    5.7.1. Function points
    5.7.2. Story points
5.8. Monitoring and controlling project progress

Reading List
Essential Reading

Dennis, A., Wixom, B. H., & Roth, R. M. (2021). *Systems analysis and design* (8th ed.). Wiley.
Kendall, K. E., & Kendall, J. E. (2019). *Systems analysis and design* (10th ed.). Pearson.
Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2020). *Systems analysis and design in a changing world* (7th ed.). Cengage Learning.

Additional Reading

Nguyen, T. M., & Bui, T. X. (2021). *Recent trends in systems-of-systems design, modeling, simulation and analysis*. IntechOpen. https://doi.org/10.5772/intechopen.100000
Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
Turban, E., Pollard, C., & Wood, G. (2018). *Information technology for management: On-demand strategies for performance, growth and sustainability* (11th ed.). Wiley.
Valacich, J. S., George, J. F., & Hoffer, J. A. (2021). *Essentials of systems analysis and design* (6th ed.). Pearson.

Date: May 2026

Module Code and Title: **WEB202 Dynamic Front-end Development**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Arun Kumar Singh

General Objective
The aim of this module is to allow the students to advance their knowledge in front-end web development by using modern front-end frameworks. The module aims to strengthen students' understanding of designing, developing and managing interactive, responsive and user-friendly web applications while applying industry-standard tools, best practices and development methodologies.

Learning Outcomes
On completion of the module, students will be able to:

i. apply modern JavaScript to create maintainable client-side applications
ii. develop interactive interfaces using component-based front-end design
iii. manage application state and data flow to build reliable client-side applications
iv. implement client-side navigation and routing in single-page applications
v. integrate external APIs to enhance application functionality
vi. design accessible and user-friendly web interfaces
vii. optimise front-end application performance
viii. apply version control practices to manage collaborative development.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent Study	Assignment	2	60
	Self-study	2	
Total		**8**	**120**

Assessment Approach

A. VLE Quiz (10%)

A quiz will be conducted individually to assess students' understanding of key front-end development concepts. The quiz will cover Unit I and Unit II and will include multiple-choice questions. It will be conducted in class over a duration of 40 minutes and a total of 30 marks, contributing 10% to the final course mark.

B. Assignment- Building an Interactive Web Application (10%)

Each student will implement an interactive feature using ES6+ concepts, including modules, classes and promises/async–await, while maintaining a clear separation of concerns. The assignment will be assessed for a total of 50 marks and the marks obtained will be converted into 10%. The following criteria will be followed for marking:

20 Marks	Code Functionality (Correct implementation of the feature and accurate use of ES6+ concepts)
15 Marks	Code structure and organisation (use of modules, classes and organised code structure)
10 Marks	Code Quality and Readability (consistent naming, formatting and readable code)
5 Marks	Documentation (Well-commented code and clear explanation of the logics)

C. Project (20%)

Students in groups of 3–4 members will develop a small single-page application (SPA) that integrates an external API, implements key accessibility practices following WCAG/ARIA guidelines, and localises core UI strings. The project will be divided into two parts: Prototype and Functionality (10%) and Final Application and Documentation (10%).

i. **Prototype and Functionality (10%)**

Each group will submit a working prototype demonstrating basic SPA structure, API integration and initial accessibility/localisation features. It will be assessed based on the following criteria:

4 Marks	Core Functionality (Basic SPA structure and API integration works correctly)
3 Marks	Accessibility and Localisation (Initial implementation of WCAG/ARIA practices and localisation)

3 Marks Team Collaboration (Contribution of all group members and coordination in developing prototype)

ii. **Final Application and Documentation (10%)**

Each group will submit the complete SPA with full API integration, enhanced accessibility/localisation, and clear documentation explaining design decisions, implementation, and testing. It will be assessed based on the following criteria:

4 Marks Functionality and Features (Full implementation of SPA and API integration, dynamic content and accessibility/localisation)
3 Marks Code Quality and Organisation (Clean, maintainable and modular code)
3 Marks Documentation and Presentation (Clear explanation of implementation, design decisions and demonstration of SPA)

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 100 marks over a duration of one and half hours.

E. Semester-End Examination (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours.

Overview of assessment approaches and weighting

Areas of Assignments		**Quantity**	**Weighting**
A. VLE Quiz		1	10%
B. Assignment		1	10%
C. Project	Prototype & Functionality	1	10%
	Final Application and Documentation	1	10%
D. Mid-Semester Examination		1	20%
E. Semester-End Examination (SEE)		1	40%
Total			**100%**

Pre-requisites: WEB101 Foundations of Web Development

Subject Matter

Unit I: Modern JavaScript and Tooling

1.1. ES6+ essentials: modules, classes, destructuring, template literals
1.2. Asynchronicity: promises, async/await, error handling patterns
1.3. Build/tooling basics: npm scripts, linting, formatting, bundling at a conceptual level

Unit II: Advanced Browser APIs and Forms

2.1. Fetch API and JSON parsing; basic REST patterns.
2.2. Storage and caching concepts (local/session storage, cache hints).
2.3. Forms: validation patterns, accessible error messages, progressive enhancement

Unit III: React Fundamentals

3.1. JSX and components: props, children, composition
3.2. Hooks (useState, useEffect); basic side-effects & cleanup
3.3. Component lifecycles (functional hook patterns), separation of concerns

Unit IV: State Management and Data Flow

4.1. Lifting state and data down; Context API basics
4.2. Reducer patterns (useReducer) and derived state; immutable updates
4.3. Handling async states: loading, error, retry, empty states

Unit V: Client-Side Routing

5.1. SPA concepts; routes, links, params, nested routes
5.2. Route guards and redirection; accessibility considerations (focus on route change)
5.3. URL design and deep-linking

Unit VI: Performance and Quality

6.1 Memoization (memo/useMemo/useCallback); avoiding unnecessary renders
6.2 Code splitting and lazy loading; image/asset optimisation
6.3 Profiling and measuring performance; basic bundle insights

Unit VII: Testing

7.1 Jest and React Testing Library setup; testing philosophy
7.2 Unit tests for components; integration tests for user flows
7.3 Mocking fetch/async; testing accessibility (roles, labels, keyboard)

List of laboratory works

i. ES6+ clinic: Refactor legacy JS to modules/classes; add async/await.
ii. Fetch and forms: Build an accessible search form; render API results.
iii. React basics: Create a component tree; props/state; simple effects.
iv. State and context: Implement shared state via Context; add derived state.
v. Routing: Add multiple routes/views; ensure focus and keyboard paths on route changes.
vi. Accessibility sprint: ARIA roles, focus traps, live regions; audit and fix issues.
vii. Performance sprint: Memoize heavy components; code split a route; profile improvements.
viii. Testing sprint: Write unit/integration tests using Jest/RTL
ix. Localisation: Introduce a Dzongkha translation file; switch UI strings based on locale.

Reading List

Essential Reading

Banks, A., & Porcello, E. (2020). *Learning React: Modern patterns for developing React apps* (2nd ed.). O'Reilly Media.

Caldwell, B., Cooper, M., Reid, L. G., & Vanderheiden, G. (2008). *Web content accessibility guidelines (WCAG) 2.0* (W3C Working Draft). World Wide Web Consortium.

Flanagan, D. (2020). *JavaScript: The definitive guide* (7th ed.). O'Reilly Media.

Grigorik, I. (2013). *High performance browser networking: What every web developer should know*. O'Reilly Media.

Additional Reading

Abou-Zahra, S., & W3C. (2021). *Web accessibility curriculum guide*. World Wide Web Consortium.

Clark, J., & Cooper, M. (2012). *Accessible America: A history of disability and design*. University of Massachusetts Press.

Crockford, D. (2008). *JavaScript: The good parts*. O'Reilly Media.

Esselink, B. (2000). *A practical guide to localisation* (2nd ed.). John Benjamins.

Meszaros, G. (2007). *xUnit test patterns: Refactoring test code*. Addison-Wesley Professional.

Wright, A., & Suderman, D. (2023). *React testing in action*. Manning Publications.

Date: May 2026

Module Code and Title: **MTH203 Statistics and Probability**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutors: Hari Kumar Tiwari and Tandin Wangchuk
Module Coordinator: Hari Kumar Tiwari

General Objective
This module aims to develop problem-solving skills through probabilistic reasoning and statistical methods, enabling students to apply these concepts in computing, data science, and real-world IT applications.

Learning Outcomes
On completion of the module, students will be able to:

i. classify the types of data to make better analysis
ii. construct a frequency distribution table from the raw data
iii. represent the data in graphs from the data tables
iv. interpret the data based on calculated values
v. use different methods to solve problems on correlation
vi. determine lines of regression from the given data
vii. solve the various types of problems from probability
viii. differentiate the types of probability distribution.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Class Activities	2	
Independent Study	Self-study	2	60
	Written Assignment	1	
	VLE Discussion	1	
Total		**8**	**120**

Assessment Approach
A. Data representation and visualisation (15%)

Students will form groups of four or five to research on the topics *Visualising Data* and *Graphical Representation of Data*. Each group will be responsible for collecting relevant data and applying different statistical tools to analyse it. Based on their analysis, they must prepare a clear and well-organised report, with a length of 450 – 500 words, summarising their findings. In addition to submitting the written report, each member from the group will present their work for a duration of 2 to 3 minutes in class through a viva or presentation, which will be used to assess the originality and depth of their research. From this assignment, students should be able to present the raw data in the frequency distribution table, draw different types of charts and interpret the data.

Task 1: Report Writing (10%)

3 Marks	Content Quality (accuracy, relevance, balance of fact and opinion and evidence-based)
2 Marks	Language (formal tone, clarity, conciseness, audience awareness)
2 Marks	Formatting and Presentation (consistency, visual, numbering and heading and proofreading)
3 Marks	Structure (title page, table of contents, introduction, body, conclusion, recommendation and reference)

Task 2: Viva or Presentation

The Viva or presentation will be marked out of 10 marks and converted to 5%.

For viva-voce:

5 Marks	Subject Knowledge (Precise, comprehensive, well-supported answers)
3 Marks	Communication (Clear, structured, fluent responses)
2 Marks	Confidence (Calm, assertive, effective handling of questions)

For Presentation:

5 Marks	Visual Design (consistency, clarity, visual aids and cognitive ease)
3 Marks	Delivery and style (Confident and body language, voice modulation, timing)
2 Marks	Audience engagement (know your audience, interactive elements and relatable language)

B. Problem Solving (10%)

Students will individually use the Python programming language to solve problems from Unit I and Unit II before the mid-semester examination, and from Unit IV after the mid-semester. The specific dates for these problem-solving sessions will not be announced in advance. Each student is required to complete two problem-solving exercises, each worth 15 marks. The marks obtained will be converted into 5% of the overall grade. Each exercise will have a duration of 45 minutes.

C. Class Tests (15%)

Each student will individually complete two class tests, consisting of application-based problems drawn from the theoretical concepts taught in class from Unit III and Unit V respectively. Each test will carry 20 marks and the marks obtained will be converted into 15%. The written test will have a duration of 45 minutes.

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over the duration of one and half hours converted to 20%.

E. Semester-End Examination (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours converted to 40%.

Overview of the assessment approaches and weighting:

Areas of Assessments	Quantity	Weighting (%)
A. Data representation and visualisation	1	15
B. Problem Solving	2	10
C. Class tests	2	15
D. Mid-Semester Examination	1	20
E. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites:

Subject Matter

Unit I: Visualising and Presentation Data Tables

1.1 Meaning of statistics
1.2 The different types of data variables
1.3 Types of sampling

1.4 Method of data collection
1.5 Creating a frequency distribution table
    1.5.1 inclusive method
    1.5.2 exclusive method
1.6 Cumulative frequency and Relative frequency

Unit II: Graphical Representation of Data

2.1. Line graph
2.2. Pie chart
2.3. Bar graph
    2.3.1. Simple bar graph
    2.3.2. Multiple bar graph
    2.3.3. Compound bar graph
2.4. Histogram
    1.4.1. Histogram with unequal class intervals
    1.4.2. Frequency polygon and frequency curve

Unit III: Data Descriptors

3.1 Measures of Central tendency
    3.1.1. Mean, median, and mode
    3.1.2. Percentiles and quartiles
    3.1.3. Averages from frequency distributions
    3.1.4. Weighted Averages
    3.1.5. Functions used in MS Excel:
        3.1.5.1 AVERAGE, MEDIAN, MODE, and PERCENTILE,
        3.1.5.2 QUARTILE, SUM, and SUMPRODUCT
3.2 Measures of dispersion
    3.2.1. Range
    3.2.2. Inter-quartile range and semi- inter-quartile range (SIQR)
    3.2.3. Standard deviation and Variance
    3.2.4. Coefficient of variation
    3.2.5. Measurement of skewness, kurtosis and moments.
    3.2.6. Functions used in MS Excel:
        3.2.6.1 MAX, MIN, VAR, VARP, STDEV, SQRT, COUNT, SKEW, KURT
        3.2.6.2 Box plots and Descriptive statistics from Excel toolbar add-in.

Unit IV: Correlation and Regression Analysis

4.1 Linear correlation analysis
    4.1.1 Scatter plots
    4.1.2 Covariance
    4.1.3 Pearson's coefficient of correlation
    4.1.4 Significance of linear correlation
    4.1.5 Spearman's rank correlation coefficient
    4.1.6 Kendall's rank correlation coefficient
4.2 Linear regression analysis
    4.2.2 Construct scatter plot
    4.2.2 Fit line to sample data
    4.2.3 Test model reliability
        4.2.3.1 Standard Error of Estimate
        4.2.3.2 Coefficient of Determination
    4.2.4 Excel data analysis regression analysis
    4.2.5 Functions used in MS Excel:
        4.2.5.1 INTERCEPT, SLOPE, STEYX,

4.2.5.2 Regression from Excel toolbar add-in

Unit V: Probability

5.1 Introduction to Probability
- 5.1.1 Basic terms and types of events
- 5.1.2 Approaches of probability
  - 5.1.2.1 Classical
  - 5.1.2.2 empirical and axiomatic
- 5.1.3 Addition and Multiplication theorems on probability (without proof)
- 5.1.4 Conditional probability and Bayes' formula

5.2 Probability distributions
- 5.2.1 Binomial distributions
- 5.2.2 Poisson distributions
- 5.2.3 Normal distributions

Reading List
Essential Reading

Aczel, A. D. (2012). *Complete business statistics* (8th ed.). Wohl Publishing.
Beri, G. C. (2011). *Business statistics* (3rd ed.). McGraw-Hill Education.
Sharma, J. K. (2014). *Business statistics* (4th ed.). Vikas Publishing House Pvt. Ltd

Additional Reading

Gupta, S. P., & Gupta, M. P. (2017). *Business statistics* (19th rev. ed.). Sultan Chand & Sons.
Levine, D. M., Szabat, K. A., & Stephan, D. F. (2013). *Statistics for managers using Microsoft Excel* (7th ed.). Pearson.
Thukral, J. K. (2011). *Business statistics* (3rd ed.). Taxmann Publication Pvt. Limited

Date: May 2026

Module Code and Title:	**DZG101 རྩོད་གནད་དོན་སྤྱིད་ལེན།**
སྤྱོང་ཚན་གྱི་མིང་དང་སྤྱོང་ཚན་ཨང་	རྩོད་གནད་དོན་སྤྱིད་ལེན།
ལས་རིམ་	གཅུག་ལག་སློབ་སྟེ་ཡོངས་ཁྱབ།
སྤྱོང་འཐུས་	༣
སྤྱོང་ཚན་སློབ་སྟོན་པ།	རྩོད་གཞི་ལེགས་བཤད་པ།
སྤྱོང་ཚན་འགོ་འཛིན་པ།	བསོད་ནམས་སྟོབས་རྒྱས།

སྤྱོང་ཚན་འདི་གི་དམིགས་ཡུལ་གཙོ་བོ་ར་ སློབ་སྤྱོང་པ་ཚུ་གིས་ རང་གི་མི་ཚེ་ནང་ ལྟ་གཡོག་དང་འབྲེལ་བའི་གནད་དོན་ གཅིའི་ཐད་ལས་འབད་རྒྱུང་ རྩོད་གཞི་ཐོག་ལུ་ བརྟོན་སྤྱིད་ལེན་ ཚུལ་དང་མཐུན་ཏོག་ཏོ་སྦེ་ འབད་ཚུགས་ནི་འི་དོན་ལུ་ཨིན། དེ་མ་ཚད་ སྤྱོང་ཚན་འདི་གིས་ རྩོད་གཞི་ཐོག་ལུ་ ཤེས་ཡོན་འཛི་རྒྱལ་གྱི་ལས་ལུགས་དང་འཁྲིལ་ཏེ་ བྱེ་ཚུགས་ནི་ལུ་ཡང་དམིགས་གཏད་བསྐྱེད་ཨིན།

སློབ་སྤྱོང་གྲུབ་འཐས།
སྤྱོང་ཚན་འདི་མཇུག་བསྡུ་ད་ སློབ་ཐུག་ཚུ་གིས་

i. རྩོང་ཁའི་སྐད་ཡིག་གི་ངོས་བླངས་དང་ རྩོང་ཁ་ལྟབ་དགོ་པའི་ཁྱངས་དང་དགོས་པ་ཚུ་ སྤྲལ་ཚུགས།

ii. སློབ་རིག་ནང་ལུ་ རྩོང་ཁའི་ཡིག་གཟུགས་བཙུགས་ཏེ་ གཡོག་བཀོལ་ཚུགས།

iii. ཡི་གུའི་སྒྱོར་བ་ འཛེལ་སྒྲ་དང་བྱེད་སྒྲ་ ལྟག་བཅས་ ཚིག་གྲོགས་ འཛེལ་ཚིག་ཚུ་ དབྱེ་དཔྱད་འབད་དེ་ མ་འཛིལ་བར་ལག་ལེན་འཐབ་ཚུགས།

iv. འབྲི་རྩལ་ ཚག་ཤད་དང་ རྩོད་ཚིག་དང་ དོན་མཚམས་ཚུ་ ཚུལ་མཐུན་སྒྲིག་ཏི་ཚུགས།

v. ལུང་འདེན་དང་རྒྱབ་རྟེན་ འོས་འབབ་ལྟན་ཏོག་ཏོ་སྒྲི་ གཡོག་བཀོལ་ཚུགས།

vi. ཡུལ་ དུས་ གནས་སྟངས་དང་བསྟུན་ཏེ་ རྩོང་ཁའི་ཐོག་ལུ་ ཉན་སྒྲབ་འབད་ཚུགས།

vii. རྩོང་ཁའི་ཐོག་ལུ་བྱིས་ཏེ་ཡོད་མི་ཚུ་ ངག་གཤིས་དང་མཐུན་སྒྲི་ལྟག་ཚུགས།

viii. གནས་སྟངས་དང་བསྟུན་པའི་དཔེ་གཏམ་ གཡོག་བཀོལ་ཚུགས།

ix. གཞུང་སྐེར་གྱི་ཡིག་རིགས་ ལུགས་མཐུན་དང་ལུགས་ཡངས་གཉིས་ཆར་འི་ཐོག་ལུ་ བྱེ་ཚུགས།

x. རྩོང་ཁ་ནང་ཡོད་པའི་ འབྲི་ཤོག་གི་རིགས་ག་ཅི་ར་ཡིན་རྟོང་ བཀང་ཚུགས།

སློབ་སྟོན་ཐབས་ལམ།

དཔྱ་བ།	ཐབས་ལམ།	བདུན་ཕྲག་གི་ཚེག་ནང་ཆུ་ཚོད།	སྐྱོང་འཛུས་ཆུ་ཚོད།
དངོས་འཛིལ།	གསལ་བཤད།	༢	༤༠
	སྐྱོང་ལྷ།	༡	
	སྤྱན་ཀྱ།	༡	
རང་སྐྱོད།	ལས་འགུལ་བྱེ་ནི།	༢	༤༠
	དཔེ་མཛོད། ལྟག་དེབ་ལྟག་ནི།	༢	
སྐྱོང་ཚན་འདི་གི་དོན་ལུ་ ཡོངས་བསྡོམས་ ཆུ་ཚོད་			༡༢༠

དཔྱ་ཞིབ་ཐབས་ལམ།

སྐྱོང་ཚན་འདི་གི་དོན་ལུ་ དུས་རྒྱུན་དཔྱ་ཞིབ་དང་སྐྱངས་རྒྱུགས་དཔྱ་ཞིབ་ གཉིས་ཆར་འི་ཐོག་ལས་ དཔྱ་ཞིབ་འབད་དགོས་ཨིན།

༡ ལས་འགུལ་དང་པ། ཉན་ཞིའི་རིག་རྩལ་དཔྱ་ཞིབ། ： ༢༠%

སློབ་སྟོན་པ་གིས་འོས་འབབ་ལྟན་པའི་དོན་ཚན་ གང་རུང་ཅིག་གདམ་ཁ་རྒྱབ་སྟེ་ (སྒྲ་བཟུང་ཡོད་མི་ ཡང་ཅིན་ སློབ་དཔོན་གྱིས་ ལྟག་སྟེ་) སློབ་ཁྲག་ངང་རྒྱང་གི་ཐོག་ལས་ ཁུ་སིམ་སིམ་སྟེ་ ཉན་བཙུག་ཞིནམ་ལས་ ལོག་སྟེ་ བྱེ་བཙུག་ནི། དེ་གི་དམིགས་ཡུལ་གཙོ་བོ་ཡང་ ཉན་ཏེ་ཉ་གོ་ནི་དང་ བྱེ་ཚུགས་མི་ཚུགས་ཀྱི་དཔྱ་ཞིབ་འབད་ཞིའི་དོན་ལུ་ཨིན།

རྒྱགས་ཀྱི་ཚད་གཞི།

༤	གོ་བ་ལེན་ཐངས། (གནད་དོན་ཆ་ཚང་ ལྟག་ཆད་ཞུགས་མ་ཞུགས)
༤	ཡིག་སྟེབ། ཡིག་སྟེབ་འཚོལ་བ་མེདཔ། སྐྱོར་བ་དག་པ།
༣	ཚིག་སྐྱོར། (མིང་ཚིག་དང་ ཁྱད་ཚིག་ ཚིག་གྲོགས་ཚུ་ གོ་རིམ་ ཚུལ་མཐུན་ལག་ལེན་འཐབ་ཐངས།)
༣	འབྲི་ཐངས། (ཚིག་མཚམས་དང་རྩོད་མཚམས་བཞག་ཐངས།)
༡	ཡིག་བཟོ། ཡི་གུའི་བཟོ། སྐྱེལ་ཐངས། བཀོད་རིས།

༢ ལས་འགུལ་གཉིས་པ། སྤྲལ་ཞིའི་རིག་རྩལ་དཔྱ་ཞིབ། ： ༢༠%

སློབ་སྟོན་པ་གིས་ འཐུས་མི་ ༩ དང་ ༡ གི་བར་ན་འབད་མེའི་སྟེ་ཚན་རེ་བཟོ་སྟེ་ མི་སྟེའི་ནང་ལུ་ རྩོལ་བའི་སྐབས་ལུ་ སྐྱོན་གཅིར་ འཐུང་དོ་ཡོད་པ་ཞིན་ན་ ལྟ་རྟོག་གི་ཐོག་ལས་ བསྟུ་ལེན་འབད་ཞིན་ན་ སློབ་ཁང་ནང་ལུ་ སྐར་མ་ ༡༥ གི་རིང་ལུ་ སྤུན་ཞུ་འབད་བཅུག་ནི། སྟེ་ཚན་གྱི་འཐུས་མི་ཆ་མཉམ་གྱིས་ བཅའ་མར་ གཏོར་ཚུགས་ཆུགས་པའི་ཐབས་ཤེས་བརྟེན་དགོ། དེ་གི་དམིགས་ཡུལ་གཙོ་བོ་ སྤུབ་ནིའི་རིག་རྩལ་དང་ འཐེ་ནིའི་རིག་རྩལ་ རྩལ་ཁྱི་དེ་ཞིབ་འབད་ནིའི་དོན་ལུ་ཞིན།

སྐྱོགས་ཀྱི་ཚད་གཞི།

- ༣ དོན་ཚན། (ཁ་གསལ་དང་འཚེལ་བ།)
- ༣ སྐད་ཡིག། (ཡིག་སྟེབ་དང་ཐད་ དཔོག་སྟོན་ལུ་སྐྱོན་མེད།)
- ༡ སྤུན་ཞུའི་བཀོད་རིས།
- ༡ དུས་ཡུན།

གསལ་བཤད་འབད་ཐངས།

- ༣ ཐོགས་ཚགས་མེད་པ།
- ༣ རྩོད་སྤྲ་དག་པ།
- ༡ སྐད་ཀྱི་སེངཁབ།
- ༡ གཟུགས་ཀྱི་རྣམ་འགྱུར།

ག ལས་འགུལ་གསུམ་པ། ལྟག་ཏེ་དང་བྱེ་ནིའི་རིག་རྩལ་དབྱེ་ཞིབ། ： ༡༠%

ཚོག་འཐུ་ ༡༧༠༠-༢༠༠༠ གི་བར་ན་ཡོད་པའི་རྩོམ་བྱེས་ཅིག་ སློབ་དཔོན་དང་སློབ་ཕྲུག་མཉམ་རུབ་ཀྱི་ཐོག་ལས་ གདམ་ཁ་ཆུབ་ཞིན་ལས་ རངས་ཆུང་གི་ཐོག་ལས་ དབྱེ་དཔྱད་སྟེན་ཞུ་ ཚོག་འཐུ་ ༡༥༠༠- ༣༠༠༠་གི་བར་ན་ བྱེ་ནི། དེ་གི་དམིགས་གཏད་འདི་ སློབ་ཕྲུག་ཚུ་གིས་ ལྟག་སྟེ་གོ་དོན་ལེན་ཏེ་ བཅུད་དོན་ཅི་ཞེ་ཡིན་དང་ དབྱེ་དཔྱད་འབད་ཐངས་ཀྱི་རིག་རྩལ་ཐོབ་སྟེ་ ཡིག་ཐོག་ལུ་ བཀོད་ཚུགས་ནིའི་དོན་ལུ་དང་ ལུང་འདེན་དང་ཆུབ་རྐེན་ལག་ལེན་འཐབ་ཐངས་ཚུ་ དབྱེ་ཞིབ་འབད་ནི་ཞིན། སློབ་ཕྲུག་རེ་གིས་ རྩོམ་བྱེས་ཀྱི་བཅུད་དོན་རེ་དང་ དབྱེ་དཔྱད་སྟན་ཞུ་རེ་བྱེ་དགོས་ཞིན།

སྐྱོགས་ཀྱི་ཚད་གཞི།

- ༥ རྩོམ་བྱེས་ཀྱི་བཅུད་དོན། (མགོ་མཇུག་བར་གྱི་གནད་དོན་གཅོ་བོ་བཀོད་ཐངས།)།
- ༤ དབྱེ་དཔྱད་འབད་ཐངས། (གཏན་སྡུར་དང་བསམ་འཆར་ ཁྱངས་བཀལ་ཐངས།)
- ༤ སྟེབ་སྦྱོར། (ཡིག་སྟེབ་དང་ཐད་རྣམ་དབྱེ།)
- ༣ གནད་དོན་གོ་རིམ་སྒྲིག་ཐངས།
- ༣ འོས་འབབ་ལྷན་པའི་མིང་ཚིག་ལག་ལེན་འཐབ་ཐངས།
- ༢ ལུགས་མཐུན་དང་འཁྲིལ་ཏེ་འབྲི་ཐངས།
- ༢ ཚིག་སྦྱོར་ཚུལ་མཐུན་ལག་ལེན་འཐབ་ཐངས།
- ༣ ལུང་འདེན་དང་ཆུབ་རྐེན་ལག་ལེན་འཐབ་ཐངས།
- ༢ འབྲི་བཀོད།

ང སློབ་དུས་ཚོས་རྒྱུགས། ： ༡༠%

ལས་རིམ་འདི་ནང་བཅའ་མར་གཏོགས་མི་ཚུ་གིས་ སློབ་དུས་མཇུག་བསྡུཔ་དང་ དུས་ཡུན་ཚུ་ཚོད་ ༡ ཀྱི་རིང་ལུ་ སྐུགས་ ༡༠ གི་འབྲི་རྒྱུགས་ཕུལ་དགོཔ་ཨིན།

དན་གསོ། འབྲི་རྒྱུགས་ཀྱི་སྐུགས་དང་དུས་ཡུན་ཚུ་ཚོད་ཚུ་ རང་སོའི་མཐོ་རིམ་སློབ་གྲྭའི་ལམ་ལུགས་དང་འཁྲིལ་ཏེ་ལེན་ཞིནམ་ལས་ སྐུགས་བརྒྱ་ཆ་ ༡༠ ནང་ལུ་ ཐབ་དགོཔ་ཨིན།

དབྱེ་ཞིབ་ཐབས་ལམ་དང་སྤྱིད་ཚད་ཀྱི་བཀོད་རིས།

ཐབས་ལམ།	**དབྱེ་ཞིབ་ཀྱི་དབྱེ་བ།**	**གྲངས་ཁ།**	**སྐུགས་ཀྱི་བརྒྱ་ཆ། (%)**
དུས་རྒྱུན་དབྱེ་ཞིབ།	ཀ ཉན་ནིའི་རིམ་རྩལ་དབྱེ་ཞིབ།	༡	༢༠
	ཁ སྒྲུབ་ནིའི་རིམ་རྩལ་དབྱེ་ཞིབ།	༡	༢༠
	ག ལྟག་ནི་བྱེ་ནིའི་རིམ་རྩལ་དབྱེ་ཞིབ།	༡	༣༠
སློབ་དུས་ཚོས་རྒྱུགས།	ང འབྲི་རྒྱུགས།	༡	༣༠
སྐུགས་བསྡོམས་སྡེན་ཚད།			**༡༠༠**

ནདནོཏ།

ལས་ཚན་དང་པ། སྐད་ཡིག་གི་ངོ་སྤྲོད།

- ༡.༡ རྩོང་ཁའི་སྐད་ཡིག་གི་ངོ་སྤྲོད།
- ༡.༢ སྤྱི་བཀྲ་ རྩོང་ཁ་ལྟབ་དགོ་པའི་དགོས་པ།
- ༡.༣ དམིགས་བསལ་ མཐོ་རིམ་སློབ་གྲྭ་ནང་ལུ་ རྩོང་ཁའི་སྤྱོད་ཚན་འདི་ ལྟབ་དགོ་པའི་དགོས་པ།

ལས་ཚན་གཉིས་པ། བརྡ་དོན་འཕྲུལ་རིག།

- ༢.༡ གློག་རིག་ནང་ རྩོང་ཁའི་ཡིག་གཟུགས་བཙུགས་ཐངས།
- ༢.༢ གློག་རིག་ནང་ལུ་ རྩོང་ཁ་འབྲི་ཐངས།

ལས་ཚན་གསུམ་པ། ཡི་གུའི་སྤྱོད་པ།

- ༣.༡ ཚོགས་གྲོགས།
- ༣.༢ འབྲེལ་ཚོགས་ལག་ལེན་འཐབ་ཐངས།
- ༣.༣ འབྲེལ་སྦྲ་
- ༣.༤ བྱེད་སྒྲ
- ༣.༥ ལྟག་བཅས།

ལས་ཚན་བཞི་པ། འབྲི་རྩལ་གཞི་རྩ།

- ༤.༡ ཚག་ཤད་ལག་ལེན་འཐབ་ཐངས།
- ༤.༢ མགོ་རྒྱན་གྱི་འཐོབ་ལམ།
- ༤.༣ རྩོད་ཚོགས་འབྲི་ཐངས།

        ๔.๔ དོན་མཚམས་གཅད་ཐངས།

ལས་ཚན་༥་པ། ཤེས་ཡོན་འཛི་རྩལ།

        ๕.๑ ལུགས་མཐུན་དང་ ལུགས་ཡངས་ཀྱི་འཛི་ཐངས།

        ๕.๒ ལུང་འདེན་ འབད་ཐངས།

            ๕.๒.๑ ཚོག་ལུང་།

            ๕.๒.๒ དོན་ལུང་།

        ๕.๓ རྒྱབ་རྟེན་ཐོ་བཀོད་འབད་ཐངས།

            ๕.๓.๑ དཔེ་དེབ་ཐོ་བཀོད་འབད་ཐངས།

            ๕.๓.๒ གནས་དེབ་ཐོ་བཀོད་འབད་ཐངས།

ལས་ཚན་༦་པ། རྩོང་ཁའི་ངག་གཤིས་དང་འཁྲིལ་ཏེ་སྒྲུབ་ཐངས།

        ๖.๑ ཁ་བཟང་།

        ๖.๒ ཨལ་སྐད་ཞེ་ས།

        ๖.๓ རྩོང་ཁའི་ཚོག་སྡོུར་ཚུ་ དོས་འབབ་ལྟནམ་སྤྲ་ ལག་ལེན་འཐབ་ཐངས།

        ๖.๔ དཔེ་གཏམ་ལག་ལེན་འཐབ་ཐངས།

        ๖.๕ རྩོང་ཁ་ཉག་རྒྱང་གི་མིངཚོག་ལག་ལེན་འཐབ་ཐངས།

ལས་ཚན་༧་པ། རྩོང་ཁའི་ངག་གཤིས་དང་འཁྲིལ་ཏེ་ལྟག་ཐངས།

        ๗.๑ ཚོག་མཚམས་བཅད་དེ་ལྟག་ཐངས།

        ๗.๒ མགོ་འདོགས་དབྱངས་གསུམ་ཞུགས་པའི་རྩོད་སྒྲ།

        ๗.๓ རྩེས་འཇུག་གི་སྒྲ་ཉིལ་བུ་བཏེག་དགོས་དང་མ་དགོ་པའི་རིགས་ཚུ་ ཁྱད་པར་ཕྱེ་སྟེ་ ལྟག་ཐངས།

        ๗.๔ རྩེས་འཇུག་མེད་རྩྭངཡོད་པ་བཟུམ་ལྟག་ཐངས།

        ๗.๕ མིང་མཐའ་མེད་རྩྭངཡོད་པ་བཟུམ་ལྟག་ཐངས།

ལས་ཚན་བཅུད་པ། ཡིག་འགྱུལ།

        ๘.๑ ཡིག་ཆུང་འཛི་ཐངས།

        ๘.๒ མཐོན་ཞུ་འཛི་ཐངས།

        ๘.๓ གཏུངཡིག་འཛི་ཐངས།

        ๘.๔ ཞུ་ཡིག་དང་ཞུ་ཚོག་/བཤེར་ཡིག་ འབྲི་ཐངས།

        ๘.๕ གནོན་ཡིག་འཛི་ཐངས།

        ๘.๖ ཁྲབ་བསྒྲགས་ཀྱི་རིགས་འབྲི་ཐངས།

        ๘.๗ སྐྱན་ཞུ་འབྲི་ཐངས།

        ๘.๘ འབྲི་ཤོག་གི་རིགས་བཀག་ཐངས།

ལྟག་དགོ་པའི་དཔེ་ཐོ།

རེས་པར་དུ་ལྟག་དགོ་པའི་དཔེ་ཐོ།

ཀུན་བཟང་རྡོ་རྗེ། (২০১২) རྩོད་གའི་རྩོད་སྐྱ་ངག་རྒྱུན་དག་པའི་ཁུས་ཚུ། རྩོད་ཁ་གོང་འཕེལ་ལྟན་ཚོགས།

ཀུན་བཟང་རྡོ་རྗེ། (২০১৫) དཔེ་གཏམ་དོན་ཀྱི་རྒྱུན་ཚ། རྩོད་ཁ་གོང་འཕེལ་ལྟན་ཚོགས།

རྩོད་ཁ་གོང་འཕེལ་ལྟན་ཚོགས། (২০২২) འབྲུག་གི་ཡིག་བསྒྱུར་རྣམ་གཞག། རྩོམ་པ་པོ།

རྩོད་ཁ་གོང་འཕེལ་ལྟ་ཚོགས། (২০২২) རྩོད་གའི་ལུང་འདེན་དང་རྒྱབ་རྟེན་བགོད་ཐངས་དཔེ་དེབ། རྩོམ་པ་པོ།

རྩོད་ཁ་གོང་འཕེལ་ལྟ་ཚོགས། (২০২৩) རྩོད་གའི་བརྡ་གཞུང་སྐད་བའི་སྒྲིན་ཚེ། རྩོམ་པ་པོ།

ཁ་སྐོང་ལྟག་དགོ་པའི་དཔེ་ཐོ།

སྐལ་བཟང་དབང་ལྡུག (২০০২) རྩོད་ཁ་བརྡ་དོན་རྒྱུན་འཛེལ། བསམ་རྩེ།

རྩོད་གོང་ལྟན་ཚོགས། (২০১১) ཤལ་སྐད་དང་ཞེ་སའི་རྣམ་གཞག་སྐར་མའི་འོད་ཟེར། རྩོད་ཁ་གོང་འཕེལ་ལྟན་ཚོགས།

རྩོད་ཁ་གོང་འཕེལ་སྡེ་ཚན། (২০২২) ཞིང་རྩོད་ཤན་སྦྱར་མིང་ཚིག་པར་དེབ། སྲོལ་འཛིན་དང་རྩོད་ཁ་གོང་འཕེལ་ལས་ཁུངས།

རྩོད་ཁ་གོང་འཕེལ་སྡེ་ཚན། (২০২২) རྩོད་ཁག་དང་ཆེད་འོག་གཡུས་ཚན་གྱི་མིང་། སྲོལ་འཛིན་དང་རྩོད་ཁ་གོང་འཕེལ་ལས་ཁུངས།

སྤྱི་ཚེས། སྤྱི་ལོ་ ২০২৫ སྤྱི་ཟླ་ ལྟ་པ།

Year 3 Semester I

Module Code and Title: **EMT301 Introduction to Artificial Intelligence & Machine Learning**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Ngawang Tenzin

General Objective
This module aims to provide students with a foundational understanding of the concepts, techniques, tools, and applications of Artificial Intelligence (AI) and Machine Learning (ML). Students will develop core computational and analytical skills to build basic ML models which helps in critically examining the ethical implications and societal impact of AI.

Learning Outcomes
On completion of the module, students will be able to:

i. explain the fundamental concepts of Artificial Intelligence (AI) and Machine Learning (ML)
ii. apply Python programming for data preparation and manipulation
iii. use ML libraries to develop basic models
iv. design supervised and unsupervised learning algorithms
v. evaluate ML model performance using appropriate metrics
vi. define ethical and societal considerations in AI/ML applications
vii. develop small-scale AI/ML solutions addressing real-world needs.

Learning and Teaching Approach

Type	**Approach**	**Hours per week**	**Total credit hours**
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent Study	Assignment/Project	3	60
	Self-study	1	
Total		**8**	**120**

Assessment Approach

A. Class Tests (10%)

One class test will be conducted individually to assess students' understanding of fundamental concepts in Artificial Intelligence (AI) and Machine Learning. These will cover the first 3 Units, focusing on basic definitions, types of learning and data preprocessing steps. The assessments will include multiple-choice questions, true/false questions and short answer questions conducted in class. This test will be conducted out of 30 marks over the duration of one hour and the marks obtained will be converted into 10%.

B. Project (20%)

Students will work in groups of 4 – 5 members to carry out an AI/ML project by analysing datasets relevant to the Bhutanese context, such as weather data, electricity consumption data, or publicly available datasets from the NDSI website. The project will be divided into four parts: Proposal (5%), Model Development (5%), Final Presentation (5%), and Final Report (5%).

i. **Proposal (5%)**

Each group will submit a short proposal explaining what problem they will address, which dataset they will use and how they plan to approach it. It will be assessed based on following criteria:

3 Marks	Problem Definition and approach (Clarity, feasibility and relevance of the problem)
2 Marks	Dataset Selection (Suitability and source of dataset)

ii. **Model Development (5%)**

Students process the dataset and develop a working model using AI/ML techniques. Focus on data preparation, model implementation and preliminary evaluation. It will be assessed based on following criteria:

3 Marks	Data Preparation (Cleaning, formatting and understanding the data)
2 Marks	Model Implementation (Correct application of chosen methods)

iii. **Final Presentation (5%)**

Students will present their project explaining the problem, dataset, model development and key findings. A demonstration of the model is included. It will be assessed based on following criteria:

3 Marks	Clarity of Presentation (Organisation, teamwork and explanation)
2 Marks	Demonstration of Model (Visual or live demonstration)

iv. **Final Report (5%)**

Each group must submit a concise report of approximately 500–750 words, documenting the problem, dataset, model development, results and conclusions. It will be assessed based on following criteria:

3 Marks	Methodology (Problem statement, Description of data handling and model design)
2 Marks	Report presentation and Discussion (Structure, Interpretation and relevance of findings)

C. Practical Examination (10%)

The practical exam of 40 marks over a duration of 2 hours will be conducted and the marks obtained will be converted into 10%. The examination will be conducted after covering approximately 80% of the syllabus to assess students' ability to design, implement, and evaluate supervised and unsupervised learning algorithms, focusing on coding, model

training, data preprocessing, and performance evaluation using Python. The grading will focus on:

5 Marks	Accuracy of implementation (Correct coding, valid outputs, proper use of ML Libraries)
4 Marks	Application of concepts (Ability to apply supervised and unsupervised learning algorithms to practical problems)
4 Marks	Problem-solving and reasoning (Logical approach, efficiency in model design, clarity of steps)
2 Marks	Presentation and clarity (Readable code, neat formatting, structured output, presentation)

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours. It will be converted to 20%.

E. Semester-End Examination (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours. It will be converted to 40%.

Overview of assessment approaches and weighting

Areas of Assignments		**Quantity**	**Weighting**
A. Class Test		1	10%
B. Project	Proposal	1	5%
	Model Development Review	1	5%
	Final Presentation	1	5%
	Report	1	5%
C. Practical Examination		1	10%
D. Mid-Semester Examination		1	20%
E. Semester-End Examination (SEE)		1	40%
Total			**100%**

Pre-requisites: PGL101 Fundamentals of Programming**,** PGL102 Object-Oriented Programming, MTH102 Calculus, MTH203 Statistics and Probability

Subject Matter

Unit I: Foundations of AI and ML

1.1. Definition, scope, and goals of AI
1.2. History and evolution of AI
1.3. Branches of AI
  1.3.1. Robotics, NLP and Computer Vision
1.4. Introduction to ML: Types (Supervised, Unsupervised, Reinforcement Learning)

Unit II: Data, Tools, and Preprocessing

2.1 Python machine learning environment
  2.1.1 Overview of Python ML ecosystem
  2.1.2 Libraries: NumPy, Pandas, Scikit-learn, Matplotlib
2.2 Types of data
  2.2.1 Tabular
  2.2.2 Image
  2.2.3 Text
2.3 Data preprocessing
  2.3.1 Data cleaning
  2.3.2 Handling missing values

2.3.3 Normalization and standardisation
2.3.4 Feature scaling: MinMax, Standard scaling

2.4 Handling categorical data
2.4.1 Label encoding
2.4.2 One-hot encoding

2.5 Dataset preparation
2.5.1 Feature selection
2.5.2 Train-test split
2.5.3 Concepts of overfitting and underfitting

Unit III: Supervised Learning – Regression

3.1 Concepts of supervised learning
3.1.1 Regression vs Classification

3.2 Linear regression models
3.2.1 Simple linear regression
3.2.2 Multiple linear regression

3.3 Model training
3.3.1 Cost function: Mean squared error

3.4 Model evaluation metrics
3.4.1 R^2 (Coefficient of Determination)
3.4.2 Mean Absolute Error (MAE)
3.4.3 Root Mean Squared Error (RMSE)

Unit IV: Supervised Learning – Classification

4.1 Introduction to classification
4.1.1 Concepts of classification
4.1.2 Binary vs multiclass classification

4.2 Classification algorithm
4.2.1 Logistic Regression and Sigmoid Function
4.2.2 K-Nearest Neighbours (KNN)
4.2.3 Naïve Bayes Classifier
4.2.4 Decision Trees and Random Forests
4.2.5 Support Vector Machine

4.3 Model evaluation metrics
4.3.1 Accuray
4.3.2 Precision
4.3.3 Recall
4.3.4 F1-Score

4.4 Confusion Matrix interpretation

Unit V: Unsupervised Learning

5.1 Concept of Unsupervised Learning

5.2 Clustering Algorithms
5.2.1 K-Means Clustering
5.2.2 Hierarchical Clustering
5.2.3 Density-Based Clustering

5.3 Dimensionality Reduction
5.3.1 Introduction to Principal Component Analysis (PCA)
5.3.2 Concept of variance and dimensionality reduction

Unit VII: Basic Neural Networks

6.1. Perceptron and basic ANN structure
6.2. Activation functions: Sigmoid, ReLU
6.3. Introduction to deep learning concepts

List of Practicals

i. Environment Setup – Install Python, Jupyter Notebook, and ML libraries; explore the ML environment
ii. Data Preprocessing – Load datasets, clean data, handle missing values, scale features, and encode categorical data.
iii. Dataset Preparation – Select features, split data for training/testing, and understand overfitting/underfitting.
iv. Regression Models – Implement linear regression and evaluate model performance
v. Classification Models – Build classifiers (Logistic Regression, KNN, Decision Trees, etc.) and evaluate using standard metrics.
vi. Clustering – Apply K-Means, Hierarchical, and Density-Based clustering; interpret cluster results
vii. Dimensionality Reduction – Apply PCA to reduce features and analyze variance

Reading List
Essential Reading

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras & TensorFlow: Concepts, tools, and techniques to build intelligent systems* (3rd ed.). O'Reilly Media.

Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

Additional Reading

Alpaydin, E. (2020). *Introduction to machine learning* (4th ed.). The MIT Press.

McKinney, W. (2022). *Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter* (3rd ed.). O'Reilly Media.

Date: May 2026

Module Code and Title: **PGL304 Mobile App Development**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Arun Kumar Singh

General Objective
This module equips students with the knowledge and practical skills to create modern mobile applications. Building on prior experience in programming and software development, it focuses on mobile architecture, user-centred design, and the use of contemporary development frameworks in building efficient, secure, and high-performing applications through real-world project experience.

Learning Outcomes
On completion of the module, students will be able to:
i. explain the architecture and components of modern mobile computing platforms
ii. design responsive and user-centred mobile user interfaces
iii. develop mobile applications using contemporary development frameworks and tools
iv. implement event-driven programming techniques in mobile applications
v. implement local data storage and persistence mechanisms in mobile applications
vi. integrate mobile applications with RESTful APIs and JSON data formats
vii. implement mobile application features using device hardware and location services
viii. apply security practices in mobile applications

ix. test mobile applications using appropriate testing tools
x. deploy mobile applications using professional development practices.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Sessions	2	
Independent Study	Assignments	3	60
	Self-Study	1	
Total		**8**	**120**

Assessment Approach

A. Class Test (10%)

Two short written tests (20 marks each, 30 minutes duration) will assess students' conceptual understanding of mobile computing principles, application architecture, and development frameworks. The first test will cover Units I–IV. The second test will be conducted after the mid-semester examination and will cover Units V–IX to assess students' understanding of the subsequent portion of the syllabus.

B. Laboratory Portfolio (30%)

The Laboratory Portfolio consists of guided practical exercises conducted throughout the semester. Students will progressively develop a mobile application by applying the concepts and techniques covered in the module. Weekly practical submissions will be reviewed during laboratory sessions to monitor progress and provide continuous feedback.

Students are required to maintain organised source code, implementation records, screenshots of outputs, and brief documentation for each practical task. The final assessment will be based on the completed integrated mobile application and the supporting portfolio documentation submitted at the end of the semester.

10 Marks	Functionality and implementation (Mobile application features operate correctly; user interface components, navigation, data handling, and outputs function as intended with minimal errors)
8 Marks	Application of development concepts (Appropriate implementation of mobile application development concepts including layouts, navigation mechanisms, event handling, local storage/database usage, and API integration)
6 Marks	Code organisation and quality (Clear program structure, meaningful naming conventions, modular design, readability, and maintainable programming practices)
4 Marks	Completeness of required features (All required practical tasks and application functionalities are implemented according to the specified requirements)
2 Marks	Documentation and presentation (Well-organised project files, screenshots of outputs, and concise documentation describing implemented features and application behaviour)

C. Mobile Application Project (30%)

Students will work in groups of 3 – 4 members to design, develop, and present a functional mobile application that addresses a realistic problem scenario. The project will be developed progressively throughout the semester and should demonstrate the integration of user interface design, application logic, local data storage, API integration, and device features where appropriate.

The project consists of three phases:

i. System Design and Planning

ii. Application Development and Implementation
iii. Presentation and Demonstration

Each phase will be assessed separately to ensure that student effort at every stage of the project is appropriately recognised and objectively evaluated. Individual contribution and teamwork will be considered throughout the project assessment.

Phase I: System Design and Planning (5%)
This will be assessed out of 10 marks and converted to 5%.

3 Marks	Requirement analysis and problem identification (Clear understanding of the problem scenario, user requirements, and application objectives)
3 Marks	System design and interface planning (Appropriate application workflow, interface mock-ups, navigation structure, and database/API planning)
2 Marks	Feasibility and solution planning (Realistic and appropriate proposed solution with suitable development strategy and technology selection)
2 Marks	Organisation (submission of well-organised work)

Phase II: Application Development and Implementation (15%)

5 Marks	Functionality and implementation (Mobile application features operate correctly; user interface components, navigation, data handling, and outputs function as intended with minimal errors)
3 Marks	Application of development concepts (Appropriate implementation of mobile application development concepts including layouts, navigation mechanisms, event handling, local storage/database usage, and API integration)
3 Marks	Coding (Clear program structure, meaningful naming conventions, modular design, readability, and maintainable programming practices)
2 Marks	Completeness of required features (All required practical tasks and application functionalities are implemented according to the specified requirements)
2 Marks	Individual contribution and teamwork (Active participation, collaboration, equal contribution, and effective coordination among group members during development activities)

Phase III: Presentation and Demonstration (5%)
This will be assessed out of 10 marks and converted to 5%.

3 Marks	Presentation and demonstration (Clear explanation and effective demonstration of application features, workflow, and implemented functionalities)
3 Marks	Communication and response to questions (Confidence, clarity of explanation, technical understanding, and appropriate responses during viva voce)
2 Marks	Teamwork and collaboration (Effective coordination, balanced participation, and professional group interaction during presentation)
2 Marks	Language (Clear, fluent, and professional oral communication using appropriate technical terminology, accurate grammar, and effective pronunciation.

D. Semester-End Examination (30%)

The Semester-End examination will be conducted for 100 marks over a duration of three hours. It will be converted to 30%.

Overview of Assessment Approaches and Weighting

Areas of Assessment	Quantity	Weighting (%)
A. Class Tests	2	10

B. Laboratory Portfolio	1	30
C. Mobile Application Project	1	30
D. Semester-End Examination (SEE)	1	30
Total		**100**

Pre-requisites: WEB202 Dynamic Front-end Development

Subject Matter

Unit I: Introduction to Mobile Computing
1.1 Introduction to mobile computing
- 1.1.1 Evolution of mobile computing
- 1.1.2 Characteristics of mobile devices
- 1.1.3 Mobile computing ecosystem

1.2 Mobile platforms and operating systems
- 1.2.1 Overview of Android platform
- 1.2.2 Mobile operating system architecture

1.3 Types of mobile applications
- 1.3.1 Native applications
- 1.3.2 Hybrid applications
- 1.3.3 Cross-platform applications

1.4 Difference: mobile vs desktop vs web apps

Unit II: Mobile Application Development Environment
2.1 Mobile development tools
- 2.1.1 Android Studio development environment
- 2.1.2 Project structure and configuration files

2.2 Application lifecycle
- 2.2.1 Activity lifecycle
- 2.2.2 Application components

2.3 Development and testing environments
- 2.3.1 Emulators and virtual devices
- 2.3.2 Running applications on physical devices

Unit III: Mobile User Interface Design
3.1 Mobile UI components
- 3.1.1 Text views, buttons, and input fields
- 3.1.2 Image views and layout containers
- 3.1.3 Recycler View

3.2 Layout and responsive interface design
- 3.2.1 Layout structures
- 3.2.2 Responsive UI design principles

3.3 Navigation and usability
- 3.3.1 Navigation between screens
- 3.3.2 Material design principles

Unit IV: Event Handling and Application Logic
4.1 Event-driven programming
- 4.1.1 Event listeners and callbacks
- 4.1.2 Handling user interactions

4.2 Input handling and validation
- 4.2.1 Form input processing
- 4.2.2 Input validation techniques

4.3 Application logic management
- 4.3.1 Activity interaction

4.3.2 Managing application state

Unit V: Mobile Data Storage

5.1 Introduction to mobile data persistence
- 5.1.1 Local storage concepts
- 5.1.2 Data persistence strategies

5.2 Shared preferences and local storage
- 5.2.1 Saving and retrieving application data

5.3 Mobile database systems
- 5.3.1 SQLite database fundamentals
- 5.3.2 Room Database
- 5.3.3 CRUD operations in mobile applications

Unit VI: Networking and Web Service Integration

6.1 Networking in mobile applications
- 6.1.1 HTTP communication
- 6.1.2 Internet connectivity

6.2 Web service integration
- 6.2.1 RESTful APIs
- 6.2.2 API request and response handling

6.3 Data exchange formats
- 6.3.1 JSON parsing
- 6.3.2 Displaying remote data in mobile applications

Unit VII: Device Features and Sensors

7.1 Location services
- 7.1.1 GPS access
- 7.1.2 Location-based services

7.2 Device hardware features
- 7.2.1 Camera integration
- 7.2.2 Media capture and display

7.3 Mobile sensors
- 7.3.1 Accelerometer
- 7.3.2 Sensor-based interactions

Unit VIII: Mobile Security

8.1 Application security concepts
- 8.1.1 Security risks in mobile applications
- 8.1.2 Privacy considerations

8.2 Authentication and authorisation
- 8.2.1 User authentication methods
- 8.2.2 Access control mechanisms

8.3 Secure data handling
- 8.3.1 Secure storage techniques
- 8.3.2 Permission management

Unit IX: Testing and Deployment

9.1 Mobile application Testing
- 9.1.1 Functional testing
- 9.1.2 User interface testing

9.2 Types of testing
- 9.2.1 Unit testing
- 9.2.2 UI testing

9.3 Testing tools
    9.3.1 JUnit for unit testing
    9.3.2 Espresso for UI testing
9.4 Debugging techniques
    9.4.1 Debugging tools
    9.4.2 Error identification and resolution
9.5 Application deployment
    9.5.1 Application packaging
    9.5.2 Deployment and distribution
9.6 App store deployment considerations

List of Practicals

i. Development environment setup and creation of a basic mobile application interface.
ii. Design and implementation of user interface layouts and navigation components.
iii. Implementation of user input controls and event handling mechanisms.
iv. Integration of local data storage and data management features.
v. Development of form validation and user interaction functionalities.
vi. Integration of external data sources or APIs into the application.
vii. Implementation of authentication and security features in the application.
viii. Testing and debugging of application components.
ix. Optimisation and refinement of application functionality and user experience.
x. Final integration and deployment of the mobile application.

Reading List

Essential Reading

Lee, W.-M. (2021). *Beginning Android application development* (5th ed.). Wiley.

Phillips, B., Stewart, C., & Hardy, K. (2023). *Android programming: The Big Nerd Ranch guide* (5th ed.). Big Nerd Ranch Guides.

Additional Reading

Darwin, I. F. (2020). *Android cookbook: Problems and solutions for Android developers* (2nd ed.). O'Reilly Media.

Esposito, D., & Saltarello, A. (2018). *Microsoft .NET: Architecting applications for the enterprise* (2nd ed.). Microsoft Press.

Meier, R. (2018). *Professional Android* (4th ed.). Wiley.

Richards, M., & Ford, N. (2020). *Fundamentals of software architecture: An engineering approach*. O'Reilly Media.

Flutter Developers. (2024). *Flutter documentation*. https://docs.flutter.dev

Date: May 2026

Module Code and Title: **SDP302 Software Engineering**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Sonam

General Objective

This module aims to build a strong foundation in Software Engineering by developing students' understanding of how software systems are created, improved, and sustained in real-world environments. It emphasises critical thinking, problem-solving, and the ability to make informed decisions throughout the software lifecycle. The module also prepares

students to act as responsible professionals by integrating ethical, societal, and contemporary industry perspectives.

Learning Outcomes

On completion of the module, students will be able to:

i. explain the principles and scope of software engineering
ii. apply software development life cycle models to real-world scenarios
iii. document software requirements using appropriate techniques
iv. design software systems using modular and architectural design principles
v. implement software testing strategies to evaluate software quality
vi. manage software maintenance through corrective, adaptive, and perfective practices, while addressing refactoring, version control, and technical debt
vii. evaluate emerging trends in software engineering to assess their implications for professional practice and future methodologies
viii. demonstrate ethical and professional practices in software development.

Learning and Teaching Approach:

Type	Approach	Hours per week	Total Credit Hours
Contact	Interactive lecture and computer laboratory sessions	4	60
Independent Study	Assignments, Project work, Reading and review of class materials	4	60
Total		**8**	**120**

Assessment Approach

A. Class Test (10%)

Students will take a written class test on the topics covering the foundational concepts of software engineering principles, myths, and process models from Units 1 and 2 for a duration of 35 minutes of maximum 30 marks, and the marks obtained will be converted into 10%. The test includes multiple-choice, short-answer, and diagram-based questions.

B. Assignment (15%)

Students are required to complete an individual assignment in the form of a written report of 800–1000 words. The report must demonstrate understanding of software engineering concepts, software process models, and requirements engineering. Students should integrate theoretical knowledge with practical examples and reflect on ethical and professional considerations in software development. The maximum mark of 30 marks will be assessed and converted to 15%. It will be accessed based on following criteria:

6 Marks	Conceptual Understanding (Clear explanation of SE principles, scope, myths, and professional responsibilities)
7 Marks	Process Models Application (Accurate description and comparison of at least two models, with justification of model selection)
6 Marks	Requirements Analysis (Demonstration of elicitation techniques and prioritization methods for a hypothetical system)
4 Marks	Integration (Logical linkage between principles, models, and requirements)
3 Marks	Ethical and Professional Reflection (Consideration of ethics, inclusivity, privacy, and fairness in requirements and process choice)
2 Marks	Language (Clear, coherent, well-organised presentation, and accurate grammar)
2 Marks	Reference (Intext and End text citation adhering to APA referencing)

C. Project (25%)

This is a group project work (4 – 5 students per group) consisting of three integrated components a written report of 2000 – 2200 words, practical design output and a presentation to be delivered to the class. Students are required to demonstrate comprehensive understanding of Software Engineering concepts by selecting a real-world or simulated system, applying appropriate process models, conducting requirement analysis, and producing a system design with testing and maintenance considerations. The project emphasizes integration of ethical and professional practices throughout the software lifecycle. The maximum mark of 100 marks will be assessed as; 40 marks on report submission, 30 marks on system design and 30 marks on presentation. It will be accessed based on following criteria:

i. Report (40 marks)

8 Marks	Application (Correct use of concepts and models)
7 Marks	Requirement and Feasibility (Realistic elicitation, specification, and prioritization of requirements; proposed solution is achievable)
7 Marks	Integration (linkage between SDLC phases, requirements, design, testing, and maintenance)
6 Marks	Reasoning and Justification (Logical explanation of chosen models, tools, and design decisions)
5 Marks	Ethical and professional consideration (Ethical and Professional Considerations)
4 Marks	Language (Clear, coherent, well-organised presentation, and accurate grammar)
3 Marks	Reference (Intext and End text citation adhering to APA referencing)

ii. Software design (30 marks)

8 Marks	Application (Correct implementation of design principles, diagrams and models)
7 Marks	Feasibility (design output are practical, align with requirement and technically sound)
6 Marks	Integration (seamless connection between requirements, models and design artifacts)
5 Marks	Reasoning (Clear justification of design choice)
4 Marks	Innovation/Creativity (originality, use if emerging tools or sustainable practice)

iii. Presentation (30 marks)

10 Marks	Delivery and Confidence (Clear, engaging, and confident communication of project findings)
6 Marks	Visual Aids (Professional, concise, and relevant slides or diagrams)
7 Marks	Content Integration (Effective linkage of requirements, design, testing, and maintenance in the presentation)
4 Marks	Q and A Handling (Ability to respond to questions with clarity and dept)
3 Marks	Time Management (Presentation delivered within the allocated time frame)

D. Mid-Semester Examination (15%)

The Mid-semester examination is a comprehensive written assessment conducted over three hours for a total of 50 marks.

E. Semester-End Examination (35%)

The Semester-end examination is a comprehensive written assessment conducted over three hours for a total of 100 marks.

Overview of the assessment approaches and weighting:

Area of Assessment	Quantity	Weighting (%)
A. Class Test	1	10

B. Assignment	1	15
C. Project	1	25
D. Mid-Semester Examination	1	15
E. Semester-End Examination (SEE)	1	35
Total		**100**

Pre-requisites: None

Subject Matter

Unit I: Introduction to Software Engineering

- 1.1 Definition and scope of software engineering
- 1.2 Software characteristics and myths
- 1.3 Software engineering vs. programming
- 1.4 Importance in modern computing
- 1.5 Evolution of software engineering as a discipline
  - 1.5.1 Historical milestones
  - 1.5.2 Emerging trends
- 1.6 Ethical foundations in software engineering
  - 1.6.1 Professional responsibilities
  - 1.6.2 ACM/IEEE codes of ethics

Unit II: Software Process Models

- 2.1 Generic process framework
- 2.2 Traditional Models
  - 2.2.1 Waterfall
  - 2.2.2 V-Model
  - 2.2.3 Incremental
- 2.3 Evolutionary Models
  - 2.3.1 Spiral
  - 2.3.2 Prototyping
- 2.4 Agile Methodologies
  - 2.4.1 Scrum
  - 2.4.2 Kanban
  - 2.4.3 XP (Extreme Programing)
- 2.5 Model selection criteria and tailoring
- 2.6 DevOps and Continuous Delivery pipelines
- 2.7 Ethical considerations in process choice
  - 2.7.1 Sustainability
  - 2.7.2 Inclusivity
  - 2.7.3 Social impact of methodologies

Unit III: Requirements Engineering

- 3.1 Types of requirements
  - 3.1.1 Functional and non-functional
- 3.2 Elicitation techniques
  - 3.2.1 Interviews
  - 3.2.2 Workshops
  - 3.2.3 Observation
  - 3.2.4 Questionnaires
  - 3.2.5 Document review

- 3.3 Use cases, user stories, and scenarios
- 3.4 Requirement specification and validation
- 3.5 Requirements traceability and prioritization techniques
  - 3.5.1 MoSCoW
  - 3.5.2 Kano model
- 3.6 Requirements management tools
- 3.7 Ethical and social impact of requirements
  - 3.7.1 Privacy
  - 3.7.2 Accessibility
  - 3.7.3 Fairness in system design

Unit IV: Software Design Principles

- 4.1 Design concepts
  - 4.1.1 modularity
  - 4.1.2 abstraction
  - 4.1.3 cohesion
  - 4.1.4 coupling
- 4.2 Architectural styles
  - 4.2.1 layered
  - 4.2.2 client-server
  - 4.2.3 microservices
- 4.3 Design patterns
  - 4.3.1 Singleton
  - 4.3.2 Factory
  - 4.3.3 MVC

Unit V: Software Testing and Quality Assurance

- 5.1 Testing levels
  - 5.1.1 Unit
  - 5.1.2 Integration
  - 5.1.3 System
  - 5.1.4 Acceptance
- 5.2 Testing techniques
  - 5.2.1 Black-box
  - 5.2.2 White-box
  - 5.2.3 Regression
- 5.3 Test case design and automation tools
- 5.4 Quality assurance practices and software metrics
- 5.5 Continuous testing in CI/CD environments
- 5.6 Non-functional testing
  - 5.6.1 Performance
  - 5.6.2 Security
  - 5.6.3 Usability
- 5.7 Ethical responsibility in quality assurance
  - 5.7.1 Reliability
  - 5.7.2 Safety
  - 5.7.3 User trust

Unit VI: Software Maintenance and Evolution

- 5.1 Types of maintenance

5.1.1 Corrective
5.1.2 Adaptive
5.1.3 Perfective
5.2 Legacy system modernisation
5.3 Refactoring techniques
5.4 Configuration management and version control
5.5 Software sustainability
5.6 Technical debt management
5.7 Continuous evolution in agile/DevOps contexts
5.8 Ethical and social responsibility in maintenance
5.8.1 Long-term sustainability
5.8.2 User impact
5.8.3 Compliance with data privacy/security standards

Unit VII: Emerging Trends in Software Engineering

6.1 Cloud-native development
6.2 Continuous integration/continuous deployment (CI/CD)
6.3 AI-driven software tools
6.4 Low-code/no-code platforms
6.5 Sustainable software engineering

List of Practical(s)

Students will perform practical exercises aligned with the subject matter, including

i. SDLC model application and documentation
ii. Requirement elicitation and SRS preparation
iii. UML and Design Modelling
iv. Design Pattern Implementation
v. Test case design and automation
vi. Version control using Git and GitHub
vii. Software quality analysis using metrics and tools

Reading List

Essential Reading

Fowler, M. (2018). *UML distilled: A brief guide to the standard object modeling language* (3rd ed.). Addison-Wesley.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

Sommerville, I. (2015). *Software engineering* (10th ed.). Pearson.

Additional Reading

Humble, J., & Farley, D. (2010). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.

IEEE Computer Society. (2025). *Guide to the software engineering body of knowledge* (SWEBOK), version 4.0a. IEEE.

Institute of Electrical and Electronics Engineers. (2024). *IEEE software & systems engineering standards essentials collection*. IEEE Xplore.

Martin, R. C. (2008). *Clean code: A handbook of agile software craftsmanship*. Prentice Hall.

Date: May 2026

Module Code and Title: **WEB303 Back-end Development**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Arun Kumar Singh

General Objective

This module equips students with the knowledge and practical skills to design, develop, secure, test, and deploy modern back-end web applications. Students will learn to build RESTful APIs using a mainstream server-side framework, integrate relational databases, implement authentication and authorisation mechanisms, and apply secure coding practices.

Learning Outcomes

On completion of the module, students will be able to:

i. explain the architecture of full-stack web applications and the HTTP request–response model
ii. compare server-side frameworks used for backend development
iii. design relational data models using ER diagrams for web applications
iv. implement SQL database schemas using keys, constraints, and indexes
v. develop RESTful APIs supporting CRUD operations and returning JSON responses
vi. integrate front-end applications with back-end APIs using HTTP-based communication
vii. implement authentication mechanisms for web applications
viii. apply role-based access control in backend systems
ix. apply secure coding practices to mitigate common web vulnerabilities
x. test backend applications using appropriate testing techniques
xi. deploy backend applications to a cloud platform using appropriate configuration practices
xii. document APIs using standard technical documentation practices.

Learning and Teaching Approach

Type	Approach	Hours per Week	Total Credit Hours
Contact	Interactive Lectures	2	30
	Computer Laboratory Session	2	30
Independent Study	Practice assignments, reading, project work	4	60
Total		**8**	**120**

Assessment Approach

A. Quiz (10%)

Two quizzes will be conducted, worth 20 marks each from Unit I, II and Unit V, VI respectively. The duration of the quiz will be 30 minutes.

B. Mid-Semester Examination (10%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours. It will be converted to 10%.

C. Project (30%)

Students will work in groups of 2–3 to design, implement, secure, test, and deploy a small web application with a RESTful backend API. The project integrates database design, API implementation, authentication, testing, and deployment practices covered in the module.

Each group will submit a complete project package that includes the database schema with seed data, the implementation of a RESTful API, and the integration of authentication and security features. The application should be deployed on a cloud platform with appropriate

configuration. In addition, students must provide API documentation and repository documentation, along with a short operational runbook describing the setup, deployment steps, and basic operational procedures for the application.

- 6 Marks Database design (ER model, schema, constraints, seed data)
- 6 Marks RESTful API design (resources, CRUD operations, HTTP status codes)
- 5 Marks Authentication and security implementation (login, protected routes, secure coding practices)
- 5 Marks Deployment and configuration (environment variables, database connection, cloud deployment)
- 4 Marks Testing and error handling (unit/integration tests, validation, resilience)
- 2 Marks Documentation (API documentation, README, Git usage)
- 2 Marks Professional presentation (communication clarity, smooth transition, equal participation)

D. Practical Examination (10%)

In-lab assessment on auth and secure coding where students add login/logout (or JWT), protect an endpoint, and patch a simple vulnerability (e.g., injection/XSS/CSRF).

- 4 Marks Auth flow (authentication, authorisation logic)
- 3 Marks Vulnerability mitigation correctly implemented (input validation, secure storage, error handling)
- 3 Marks Code clarity (readability, documentation, commit practices)

E. Semester-End Examination (40%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours. It will be converted to 40%.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Quizzes	2	10
B. Mid-Semester Examination	1	10
C. Group Project	1	30
D. Practical Examination	1	10
E. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: WEB202 Dynamic Front-end Development, DMS202 Database Programming

Subject Matter

Unit I: Full-Stack and Server-Side Foundations

1.1 HTTP semantics, URLs, headers, status codes; request–response cycle
1.2 Server-side frameworks overview (Express/Django), routing and controllers
1.3 JSON as interchange format; RFC basics; content negotiation

Unit II: Data Access Layer / ORM

2.1 ORM patterns (model definitions, migrations, query APIs)
2.2 Repository/service layers; separation of concerns
2.3 Error handling and transactional consistency

Unit III: RESTful API Design

3.1 Resource naming; CRUD mappings; versioning; hypermedia links (conceptual)
3.2 Pagination, filtering, sorting; bulk operations
3.3 Error responses, validation, and API documentation (OpenAPI)

Unit IV: Front-End Integration

4.1 ORS and security headers; consuming endpoints from SPA
4.2 Handling loading/error states; retries/backoff strategies
4.3 API evolution/versioning and client compatibility

Unit V: Authentication and Authorisation

5.1 Session-based auth vs. token/JWT; password hashing and resets
5.2 Role-based access control (RBAC) and permission checks
5.3 Audit trails and privacy considerations

Unit VI: Security and Hardening

6.1 Input validation and sanitization; output encoding; prepared statements
6.2 Mitigations for OWASP Top 10 risks (e.g., injection, XSS, CSRF, broken auth)
6.3 Secure secrets management and environment configuration

Unit VII: Testing and Quality

7.1 Unit and integration tests; test data seeding; API contract tests
7.2 Automation and coverage basics; CI test steps
7.3 Performance smoke tests (simple probes)

Unit VIII: Deployment and Operations

8.1 Environment setup (dev/staging/prod); env variables; build and release
8.2 Cloud deployment patterns; static assets; SSL/TLS
8.3 CI/CD basics; rollback strategies

Unit IX: Observability, Logging and Runbooks

ix.1 Structured logging; error tracking; health endpoints
ix.2 Metrics (response time, error rate, saturation); simple dashboards
ix.3 Operational runbooks; incident response basics

List of Practical(s):

i. Database Schema Design using SQL
ii. Backend Framework Project Setup
iii. ORM-based CRUD Implementation
iv. RESTful API Development
v. Front-End and Backend API Integration
vi. Authentication and Protected Routes Implementation
vii. Secure Coding and Web Vulnerability Mitigation
viii. Backend Testing using Unit and Integration Tests
ix. Cloud Deployment of Backend Application
x. Application Logging and Monitoring

Reading List

Essential Reading

Kleppmann, M. (2017). *Designing data-intensive applications.* O'Reilly Media.
Richardson, L., Amundsen, M., & Ruby, S. (2013). *RESTful Web APIs.* O'Reilly Media.

Additional Reading

Jest. (n.d.). *Getting started.* https://jestjs.io/docs/getting-started

Mozilla Developer Network. (n.d.). *Same-origin policy.* https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy
Mozilla Developer Network. (n.d.). *Web security.* https://developer.mozilla.org/en-US/docs/Web/Security
Testing Library. (n.d.). *Testing library documentation.* https://testing-library.com/docs/
The Twelve-Factor App. (n.d.). *The Twelve-Factor App.* https://12factor.net/

Date: May 2026

Module Code and Title: **HCI302 Human Computer Interaction**
Programme: BSc in Applied Computing and Information Technology
Credit Value: 12
Module Tutor: Mark Justin

General Objective
The module aims to equip students with the theoretical foundation and practical skills to design, prototype, and evaluate user interfaces and interactive systems that are usable, accessible, and inclusive. A key focus is on creating technology that enhances Community Vitality and respects Cultural Diversity in line with GNH philosophy.

Learning Outcomes
On completion of the module, students will be able to:

i. explain the fundamental principles, theories, and models of human–computer interaction (HCI) and user experience (UX) design
ii. describe the cognitive and social factors that influence human interaction with computing systems
iii. apply user-centered design (UCD) principles to identify and document user requirements
iv. conduct user research using interviews, surveys, and observation
v. create low-fidelity prototypes for interactive systems
vi. develop high-fidelity prototypes for desktop or mobile interfaces using appropriate design tools
vii. evaluate the usability of interactive systems using usability testing and heuristic evaluation
viii. analyse cultural contexts, language, and community needs relevant to interactive system design
ix. present interactive system designs aligned with cultural diversity and community vitality principles of gross national happiness (GNH).

Learning and Teaching Approach

Type	Approach	Hours per week	Total Hours
Contact	Interactive Lectures	2	30
	Computer Laboratory Sessions	2	30
Independent Study	Practical assignments, design challenges, independent user research, prototyping using software tools, and preparation of design documentation.	3	45
	Reading and reviewing design guidelines, researching cognitive psychology principles, and analysing case studies of culturally relevant and inclusive design.	1	15

Total	**8**	**120**

Assessment Approach

A. Class Test (10%)

Two class tests will be conducted during the semester. The first-class test will be conducted after completing **Unit I to Unit IV** and will be of **30 minutes duration carrying 20 marks**. It will assess students' understanding of fundamental HCI/UX concepts.

The second-class test will be conducted after completing **Unit V to Unit VIII,** also of **30 minutes duration carrying 20 marks.** Both tests will contain **multiple choice questions (MCQs), true/false items, and short-answer questions** designed to evaluate conceptual understanding and application of HCI/UX principles. The marks obtained in the tests will be converted into 10%.

B. Practical Studio Portfolio (25%)

Students will complete a portfolio made up of selected practical artefacts completed across the semester. These will include artefacts such as personas and empathy map, task analysis or scenarios or use cases, low-fidelity wireframes, heuristic evaluation notes, and accessibility checklist improvements. The following criteria will be used for marking:

8 Marks	Accuracy of artefacts (correct application of UX techniques personas, task analysis, wireframes; adherence to UX principles; logical structure of artefacts)
6 Marks	Coverage (inclusion of required artefacts such as persona, empathy map, scenarios, wireframes, heuristic notes; artefacts address the given design problem)
6 Marks	Reasoning (explanation of design decisions; linkage between user needs and design elements; clarity of design rationale)
5 Marks	Professional presentation (clear organisation of portfolio; proper labelling of artefacts; readability and visual clarity; adherence to submission guidelines)

C. UX Team Project and Portfolio (45%)

Students will work in groups of up to 4 members on a UX project that includes: user research summary and requirements, prototype, evaluation evidence, design rationale with cultural/GNH considerations, and a final UX portfolio artefact documenting the end-to-end process. The following criteria will be used for marking:

10 Marks	User research quality (appropriate research method selection; evidence of user insights; clear problem definition and requirement documentation)
10 Marks	Prototype quality (logical interface structure; usability and interaction flow; appropriate use of low- and high-fidelity prototypes)
10 Marks	Evaluation evidence (proper planning of usability test or heuristic evaluation; data collection and analysis; identification of usability issues)
8 Marks	Cultural/GNH alignment (consideration of local culture, language, and user context; design elements reflecting community vitality or cultural diversity)
5 Marks	UX portfolio documentation quality (clear description of design process; structured presentation of findings and artefacts; professional formatting)
2 Marks	Team professionalism (evidence of individual participation; collaboration within the team; submission of contribution log or reflection)

D. Semester-End Examination (20%)

The Semester-end examination will be conducted for 100 marks over a duration of three hours. It will be converted to 20%.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Class Test	1	10
B. Practical Studio Portfolio	1	25
C. UX Team Project and Portfolio	2	45
D. Semester-End Examination	1	20
Total		**100**

Pre-requisites: WEB101 Foundations of Web Development

Subject Matter

Unit I: Foundations of HCI and UX

1.1 Human–Computer Interaction (HCI)
- 1.1.1. Definition of HCI
- 1.1.2. Disciplines involved in HCI
- 1.1.3. History and evolution of HCI
- 1.1.4. HCI models (conceptual models, mental models)

1.2 User Experience (UX)
- 1.2.1 Definition of UX
- 1.2.2 Components of UX
  - 1.2.2.1 Usability
  - 1.2.2.2 Accessibility
  - 1.2.2.3 Utility

1.3 User-Centred Design (UCD)
- 1.3.1 Principles of UCD
- 1.3.2 Iterative design process model

Unit II: Cognitive and Social Aspects of HCI

2.1 Cognitive Psychology in HCI
- 2.1.1 Human information processing
- 2.1.2 Perception
- 2.1.3 Memory
- 2.1.4 Attention

2.2 Interaction Framework
- 2.2.1 Gulfs of execution
- 2.2.2 Gulfs of evaluation

2.3 Social Interaction in HCI
- 2.3.1 Collaborative systems
- 2.3.2 Communication in interactive systems
- 2.3.3 Groupware concepts

Unit III: User Research and Requirements Gathering

3.1 User Research Methods
- 3.1.1 Interviews
- 3.1.2 Surveys
- 3.1.3 Ethnographic studies
- 3.1.4 Contextual inquiry

3.2 Requirements Analysis Techniques
- 3.2.1 Persona creation
- 3.2.2 Scenarios
- 3.2.3 Task analysis
- 3.2.4 Use cases
- 3.2.5 Empathy mapping

3.3 GNH Alignment
- 3.3.1 Identifying community needs
- 3.3.2 Documenting cultural requirements

3.3.3 Community vitality considerations

Unit IV: Design Principles and Guidelines

4.1 Usability Heuristics
- 4.1.1 Nielsen's ten usability heuristics

4.2 Design Principles
- 4.2.1 Visibility
- 4.2.2 Feedback
- 4.2.3 Constraints
- 4.2.4 Consistency
- 4.2.5 Affordance

4.3 Interaction Styles
- 4.3.1 Command-line interfaces
- 4.3.2 Menu-based interfaces
- 4.3.3 Direct manipulation interfaces
- 4.3.4 Form-based interfaces

Unit V: Prototyping and Wireframing

5.1 Wireframing
- 5.1.1 Low-fidelity sketching
- 5.1.2 Digital wireframes

5.2 Prototyping Tools
- 5.2.1 Introduction to Figma
- 5.2.2 Introduction to Adobe XD

5.3 High-Fidelity Prototyping
- 5.3.1 Desktop interface prototypes
- 5.3.2 Mobile interface prototypes

Unit VI: Usability Evaluation Techniques

6.1 Evaluation Methods
- 6.1.1 Usability testing (planning, execution, reporting)
- 6.1.2 Heuristic evaluation
- 6.1.3 Cognitive walkthrough

6.2 Measuring Usability
- 6.2.1 Performance metrics (time, error rate, GOMS)
- 6.2.2 User satisfaction metrics (SUS)

Unit VII: HCI and GNH: Cultural and Ethical Design

7.1 Accessibility
- 7.1.1 Designing for diverse abilities
- 7.1.2 WCAG basics

7.2 Cultural HCI
- 7.2.1 Local cultural values
- 7.2.2 Aesthetics and language considerations

7.3 GNH-focused Design
- 7.3.1 Cultural diversity and resilience
- 7.3.2 Designing systems for community well-being

Unit VIII: Project and Future Trends

8.1 Mini Project
- 8.1.1 Designing a GNH-aligned system
- 8.1.2 Applying UCD stages in project design

8.2 Future Trends in HCI
- 8.2.1 Ubiquitous computing
- 8.2.2 Gestural interfaces

8.2.3 Voice user interfaces (VUI)

List of Practicals

i. User Research Simulation: Conducting simulated user interviews and creating Personas and Empathy Maps.
ii. Wireframing and Prototyping: Creating low-fidelity paper prototypes and high-fidelity digital prototypes using professional design software.
iii. Heuristic Evaluation: Applying Nielsen's heuristics to critique existing Bhutanese or international systems.
iv. Usability Testing: Planning, running, and analysing data from a controlled usability test.
v. Accessibility Design: Modifying a prototype to meet basic accessibility standards (e.g., colour contrast, keyboard navigation).
vi. GNH-Aligned Design Task: Designing a specific interface element that incorporates a cultural motif or local language convention.
vii. Final UX Portfolio: Documenting the end-to-end design process of the course project.

Reading List

Essential Reading

Rogers, Y., Sharp, H., & Preece, J. (2023). *Interaction design: Beyond human-computer interaction* (6th ed.). Wiley.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human-computer interaction* (6th ed.). Pearson.

Additional Reading

Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.

Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic Books.

Date: May 2026

Year 3 Semester II

Module Code and Title: **CYS301 Introduction to Cyber Security**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Arun Kumar Singh

General Objective

This module aims to provide students with foundational knowledge and practical understanding of cybersecurity concepts, principles and practices. The module aims to develop awareness of cyber threats, vulnerabilities and security measures while equipping students with the skills to protect digital systems, networks and information in an ethical and responsible manner.

Learning Outcomes

On completion of the module, students will be able to:

i. explain fundamental concepts and principles of cyber security
ii. analyse common cyber threats, cyber-crimes, and attack techniques

iii. apply safe computing practices to reduce cyber security risks
iv. implement authentication and access control practices for securing digital systems
v. demonstrate the application of encryption and hashing concepts for protecting sensitive data
vi. evaluate data protection and privacy practices in digital environments
vii. analyse security risks associated with cloud and mobile computing environments
viii. use cyber security tools for protecting systems and networks
ix. explain the role of intrusion detection and prevention mechanisms in cyber defense
x. discuss emerging cyber security challenges in modern digital systems.

Learning and Teaching Approach

Type	**Approach**	**Hours per week**	**Total credit hours**
Contact	Interactive Lectures	4	60
Independent Study	Assignment	2	60
	Self-study	2	
Total		**8**	**120**

Assessment Approach

A. Class Tests (20%)

Two class tests will be conducted individually to assess students' understanding of fundamental concepts in cyber security. The first test will cover Unit I and Unit II and will be conducted before the mid-semester examination. The second test will cover the units following the mid-semester period. Both assessments will include multiple-choice questions, true/false questions, and short-answer questions conducted in class, with each test for a duration of 1 hour and carrying 30 marks each and converted to 10%.

B. Case Study (20%)

Students will work in groups of 3–4 members to complete a case study assessment based on a real-world cyber security incident. Each group will analyse the selected case, identify the threats involved, evaluate the preventive measures taken or overlooked and reflect on possible improvements to prevent similar incidents in the future. It is divided into two parts Case Study Analysis:10% and Presentation:10%

i. **Case Study Analysis (10%)**

Each group will select a real-world cyber security incident and conduct a structured written analysis of the case. The analysis should explain the nature of the incident, identify the security threats involved and examine the preventive measures that were implemented or neglected. It will be assessed based on following criteria:

3 Marks	Content Accuracy (Clear explanation of the incident and security issues)
3 Marks	Application of Concepts (Practical recommendations for improving preventive measures)
2 Marks	Communication Skills (Clarity, confidence, and engagement during presentation)
2 Marks	Organisation (Logical flow of ideas and well-prepared presentation slides)

ii. **Presentation (10%)**

Each group will present their analysis and reflections on the selected case study, highlighting preventive measure lapses or strategies used and suggesting improvements that could prevent similar threats in the future. It will be assessed based on following criteria:

4 Marks	Content Presentation (Clarity, accuracy and relevancy)
2 Marks	Visual support (Clear, professional slides)
2 Marks	Communication (Confident and language clarity)

2 Marks Question and Answer (Adequate and accurate answers to questions)

C. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours. It will be converted to 20%.

D. Semester-End Examination (40%)

The Semester-End Examination will be conducted for 100 marks over a duration of three hours. It will be converted to 40%.

Overview of assessment approaches and weighting

Areas of Assignments		**Quantity**	**Weighting (%)**
A. Class Test		2	20
B. Case Study	Case study Analysis	1	10
	Presentation	1	10
C. Mid-Semester Examination		1	20
D. Semester-End Examination (SEE)		1	40
Total			**100**

Pre-requisites: CCS102 Operating System, NET201 Networking Fundamentals

Subject Matter

Unit I: Introduction to Cyber Security

1.1. Fundamentals of Cyber Security
- 1.1.1. Definition and objectives of cyber security
- 1.1.2. Importance of cyber security in digital environment

1.2. Principles of Information Security
- 1.2.1. Confidentiality
- 1.2.2. Integrity
- 1.2.3. Availability

1.3. Types of Cyber Crimes
- 1.3.1. Targeted cyber attacks
- 1.3.2. Mass cyber attacks
- 1.3.3. Financial and identity-related cyber crimes

1.4. Common Cyber Threats
- 1.4.1. Malware and malicious software
- 1.4.2. Phishing attacks
- 1.4.3. Social engineering

1.5. Case Studies: Real-world examples of cyber incidents

Unit II: Safe Practices and Preventive Measures

2.1. Safe Computing Practices
- 2.1.1. Safe browsing practices
- 2.1.2. Identifying suspicious emails and links

2.2. Password and Authentication Mechanisms
- 2.2.1. Password management strategies
- 2.2.2. Multi-factor authentication

2.3. Access Control Concepts
- 2.3.1. Authentication
- 2.3.2. Authorisation
- 2.3.3. Accounting

2.4. Securing Personal Devices
- 2.4.1. System updates and patch management
- 2.4.2. Use of antivirus and firewall protection

2.5. Secure Online Transactions
- 2.5.1. Risks in digital payments

2.5.2. Safe practices for online transactions
2.6. Reporting Cyber Incidents

Unit III: Data Protection and Cryptography Basics

3.1. Sensitive Data and Information Security
3.1.1. Personal and organizational data
3.1.2. Types of sensitive information
3.2. Data Backup and Recovery
3.2.1. Local backup methods
3.2.2. Cloud backup methods
3.2.3. Data restoration procedures
3.3. Privacy and Data Protection
3.3.1. Principles of privacy protection
3.3.2. Organisational responsibility for data protection
3.4. Encryption Fundamentals
3.4.1. Symmetric encryption
3.4.2. Asymmetric encryption
3.4.3. Hash functions
3.5. Secure Use of Cloud Services
3.5.1. Cloud services concepts
3.5.2. Security risks in cloud environment

Unit IV: Cyber Security Tools and Technologies

4.1. Security Tools
4.1.1. Antivirus and anti-malware tools
4.1.2. Firewalls and network protection mechanisms
4.1.3. Password management tools
4.2. Secure Communication Practices
4.2.1. Email security practices
4.2.2. Secure file sharing
4.3. Network Security Fundamentals
4.3.1. Wireless network security
4.3.2. Security risk in public networks
4.4. Intrusion Detection and Prevention
4.4.1. Intrusion detection systems
4.4.2. Intrusion prevention systems
4.5. Mobile Device Security
4.5.1. Security risks in mobile applications
4.5.2. Secure use of smartphones and module services
4.6. Secure Use of Collaboration Platforms

Unit V: Emerging Cyber Security Challenges

5.1. Emerging Trends in Cyber Security
5.1.1. Internet of Things security
5.1.2. Artificial Intelligence in cyber security
5.1.3. Cloud security challenges
5.1.4. Mobile security threats
5.2 Cyber Security Awareness and Continuous Learning
5.3 Cyber Security Challenges in Modern System
5.3.1 Security in cloud environments
5.3.2 Security risks in IoT systems
5.3.3 Security challenges in AI-driven systems

Reading List

Essential Reading

Hadnagy, C. (2018). *Social engineering: The science of human hacking*. Wiley.

Singer, P. W; Friedman, A. (2014). *Cybersecurity and cyberwar: What everyone needs to know*. Oxford University Press.

Microsoft. (n.d.). *Cybersecurity basics*. Retrieved from https://learn.microsoft.com/en-us/security/

Additional Reading

Grimes, R. A. (2017). *Cybersecurity for beginners*. CreateSpace Independent Publishing.

Kaspersky. (n.d.). *Cybersecurity tips and resources*. Retrieved from https://www.kaspersky.com/resource-center

National Cyber Security Centre. (n.d.). *Cyber security guidance and resources*. Retrieved from https://www.ncsc.gov.uk

Date: May 2026

Module Code and Title: **BIS301 Data Analytics and Business Intelligence**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Mark Justin

General Objective

This module equips students with advanced skills in data analytics and business intelligence (BI) to support evidence-based decision-making. It focuses on data handling to enabling students to transform raw data into actionable insights for strategic and operational contexts.

Learning outcomes

On completion of the module, students will be able to:

i. explain the role of data analytics and business intelligence in organisational decision-making
ii. prepare datasets by applying data cleaning and transformation techniques using analytical tools
iii. perform exploratory data analysis to identify patterns and relationships in datasets
iv. apply statistical and predictive modelling techniques to analyse business data
v. evaluate predictive models using appropriate performance metrics
vi. design interactive dashboards using business intelligence tools
vii. interpret analytical outputs to generate actionable business insights
viii. communicate analytical findings effectively for organisational decision-making
ix. evaluate the challenges and opportunities associated with emerging trends in business intelligence.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Sessions	2	
Independent study	Assignments, Project work	2	60
	Reading and review of class materials	2	
Total		**8**	**120**

Assessment Approach

A. Data Preparation and Visualisation Assignment (10%)

Students will complete an individual analytical task using python on a given dataset. The task includes data cleaning, transformation, exploratory data analysis, and creation of appropriate visualisations. Students will submit an analytical report of 850–1000 words together with the Python script and generated charts.

3 Marks	Data preparation (correct handling of missing values, outliers and inconsistencies, appropriate data transformation and structuring using python)
3 Marks	Data visualisation design (creation of appropriate graphs and charts, clarity, readability and correct choice of visualisation techniques)
2 Marks	Interpretation of analytical results (meaningful explanation of patterns, trends and relationships identified in the dataset)
2 Marks	Organisation and presentation (logical structure of the report, clear headings and coherent explanation

B. VLE Quiz (10%)

Students will complete an individual VLE quiz covering Unit I and Unit II. The quiz will consist of multiple-choice questions, short analytical questions, and basic data interpretation tasks assessing conceptual understanding and practical knowledge of data analytics concepts and data preparation techniques. The quiz will be of 20 marks conducted over a duration of 45 minutes. It will be converted to 10%.

C. Mid-Semester Examination (10%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours. It will be converted to 10%.

D. Project (30%)

Students will work in groups of 4-5, to analyse a real-world dataset and develop a Business Intelligence solution, including data preparation, analytical modelling, and development of an interactive BI dashboard.

i. **Project Proposal (5%)**

Students will submit a short proposal outlining the selected dataset, problem statement, analytical objectives, and proposed BI approach. It will be out of 10 marks and converted to 5%.

3 Marks	Problem definition (clarity of business problem and relevance of the dataset)
5 Marks	Proposed analytical approach (feasibility of data preparation, analytics techniques, and BI solution)
2 Marks	Language (clarity, structure, and quality of writing)

ii. **Mid-Review Progress (10%)**

Groups will present the progress of their project, including dataset preparation, exploratory analysis, and preliminary visualisations.

3 Marks	Data preparation progress (evidence of data cleaning, transformation, and exploratory analysis)
5 Marks	Analytical development (initial modelling or dashboard components demonstrating analytical progress)
2 Marks	Clarity (explanation and logical presentation of progress)

iii. **Project Report (15%)**

Students will submit a project report of 1800 – 2200 words describing the methodology, analysis, dashboard design, and business insights. It will be marked our of 20 marks and converted to 15%

6 Marks	Methodology and Data Analysis (Clear and well-justified methodology; appropriate data preparation and analytical techniques; accurate interpretation of findings.)
5 Marks	Dashboard Design and Visualisation (Effective dashboard design; appropriate selection and presentation of visualisations; usability; and alignment with the analytical objectives.)
4 Marks	Business Insights and Recommendations (Depth and relevance of business insights; logical conclusions; practical and evidence-based recommendations derived from the analysis.)
3 Marks	Organisation and Structure (Logical organisation of the report; coherent flow of ideas; clear presentation of sections; and effective integration of content.)
2 Marks	Language and Referencing (Clear academic writing; accurate grammar and technical terminology; and correct application of APA 7th edition referencing.)

E. Semester-End Practical Examination (40%)

The Semester-End Practical Examination will assess students' ability to apply data analytics and business intelligence techniques to solve practical business problems using analytical tools and business intelligence platforms. Students will work on a given dataset to perform data preparation, exploratory analysis, predictive analytics, visualisation, and interpretation of analytical results. The examination will be conducted over a duration of three hours.

12 Marks	Data preparation and handling (data cleaning, handling missing values/outliers, and data transformation)
10 Marks	Exploratory data analysis (summary statistics, trend analysis, and use of analytical techniques)
8 Marks	Data visualisation (suitable charts, proper labelling, and visual presentation of data)
5 Marks	Interpretation and business insights (interpretation of findings and business recommendations)
5 Marks	Organisation and reporting (logical workflow, proper labelling, and clear presentation of results)

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Data Preparation and Visualisation Assignment	1	10
B. VLE Quiz	1	10
C. Mid-Semester Examination	1	10
D. Project	1	30
E. Semester-End Practical Examination (SEE)	1	40
Total		**100**

Pre-requisites: PGL101 Fundamentals of Programming, MTH102 Calculus, DMS101 Database Management System

Subject Matter

Unit I: Introduction to Data Analytics and Business Intelligence

1.1. Data analytics in organisational decision-making
  1.1.1. Role of analytics in modern business strategy
  1.1.2. Data-driven decision-making in organisations
1.2. Types of analytics

1.2.1. Descriptive analytics
1.2.2. Diagnostic analytics
1.2.3. Predictive analytics
1.2.4. Prescriptive analytics
1.3. Industry applications of data analytics and business intelligence
1.4. Ethical, privacy, and governance considerations in analytics

Unit II: Business Intelligence Architecture

2.1 Business intelligence architecture and analytical ecosystems
2.2 Data sources and data integration
2.2.1 Internal and external data sources
2.2.2 Data integration processes
2.3 Extract, Transform, Load processes in BI systems
2.3.1 Data extraction techniques
2.3.2 Data transformation methods
2.3.3 Data loading strategies
2.4 Data warehouses and data marts
2.5 Online Analytical Processing concepts and operations
2.6 BI platforms in organisational analytics environments

Unit III: Data Handling and Preparation using Python

3.1 Python libraries for data analytics
3.1.1 NumPy fundamentals
3.1.2 Pandas for data manipulation
3.2 Data wrangling and transformation techniques
3.3 Data quality management
3.3.1 Handling missing values
3.3.2 Outlier detection and treatment
3.3.3 Data normalization
3.4 Working with structured and semi-structured data
3.4.1 CSV data sources
3.4.2 JSON data structures
3.4.3 Application Programming Interfaces
3.5 Exploratory data analysis
3.5.1 Summary statistics
3.5.2 Correlation analysis
3.5.3 Distribution analysis

Unit IV: Predictive Analytics for Business Intelligence

4.1 Regression analysis for business prediction
4.1.1 Simple linear regression
4.1.2 Multiple regression
4.2 Clustering techniques for business segmentation
4.2.1 K-means clustering
4.2.2 Hierarchical clustering
4.3 Time-series analysis for forecasting
4.3.1 Trend analysis
4.3.2 Seasonality patterns
4.3.3 Forecasting models
4.4 Interpretation of predictive results for business insights
4.5 Applications of predictive analytics in business decision-making

Unit V: Data Visualisation and Business Intelligence Tools

5.1 Principles of effective data visualisation and storytelling
5.2 Tableau platform and interface

5.2.1 Data connections
5.2.2 Calculated fields
5.2.3 Filters
5.2.4 Parameters

5.3 Dashboard design principles
5.3.1 Key performance indicators
5.3.2 Interactive dashboards
5.3.3 Drill-down techniques

5.4 Integration of analytical outputs with BI dashboards

5.5 Communicating analytical insights to stakeholders

List of Practical(s)

i. Data Ingestion & EDA in R - Load CSV/JSON; clean missing/outliers; produce EDA plots; narrative of key signals. Output: Artefact: Python script + EDA report.
ii. Feature Engineering & Regression - Build multiple linear regression; check assumptions; compare MAE/RMSE/R^2; residual diagnostics. Output: Model notebook + metrics table.
iii. Classification with Decision Trees/Random Forests - Task: Train/test split; tune hyperparameters; evaluate with accuracy/precision/recall/F1; ROC curves. Output: Confusion matrices + ROC plots + commentary.
iv. Segmentation (K-Means/Hierarchical) - Choose K; silhouette score; describe segments; business actions per segment. Output: Cluster profile sheet.
v. Time-Series Forecasting - Decompose series; build ARIMA or ETS; compare forecast accuracy (MAPE/SMAPE). Output: Forecast charts + error summary.
vi. Tableau Dashboard Build - Connect to dataset; design KPI tiles; add interactivity (filters, parameters, drill-down). Output: Packaged workbook + screenshot storyboard.
vii. Storytelling & Ethics Mini-Reflection - Short write-up on data bias/privacy risks in the project; propose mitigations and governance checks.

Reading List

Essential Reading

Few, S. (2012). *Show me the numbers: Designing tables and graphs to enlighten*. Analytics Press.

Lilja, D. J. (2016). *Linear regression using R: An introduction to data modelling*. Retrieved https://open.umn.edu/opentextbooks/textbooks/399

Mitchell, J. L. (2025). *Leveraging data visualisation to communicate effectively*. PALNI. Retrieved from https://open.umn.edu/opentextbooks/textbooks/1409

Peng, R. D. (2016). *R programming for data science*. Leanpub. Retrieved from https://leanpub.com/rProgramme

Wickham, H., & Grolemund, G. (2017). *R for data science*. O'Reilly Media. Retrieved from https://r4ds.had.co.nz

Additional Reading

Davenport, T. H., & Harris, J. G. (2017). *Competing on analytics: The new science of winning*. Harvard Business Review Press.

OpenIntro. (2021). *Introduction to modern statistics*. OpenIntro. https://www.openintro.org/book/ims/

Power, D. J., Heavin, C., McDermott, J., & Daly, M. (2018). Defining business intelligence and analytics. *Journal of Decision Systems, 27*(sup1), 64–72. https://doi.org/10.1080/12460125.2018.1468570

Tableau Software. (n.d.). *Tableau learning resources*. https://www.tableau.com/learn/training

Date: May 2026

Module Code and Title: **EMT302 Internet of Things (IoT)**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Arun Kumar Singh

General Objective
The module aims to develop students' knowledge and practical skills to design and implement Internet of Things (IoT) systems using interconnected devices and cloud platforms. It emphasises the development of sustainable and locally relevant IoT solutions that support Environmental Conservation and Community Vitality.

Learning Outcomes
On completion of the module, students will be able to:

i. explain the architecture, components, and operational principles of Internet of Things (IoT) systems
ii. identify suitable sensors, actuators, and microcontrollers for IoT-based data acquisition tasks
iii. configure microcontroller-based IoT devices to acquire and process sensor data
iv. implement network communication between IoT devices and servers using common protocols such as MQTT or HTTP
v. integrate IoT devices with cloud platforms for data storage and basic processing
vi. visualise IoT data using simple dashboards or monitoring tools
vii. identify security, privacy, and ethical issues associated with IoT deployments
viii. develop a functional IoT prototype addressing a locally relevant environmental or community problem.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent study	Practical assignments involving programming microcontrollers	3	60
	Independent Study	1	
Total		**8**	**120**

Assessment Approach

A. Laboratory Portfolio (15%)
Students will complete a series of laboratory exercises during Units 2–5 covering IoT device setup, sensor interfacing, device communication, and cloud connectivity. Each exercise will be demonstrated during scheduled laboratory sessions. Evidence such as program code, screenshots/log outputs, and brief explanations will be compiled into a laboratory portfolio.

The final portfolio mark will be calculated based on the average performance across the completed laboratory exercises**.**

6 Marks	Technical correctness and functional outcome (Correct hardware interfacing, working program code, and successful execution of the laboratory task)
4 Marks	Completion of laboratory tasks (All required experiments completed and demonstrated during lab sessions)

3 Marks	Troubleshooting (Ability to debug errors, explain the working of the system, and respond to instructor questions)
2 Marks	Documentation (Clear portfolio submission including code snippets, screenshots/logs, and brief explanations)

B. Practical Examination (20%)

The practical examination will be of one and half hours and carry 50 marks and the marks obtained will be converted into 20%. It will assess students' ability to configure and operate an IoT device in a laboratory environment. Students will be required to interface a sensor with a microcontroller, programme the device to collect data, and transmit the data using a specified communication protocol (e.g., MQTT or HTTP). Students must also verify the successful transmission of data to a cloud platform or monitoring interface.

8 Marks	Hardware interfacing and program implementation (Correct connection of sensor/device and successful execution of the program)
6 Marks	Data communication and system functionality (Successful transmission of data using the specified protocol and verification of data reception)
4 Marks	Troubleshooting and problem solving (Ability to identify and correct errors during implementation)
2 Marks	Understanding and explanation (Ability to explain the working of the implemented solution)

C. IOT Prototype Project (25%)

Students will work in small groups of 2 – 3 members to design and develop a small-scale IoT prototype addressing a locally relevant challenge aligned with Environmental Conservation or Community Vitality (GNH pillars)**.** The project should include sensor data acquisition, device communication, basic cloud integration, and data visualisation.

The project will be assessed in three stages: proposal, progress presentation, and final demonstration**.**

i. **Project Proposal (5%)**

Students will submit a short proposal describing the problem statement, proposed IoT solution, system components, and expected outcomes. It will be assessed for 10 marks and converted to 5%.

5 Marks	Problem definition (Clear description of the problem and its local relevance)
3 Marks	Proposed system design (Identification of sensors, devices, and communication approach)
2 Marks	Feasibility (Realistic implementation plan and expected outcomes)

ii. **Progress Presentation (5%)**

Groups will present the project progress, including initial design, hardware setup, and early implementation results**.** It will be assessed for 10 marks and converted to 5%.

3 Marks	Technical progress (Evidence of prototype development and partial implementation)
5 Marks	System design explanation (Clarity in explaining architecture and workflow)
2 Marks	Teamwork (participation of group members)

iii. **Final Demonstration and Report (15%)**

Students will demonstrate the working IoT prototype and submit a short report describing system design, implementation, testing, and limitations.

a. Final Demonstration (10%)

6 Marks	Functional Prototype and Demonstration (Successful demonstration of a fully functional IoT system, including sensor data acquisition, communication, cloud connectivity, and integration of system components.)
4 Marks	Technical Explanation and Language (Clear explanation and justification of the system architecture, hardware components, communication protocols, implementation decisions, and professional oral communication using appropriate technical terminology.)

b. Report (5%)

The report will be of 700 – 800 words assessed out of 10 Marks and converted to 5%.

5 Marks	System Design and Implementation (Clear and logical description of the system design, implementation process, architecture, diagrams, and supporting code explanations.)
3 Marks	Testing, Evaluation, and Risk Considerations (Evidence of systematic testing, evaluation of system performance, discussion of limitations, and consideration of security, privacy, ethical issues, and mitigation strategies.)
2 Marks	Language and Documentation (Coherent presentation, accurate technical writing, appropriate use of diagrams and figures, and correct referencing where applicable.)

D. Semester-End Examination (40%)

The Semester-End Examination will be conducted for 100 marks over a duration of three hours. It will be converted to 40%.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Laboratory Portfolio	1	15
B. Practical / Lab Test	1	20
C. IOT Prototype Project	1	25
D. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: NET201 Networking Fundamentals, CCS102 Operating System, PGL102 Object-Oriented Programming

Subject Matter

Unit I: IoT Fundamentals and Architecture

1.1. Definition and History
  1.1.1 IoT Ecosystem
  1.1.2 Evolution of IoT
  1.1.3 IoT Application Domains (Industrial, Smart Home, Health)
1.2. IoT Architecture
  1.2.1 Sensing Layer
  1.2.2 Network Layer
  1.2.3 Application / Platform Layer
1.3. Enabling Technologies
  1.3.1 Embedded Systems

1.3.2 Wireless Communication Technologies
1.3.3 Cloud Computing Platforms

Unit II: Sensing, Actuation, and Microcontrollers

2.1 Sensors and Actuators
2.1.1 Types of Sensors
2.1.2 Types of Actuators
2.1.3 Selection Criteria
2.1.4 Interfacing Methods (Analog/Digital)
2.2 Microcontrollers
2.2.1 Introduction to Arduino
2.2.2 Introduction to Raspberry Pi
2.2.3 Basic Programming Structure (Setup and Loop)
2.3 Data Acquisition
2.3.1 Sampling
2.3.2 Resolution
2.3.3 Noise in Sensor Data

Unit III: IoT Communication Protocols

3.1 Short-Range Communication Technologies
3.1.1 Wi-Fi
3.1.2 Bluetooth
3.1.3 ZigBee
3.2 Low-Power Wide-Area Networks (LPWAN)
3.2.1 LoRa
3.2.2 LoRaWAN
3.3 Application Protocols for IoT
3.3.1 MQTT
3.3.2 CoAP
3.3.3 HTTP
3.4 Network Topology
3.4.1 Star Topology
3.4.2 Mesh Topology
3.4.3 Peer-to-Peer Topology

Unit IV: Cloud Platforms and Data Management

4.1 Cloud, Fog, and Edge Computing
4.1.1 Cloud Computing in IoT
4.1.2 Fog Computing
4.1.3 Edge Computing
4.2 Data Storage
4.2.1 Time-Series Databases
4.2.2 Data Storage Strategies
4.3 IoT Cloud Services
4.3.1 Device Connectivity
4.3.2 Data Ingestion
4.3.3 Basic Data Processing

Unit V: Data Analytics and Visualisation

5.1 Data Pre-processing
5.1.1 Data Cleaning
5.1.2 Data Transformation
5.2 Basic Analytics
5.2.1 Rule-Based Triggers
5.2.2 Conceptual Anomaly Detection

5.3 Data Visualisation
5.3.1 Dashboard Design
5.3.2 Real-Time Monitoring

Unit VI: IoT Security, Privacy, and Ethics

1.1 Security Vulnerabilities
6.1.1 Device-Level Security Issues
6.1.2 Network-Level Security Issues
6.1.3 Cloud-Level Security Issues
1.2 Privacy Concerns
6.2.1 Data Aggregation Risks
6.2.2 User Tracking Issues
1.3 Ethical Issues
6.3.1 Surveillance Concerns
6.3.2 Data Ownership
6.3.3 Algorithmic Bias
1.4 GNH Alignment
6.4.1 Responsible IoT Usage
6.4.2 Community Vitality and Sustainability

Unit VII: IoT Application Domains

1.1 Smart Cities, Agriculture, and Health
7.1.1 Smart City Applications
7.1.2 Smart Agriculture Systems
7.1.3 Healthcare Monitoring Systems
1.2 Environmental Conservation Applications
7.2.1 Air Quality Monitoring
7.2.2 Smart Irrigation Systems
7.2.3 Waste Management Monitoring

Unit VIII: IoT Project Implementation

1.1 Project Planning
8.1.1 Problem Identification
8.1.2 System Design and Prototyping
1.2 Testing and Debugging
8.2.1 System Testing
8.2.2 Error Detection and Troubleshooting
1.3 Project Presentation
8.3.1 Prototype Demonstration
8.3.2 Presentation of the GNH-Aligned IoT Solution

List of Practicals

i. Hardware Interfacing: Setting up an Arduino/Raspberry Pi and reading data from basic sensors (e.g., temperature, humidity).
ii. Actuation: Programming the microcontroller to control an actuator (e.g., LED, servo motor) based on sensor input.
iii. Network Communication: Implementing MQTT publish/subscribe using an IoT device and a broker.
iv. Cloud Integration: Connecting the device to a public IoT platform and pushing real-time sensor data.
v. Dashboard Development: Creating a simple web dashboard or mobile view to visualise the collected IoT data.
vi. Final Prototype: Designing, building, and demonstrating a small-scale, IoT system (e.g., a smart irrigation controller or noise pollution monitor).

Reading List

Essential Reading

Bahga, A., & Madisetti, V. (2014). *Internet of things: A hands-on approach.* VPT.

Banzi, M. (2014). *Getting started with Arduino.* Maker Media.

Hanes, D., Salgueiro, G., Grossetete, P., Barton, R., & Henry, J. (2017). *IoT fundamentals: Networking technologies, protocols, and use cases for the Internet of Things.* Cisco Press.

Additional Reading

Amazon Web Services. (n.d.). *AWS IoT Core documentation.* https://docs.aws.amazon.com/iot/

Eclipse Foundation. (n.d.). *MQTT: Message queuing telemetry transport.* https://mqtt.org/

Internet Engineering Task Force. (n.d.). *Constrained application protocol (CoAP).* https://datatracker.ietf.org/doc/html/rfc7252

International Telecommunication Union. (2020). *Sustainable IoT for smart villages: Guidelines and case studies.* ITU Publications.

Smart Village Initiative. (2021). *Reports on sustainable IoT applications in rural development.* https://smartvillages.org/

Date: May 2026

Module Code and Title: **RIT301 Research Methodology**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Mark Justin

General Objective

The module aims to equip students with the ability to design, conduct, and evaluate rigorous research in applied computing and information technology contexts. It emphasises critical thinking, ethical research practice, problem-solving, and evidence-based decision-making while preparing students to investigate real-world technology-related issues systematically. The module also develops students' competence in research planning, data analysis, academic communication, and professional inquiry to support future project work, innovation, and industry-relevant research activities.

Learning Outcomes

On the completion of the module, students will be able to:

i. critique research philosophies and approaches relevant to technology and applied computing
ii. formulate precise research questions, hypotheses, or objectives for IT-related problems
iii. synthesise literature to identify knowledge gaps in emerging technology domains
iv. evaluate contemporary IT research methods, AI-assisted literature review, simulation-based research, blockchain for data integrity, and other emerging approaches
v. design appropriate research strategies for technology-based studies
vi. justify appropriate research designs for prototype/system development or evaluation studies
vii. develop appropriate instruments for collecting IT-specific data
viii. apply suitable sampling techniques appropriate to different research methods
ix. interpret quantitative or qualitative research data using digital tools
x. assess ethical considerations in technology research
xi. develop structured research proposals that integrate technology design and implementation plans.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures, Discussions, and Periodic Seminars	3	45
	Guided Blended Learning and Online Engagement Activities	1	15
Independent study	Assignments	2	30
	Reading academic journals	2	30
Total		**8**	**120**

Assessment Approach

A. Assessment I – Identifying Potential Research Topics and Research Direction (30%)

This assessment is designed to help students identify technology-focused research problem areas relevant to applied computing and industry contexts. Students will progressively refine research ideas through problem identification, preliminary literature exploration, and concept note development. Assessment will focus on relevance of the research problem, literature engagement, clarity of research direction, feasibility, academic communication, and referencing practices.

Task 1 – Problem Identification (5%)

Students will be required to identify potential IT-related research problem areas relevant to emerging technologies, industry needs, organisational challenges, or societal contexts. The task should demonstrate relevance, originality, feasibility, and alignment with applied computing domains. The task will out of 10 marks (which will be converted to 5%) and be graded on the following criteria;

5 Marks Relevance (alignment with applied computing, emerging technologies, industry or societal relevance)

3 Marks Problem clarity and Feasibility (well-defined and understandable problem area)

2 Marks Use of language (clarity, organisation, grammar, and professional academic writing)

Task 2 – Annotated Bibliography (10%)

Students will be required to conduct preliminary literature exploration using academic journals, conference papers, and credible industry sources related to selected problem areas. The annotated bibliography should demonstrate understanding of existing research, technology trends, and potential research gaps. It will be out of 20 Marks and converted to 10%.

4 Marks Literature relevance and source quality (use of credible and recent academic/industry sources relevant to the identified topic)

5 Marks Synthesis and identification of research gaps (ability to compare existing work and identify limitations or gaps)

7 Marks Critical engagement with literature (understanding of concepts, technologies, and critical analysis report of the information)

2 Marks Referencing (appropriate academic APA referencing style)

2 Marks Language (clarity, coherence, academic tone, and writing quality)

Task 3 – Concept Note Development (15%)

Students will be required to prepare a concept note outlining the proposed research problem, rationale, preliminary research objectives/questions, proposed technology-focused solution direction, and potential methodology. The concept note should demonstrate feasibility, relevance, and continuity toward future project work. It will be out of 20 Marks and converted to 15%.

5 Marks Problem statement and rationale (Clarity, relevance, and justification of the selected research problem)

6 Marks Research objectives/questions and alignment with the proposed research direction (Clarity and proper alignment)

3 Marks Feasibility and proposed methodology (Appropriateness and practicality of the proposed approach)

2 Marks Technology relevance and innovation potential (Expression of the relevance, idea and sustainability)

2 Marks Structure and organisation of the concept note (Sequencing, well structured)

2 Marks Referencing (Appropriate academic APA referencing style)

B. Quiz (20%)

Two quizzes will be conducted, each contributing 10% toward the overall module assessment. Quiz 1 will particularly cover Units I–V (research philosophies, approaches, and designs), while Quiz 2 will cover Units VI–X (literature review, research ethics, data collection methods, validity and reliability and sampling). The quizzes will include multiple-choice questions, definition matching, and short answer questions, which will be conducted online via VLE for 20 minutes each worth 20 marks and the marks obtained will be converted into 10%.

C. Assessment III – Research Proposal and Technology Solution Design (50%)

This assessment requires student groups to collaboratively develop a research proposal addressing a technology-focused problem relevant to applied computing and industry contexts. Students will investigate existing solutions, identify research gaps, propose a feasible technology-based solution or prototype concept, and develop an appropriate research and evaluation plan. The assessment is intended to support continuity toward the Mini Project and final-year Capstone Project.

Each group shall be assigned a module tutor/project mentor to guide proposal refinement, monitor progress, facilitate academic support, and coordinate industry engagement activities where applicable. Students may identify industry contacts independently or with guidance from the tutor.

Individual marks may vary based on contribution evidence, participation in consultations, presentation performance, and supervisor observations.

i. Problem Statement and Research Direction

a) review and discuss the research problem areas identified by individual group members in Assessment I

b) select one technology-focused problem area that demonstrates feasibility, industry relevance, and alignment with the group's common research interest
c) refine the selected problem area into a clear and focused research problem statement
d) justify the relevance and significance of the proposed study within applied computing and information technology contexts
e) formulate preliminary research objectives/questions aligned with the proposed study
f) demonstrate the feasibility of the proposed research direction and its suitability for future prototype/system development or evaluation
g) explain the rationale for the proposed research direction and technology-focused solution approach

ii. Literature Synthesis and Proposed Technology Solution

a) critically review existing IT solutions and related studies
b) identify research gaps and limitations in current approaches
c) synthesise findings from academic and industry literature
d) propose an appropriate technology-focused solution, system concept, prototype idea, or implementation approach
e) justify the feasibility and relevance of the proposed solution

iii. Industry Consultation and Proposal Refinement

a) consult relevant industry stakeholders, professionals, or organisational representatives
b) validate the feasibility and relevance of the proposed research direction
c) gather and document industry feedback
d) refine the proposal based on consultation outcomes
e) demonstrate alignment between industry needs and the proposed technology solution

The module tutor/project mentor shall guide students in identifying appropriate industry contacts and facilitate consultation where required. Students may also independently approach relevant industry stakeholders subject to approval and guidance from the module tutor.

Deliverable

Group Research Proposal Report (40%)
Students will be required to submit a group research proposal report that presents a technology-focused research problem, literature synthesis, proposed solution direction, research methodology, evaluation approach, and industry relevance. The report should demonstrate academic rigor, feasibility, innovation potential, and alignment with applied computing and information technology contexts.

The proposal report may be assessed by the module tutor and/or designated academic assessors based on the following criteria:

A group report (1500–2000 words) including:

- Problem statement (linked to Assignment A and industry consultation).
- Literature synthesis of existing IT solutions.
- Proposed solution: conceptual design or prototype idea.
- Suggested methodology for testing/evaluation.
- Ethical and mindful considerations (including GNH perspectives).

- Industry feedback summary and how it shaped the proposal.

5 Marks	Problem statement and rationale (clarity, significance, contextual understanding, industry relevance, and justification of the selected problem area)
5 Marks	Research objectives/questions and alignment with the proposed research direction and solution approach
7 Marks	Literature synthesis and gap analysis (quality of literature review, critical synthesis of existing IT solutions, identification of research gaps, and relevance of sources)
7 Marks	Proposed technology solution/design (innovation, feasibility, appropriateness of the proposed prototype/system concept, and alignment with applied computing contexts)
5 Marks	Methodology and evaluation plan (appropriateness of research methods, implementation strategy, testing/evaluation approach, and feasibility of execution)
3 Marks	Industry consultation and integration of feedback (evidence of stakeholder engagement, relevance of consultation, and incorporation of feedback into proposal refinement)
2 Marks	Ethical and mindful technology considerations (privacy, responsible computing, ethical awareness, and social/organisational impact considerations)
2 Marks	Individual contribution and participation (consultation participation, collaboration, contribution evidence, engagement in group activities, and task responsibility)
2 Marks	Writing quality and academic structure (organisation, coherence, academic tone, clarity of communication, grammar, and formatting)
2 Marks	Referencing and citation quality (appropriate academic citation style, consistency, and source referencing)

B. Proposal Presentation and Defence (10%)

Students will be required to present and defend the proposed research direction, technology solution, and implementation plan before an academic panel and/or industry representatives where applicable. The presentation should clearly communicate the problem context, proposed solution, methodology, feasibility, and expected contribution of the study.

The presentation and defence assessment will evaluate communication skills, proposal clarity, teamwork, technical justification, and the ability to respond to questions professionally.

3 Marks	Presentation structure, clarity, and communication skills (organisation, delivery, confidence, visual presentation quality, and academic communication)
3 Marks	Technical justification, feasibility, and relevance of the proposed solution/design
2 Marks	Team collaboration and professional delivery (coordination, participation balance, professionalism, and engagement during presentation)
2 Marks	Question handling and defence of proposal decisions (ability to justify choices, explain research direction, and respond to panel questions effectively)

Overview of assessment approaches and weighting

Areas of Assessment	Quantity	Weighting (%)

A. Assessment I – Identifying Potential Research Topics and Research Direction	3	30
B. VLE Quiz	2	20
C. Assessment III – Research Proposal and Technology Solution Design	2	50
Total		**100**

Pre-requisites: None

Subject Matter

Unit I: Introduction to research

1.1. Concepts, Goals and Characteristics of research
1.2. Significance of research
1.3. Types of research
1.3.1. Basic vs Applied research
1.3.2. Descriptive Vs Analytical research
1.3.3. Exploratory Vs Explanatory research

Unit II: Foundations and Philosophy of Research

2.1 Nature of knowledge
2.2 Research Philosophies
2.2.1 Positivism/Post-Positivism
2.2.2 Interpretivism/Constructivism
2.2.3 Pragmatism

Unit III: Research Approaches

3.1 Quantitative, Qualitative, and Mixed Methods
3.4 Application of these approaches in technology research
3.6 Advantages and disadvantages of these approaches

Unit IV: Research Designs

4.1. Quantitative designs
4.1.1 Surveys
4.1.2 Experiments
4.2 Qualitative designs
4.2.1 Case study
4.2.2 Phenomenological study
4.2.3 Ethnographic study
4.2.4 Grounded Theory
4.2.5 Narrative study
4.3 Mixed method designs
4.3.1 Concurrent and Sequential designs
4.3.2 Application of these designs in technology research

Unit V: Research Question/Hypothesis

5.1 Understanding research question and hypothesis
5.2 Sources of research question and hypothesis in technology research
5.3 Formulating research question and hypothesis

Unit VI: Literature Review

6.1 Purpose and scope

6.2 Identifying relevant sources
6.2.1. Databases
6.2.2. Journals
6.2.3. Books
6.2.4. Periodicals
6.3 Critical review
6.3.1. Synthesis
6.3.2. Analysis
6.3.3. Critique of Existing IT Literature
6.4 Developing a conceptual model for the study
6.5 Contemporary IT Research Methods
6.5.1 AI-assisted Literature Review
6.5.2 Simulation-based Research
6.5.3 Blockchain for Research Integrity
6.5.4 Other Emerging Methods

Unit VII: Research Ethics and Academic Integrity
7.1 Ethical Protocols
7.2 Academic Integrity

Unit VIII: Data Collection Methods/Tools
8.1 Quantitative method and tools
8.1.1 Survey questionnaires
8.1.2 Experiments
8.2 Qualitative methods and tools
8.2.1 Interview
8.2.2 Observations
8.2.3 Documentary evidence
8.2.4 Field diaries and notes

Unit IX: Validity and Reliability
9.1 Concepts of validity and reliability
9.2 Ensuring validity and reliability
9.3 Trustworthiness and Transferability

Unit X: Sampling Methods
10.1 Concepts of sampling
10.2 Methods of sampling
10.2.1 Probability sampling
10.2.2 Non-probability sampling/Purposive sampling
10.3 Determining the sample size

Unit XI: Data Analysis and Interpretation
11.1 Quantitative data analysis
11.1.1 Descriptive Vs Inferential statistics for IT datasets
11.1.2 Use of software in quantitative data analysis
11.2 Qualitative data analysis
11.2.1 Thematic analysis
11.2.2 Use of software in qualitative data analysis

Unit XII: Project Proposal
12.1 Proposal structure
12.1.1 Problem statement grounded in technology context
12.1.2 Literature review with technology focus
12.1.3 Methodology including prototype/system design

12.1.4 Implementation plan
12.4 Transition to the Project Module

Reading List
Essential Reading

Bryman, A. (2026). *Social research methods* (7th Ed.). Oxford University Press.

Creswell, J. W., & Creswell, J. D. (2022). Research design: Qualitative, quantitative, and mixed methods approaches (6th Ed.). Sage Publications.

Dowson, C. (2015). *Projects in computing and information systems: A student's guide* (3rd Ed.), Pearson Education.

Xu, R., & Peng, J., (2024). *A comprehensive survey of deep research: Systems, methodologies, and applications*. Zhejiang University.

Additional Reading

Fathima, Y. (2025). AI and the future of scientific research: Trends and predictions. *International Journal of Business and Economics Research (IJBER)*, Special Issue, 225-237, e-ISSN: 2455-3921.

Kitchenham, B., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews.* CRC Press.

Kumar, A. (2025). Research methodology and artificial intelligence: An integrated approach. *Inspira-Journal of Commerce, Economics & Computer Science, 11*(2), 111–122.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer.

Date: May 2026

Module Code and Title:	**EDP101 Entrepreneurship**
Programme:	University-wide module
Credit:	12
Module Tutors:	Dipan Pradhan and Madhav Verma
Module Coordinator:	Dipan Pradhan
Home Base:	Gedu College of Business Studies

General objective

This module aims to provide students with a thorough understanding of entrepreneurship and the entrepreneurial ecosystem, highlighting its vital role in socio-economic development. It seeks to equip students with the knowledge, skills, and competencies necessary to identify and evaluate business opportunities, develop innovative business models, and create viable business plans. In addition to theoretical concepts, the module includes practical elements such as workshops, case studies, and group activities, allowing students to apply what they have learned in real-world situations. By cultivating entrepreneurial and leadership qualities, the module prepares students to become effective entrepreneurs and corporate leaders, capable of driving innovation, managing risks, and contributing to economic growth and development.

Learning outcomes

On completion of the module, students will be able to:

i. Apply entrepreneurial concepts and their evolution to real-world situations.
ii. Evaluate the impact of technological entrepreneurship on innovation and business growth through practical examples.
iii. Identify emerging trends to develop entrepreneurial opportunities in both local and global contexts.
iv. Develop entrepreneurial competencies by using self-assessment tools such as GETT and SRQ to identify areas for growth.
v. Identify business ideas that address real-world problems
vi. Evaluate business opportunities using a structured framework
vii. Apply Lean Canvas and Business Model Canvas to develop sustainable business models
viii. Assess the business models using the RESTART framework.
ix. Examine the economic viability of business opportunities and projects to support informed investment decision
x. Develop and present business plans and demonstrating research and presentation skills.

Learning and Teaching approaches

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures/ Guest Lecture	2	30
	Class discussions and exercises	1	15
	Presentations, and case studies	1	15
Independent Study	Written Assignments	1	60
	VLE Discussion and Research	1	
	Project Work	2	
Total		**8**	**120**

Assessment Approach

A. Group Work: Business Opportunity identification and selection (20%)

Students in groups of 3-5 members will work on the Business Opportunity Identification Process, where the group will identify, generate, evaluate and select viable business opportunities using structures, tools and techniques. Each group must submit a report within the word limit of 2000 - 3000 words, excluding annexures and references. Further, to promote fairness and accountability in group work, the module will implement several mechanisms to track individual contributions. These include reflection reports, peer evaluations, individual components, and progress checkpoints. These tools will help tutors accurately assess each student's level of engagement, effort, and learning. Additionally, these mechanisms ensure that all students actively participate and meet the intended learning outcomes. This approach not only encourages individual responsibility and discourages free-riding but also fosters the development of essential teamwork and personal skills that are crucial for academic and professional success. This assignment will be of 20 marks and will be assessed based on the following criteria:

5 marks Environmental Scanning (Critical problems identified, use of credible data, analytical insight from PESTLE Analysis)

3 marks	Idea generation (Creativity and variety in the idea generated, relevancy of the ideas to the identified problems and opportunities, application of appropriate ideation tools and techniques)
5 marks	Idea Evaluation (Use of appropriate tools and techniques, integration of creditable data to support the evaluation process, logical reasoning and analysis to assess feasibility)
4 marks	Idea selection (Detailed analysis using tools and techniques for idea selection, strong justification backed by data, and alignment of selected ideas with competencies, passion and resources)
3 marks	Referencing (Use of current and relevant data and references. In-text and end-text citations are as per the latest APA edition)

B. Initial Business Idea Pitching Using LEAN Canvas Framework (5%)

Based on the Assessment A. Students will conceptualize their business idea using the LEAN Canvas framework to achieve problem-solution fit and will pitch their conceptualized idea to the class. This assignment will be of 5 marks and will be assessed based on the following criteria:

1 mark	Problem and Customer Segments (Clearly identifies a specific problem and defines the target customer)
1 mark	Unique Value Proposition and Solution (Presents a compelling value proposition and a clear solution that addresses the problem)
1 mark	Revenue Stream and Cost Structure (Explains how the business will earn money and the expected major costs.)
1 mark	Market Channels and Key Metrics (Identifies how the product/service will reach customers and how success will be measured)
1 mark	Clarity, Presentation and Teamwork (Well-organised, confident delivery. Individual contribution will be assessed through peer evaluation and tutor observation)

Group Work: Value Proposition Statement (5%)

Student Teams will analyse their target customers and understand their needs, challenges, and aspirations. Students will create an Empathy Map, develop a Customer Profile, and draft a Value Map to create a Value Proposition Statement, using the template provided by the tutor. This assignment will be of 5 marks and will be assessed based on the following criteria:

2 marks	Empathy Map (A clear and specific understanding of what customers say, think, do, and feel, identification of key challenges or pain points, consistency with the target customer profile)
1 mark	Customer Profile (Clarity in defining customer jobs, identification of customer pains, detail in defining customer gains)
1 mark	Value Map (Alignment with a customer profile, creativity, and relevance of proposed products and services)
1 mark	Value Proposition Statement (Clear, concise, and easy-to-understand statement that avoids jargon and confusion.)

C. Group Work: Mapping Business Model Canvas (BMC) and Sustainable Business Modelling to validate ideas and create business strategy (10%)

The Project Teams will refine their customer insights into a clear and actionable business model. Using the Business Model Canvas and sustainability frameworks. Students will map their business's core elements, validate assumptions, and align their strategy to create, deliver, and capture value effectively by bridging the gap between the customer's needs and the business's capabilities, incorporating any sustainability frameworks. The students will make a group presentation of their Business Model Canvas to the class and get voted for the most sustainable business model. The presentation will comprise 10 marks and will be assessed based on the following criteria:

The presentation will be assessed based on the following criteria:

3 marks	Business Model Design and Content (The team presents a clear, coherent, and complete BMC addressing all components. Special emphasis is placed on problem-solution fit, value proposition, and practical feasibility.)
3 marks	Sustainability Integration (The model reflects social, environmental, and economic sustainability. The effort is made to align with Bhutanese values and national priorities (e.g. GNH domains, cultural preservation, environmental care)
2 marks	Team Pitching and Visual Presentation (The group demonstrates strong teamwork, coordination, and professionalism. The pitch is engaging, well-structured, and visually supported by clear materials.)
2 marks	Individual Contribution and Delivery (Each member is assessed on clarity, confidence, subject understanding, and contribution to both presentation and Qand A.)

D. Project (50%)

This project comprises two components - a business plan and the presentation of the business plan.

1. ***Business Plan Development (40%)***

For the business plan development, students will work in groups of 3-5 members to create a comprehensive business plan including all necessary components. The report should be within a limit of 6000 to 8000 words and will be assessed out of 40 marks based on the following criteria:

4 marks	Executive summary (Summary of business idea, key goals and objectives, feasibility and relevance)
6 marks	Project description (Mission and vision statement, unique value
8 marks	Marketing plan (Target market analysis, competitive analysis, marketing strategies and channels)
6 marks	Operations plan (Workflow and operational processes required)
5 marks	Organisational and management plan (Organisational structure and hierarchy, Roles and Responsibility, leadership and team dynamics)

8 marks financial plan (Financial statements, key financial ratios, break-even analysis, startup costs and funding plan)
3 marks Reference (Use of current and relevant data and references. In-text and end-text citations are as per the latest APA edition)

2. ***Presentation (10%)***

For the presentation, each group will be given 30 minutes and every member will have to deliver a presentation on the proposed business plan. Students will be assessed individually based on the following criteria:

2 marks Clarity and organisation (Logical structure and flow of the presentation)
3 marks Depth of knowledge (Thorough understanding of the topic, relevance, and alignment with the overall business plan)
3 marks Delivery and Communication Skills (Confidence, professional tone and
2 marks Question and Answer handling (Ability to address questions effectively and demonstrate their understanding of their business plan)

E. Class Test (10%)

The class test will cover the content from Unit I and Unit IX and will be conducted in the Virtual Learning Environment (VLE). It will consist of a combination of multiple-choice questions (MCQs) and short-answer questions, designed to assess both students' foundational knowledge and their ability to apply key concepts. The test will last 30 minutes. The class test will be 20 marks, which will be scaled down to 10%. Students must answer all questions within the allotted time. This assessment aims to evaluate both the breadth and depth of students' understanding of the subject matter, providing a balanced evaluation of recall, application, and analysis.

Overview of the assessment approaches and weighting

Areas of assignments	Quantity	Weighting (%)
A. Business Opportunity Identification Process (BOI)	1	20
B. Value Proposition Statement	1	10
C. Lean Canvas and Business Model Canvas	1	10
D.	1	40
a. Project: Business Plan Development b. Business Plan Presentation	1	10
E. Test on VLE	1	10
Total		**100**

Pre-requisites: None

Subject Matter:

Unit I: Introduction to Entrepreneurship

1.1 Evolution and Concept of Entrepreneurship
1.2 Entrepreneurship and Intrapreneurship

1.2.1 Entrepreneurship and Intrapreneurship as a source of Social, Sustainable, and economic change
1.2.2 Entrepreneurship and intrapreneurship as Engines of Development
1.3 Types of entrepreneurs
1.4 Technological Entrepreneurship
1.4.1 Definition of technology
1.4.2 Technology component
1.4.3 Technology entrepreneurship process
1.4.4 Technology Entrepreneurship success and resilience model
1.5 Characteristics of an entrepreneur
1.6 Entrepreneurial functions
1.7 Entrepreneurial methods
1.8 Bhutanese entrepreneurship ecosystem - Role and scope of entrepreneurship in Bhutan
1.9 Emerging trends and entrepreneurial scope

Unit II: Entrepreneurial Mind-Set and Competencies

2.1 Entrepreneurial Thinking
2.2 Grassroot Innovations
2.3 Problem-solving techniques
2.4 Entrepreneurial motivation
2.5 Evaluation of entrepreneurial competencies using GETT/SRQ
2.6 Emotional resilience and entrepreneurial discipline

Unit III: Business Opportunity Identification (BOI) and Selection

3.1 Meaning of business opportunity
3.2 BOI process
3.2.1 Environmental scanning
3.2.2 Generating ideas
3.2.3 Identifying and evaluating ideas/opportunities
3.2.4 Selecting ideas/opportunities
3.3 Value chain analysis and business development

Unit IV: Business Model

4.1 Empathy Map Canvas
4.2 Value Proposition Canvas
4.3 Lean Canvas
4.4 Business Model Canvas
4.5 Sustainable business models
4.5.1 RESTART Sustainable Business Model Innovation
4.5.1.1 Redesign Rather than Standstill
4.5.1.2 Experimentation Rather than Turnover
4.5.1.3 Service-Logic than Product-Logic
4.5.1.4 The Circular Rather than the Linear Economy
4.5.1.5 Alliances Rather than Solo-runs
4.5.1.6 Result Rather than Indulgences
4.5.1.7 Three-Dimensionality Rather than One
4.6 Trends driving the development of new Business Model

4.6.1 Sustainability Problem
4.6.2 Digitalization and Technological Opportunity Space
4.6.3 Changing consumer preferences and Lifestyles
4.7 Sustainable and Ethical business consideration
4.8 Business Pitching Skills and Pitch Decks

Unit V: Business Plan

5.1 Overview
5.1.1 Need and importance of business plan
5.1.2 Audience of business plan
5.1.3 Components of a business plan
5.1.3.1 Executive Summary
5.1.3.2 Business Profile
5.1.3.3 Industry Overview
5.1.3.4 Business Model
5.1.3.5 Market Analysis
5.1.3.6 Marketing Plan
5.1.3.7 Operations Plan
5.1.3.8 Organisational and Management Plan
5.1.3.9 Financial Plan

Unit VI: Marketing Plan

6.1 Business environment
6.2 Market Research
6.2.1 Description of product or services
6.2.2 Target Market
6.2.3 Target customers
6.2.4 Demand analysis
6.2.5 Supply analysis
6.2.6 Gap Analysis
6.3 Marketing Strategies
6.3.1 Competitors marketing strategies
6.3.2 Proposed Business Marketing Strategies
6.3.3 Forecast of Sales
6.3.4 Non-current assets for Marketing
6.3.5 Total Marketing Expenses

Unit VII: Production and Operations Plan

7.1 Production process and operational Schedule
7.2 Property, Plant and Equipment Requirements and Cost
7.3 Resource Requirement and Cost
7.3.1 Direct Resource requirement
7.3.2 Direct Labour requirement
7.3.3 Other direct cost
7.4 Overheads cost
7.4.1 Indirect Material cost
7.4.2 Indirect labour
7.4.3 Other indirect cost
7.5 Total Overhead cost and unit cost

Unit VIII: Organisational and Management Plan

8.1 Forms of business
8.2 Organisational structure
8.2.1 Types
8.2.2 Description of key position and responsibility
8.2.3 Recruitment, selection and training strategies
8.2.4 Non-current assets requirement and cost
8.2.5 Cost of personnel for administration
8.2.6 Pre-operating activities and expenses
8.2.7 Organisational and Management Expenses

Unit IX: Financials

9.1 Estimated project cost
9.2 Financing the project
9.3 Capital cost
9.4 Depreciation schedule
9.5 Financing costs include the cost of equity, the cost of debt, the weighted average cost of capital, and the loan amortisation schedule.
9.6 Cost sheet
9.7 Projected cash flow statement
9.8 Projected income statement
9.9 Projected financial position
9.10 Financial Performance Metrics
9.10.1 Break Even Analysis
9.10.2 Investment appraisal ratio
9.10.2.1 Pay-back Period
9.10.2.2 Net Present Value (NPV)
9.10.2.3 Internal Rate of Return (IRR)
9.10.3 Efficiency Ratio
9.10.3.1 Asset Turnover Ratio
9.10.3.2 Inventory Turnover Ratio
9.10.3.3 Employee Efficiency ratio
9.10.4 Solvency ratio
9.10.4.1 Debt-to-equity ratio
9.10.4.2 Interest coverage ratio
9.10.5 Liquidity Ratio
9.10.5.1 Current Ratio
9.10.5.2 Quick Ratio
9.10.5.3 Cash ratio
9.10.6 Profitability ratio
9.10.6.1 Net profit margin
9.10.6.2 Return on Investment
9.10.6.3 Return on Equity
9.10.6.4 Break Even Point

Reading List

Essential Readings

Hisrish, R., Peters, M., & Shepherd, D. (2016). *Entrepreneurship* (10th ed.). McGraw Hill Education.

Hsieh, T. (2014). *Delivering Happiness: A path to profits, passion and purpose*. Grand Central Publishing

Carpenter, K. (2021). *Introduction to Entrepreneurship*. Kwantlen Polytechnic University. https://open.umn.edu/opentextbooks/textbooks/introduction-to-entrepreneurship

Larson, A. (2011). *Sustainability, Innovation, and Entrepreneurship*. Saylor Foundation. https://open.umn.edu/opentextbooks/textbooks/39

Additional Readings

Aparicio, S., Turro, A., & Noguera, M. (2020). *Entrepreneurship and Intrapreneurship in Social, Sustainable, and Economic Development: Opportunities and Challenges for Future Research.* Sustainability, 12(21), 8958. https://doi.org/10.3390/su12218958

Barringer, B. R., & Ireland, R. D. (2024). *Entrepreneurship: Successfully Launching New Ventures (7th ed.).* Pearson.

Chishti, S., Bartoletti, I., Leslie, A., & Millie, S. M. (Eds.). (2020). *The AI book: The artificial intelligence handbook for investors, entrepreneurs and FinTech visionaries*. FINTECH Circle Ltd.

https://doi.org/10.1002/9781119551966

Drucker, P.F. (2014). *Innovation and entrepreneurship.* Harper Business.

Johnson, K.D. (2013). *The entrepreneur mind: 100 essential beliefs, characteristics, and habits of elite entrepreneurs*. Johnson Media Inc.

Jorgensen, S., Jacob, L., & Pedersen, T. (n.d.). *RESTART sustainable business model innovation*. Palgrave Studies in Sustainable Business. https://library.oapen.org/bitstream/handle/20.500.12657/23128/1007028.pdf

Kumar, A., (2012). Entrepreneurship: *Creating and leading an entrepreneurial organization*. Pearson.

Kuratko, D. F. (2024). *Entrepreneurship: Theory, Process, Practice*. Cengage Learning Asia Pvt Limited.

Mohiuddin, M., Reza, M. N. H., Hosseini, E., & Ed-Dafali, S. (Eds.). (2023). *Entrepreneurship: New insights*. IntechOpen. https://doi.org/10.5772/intechopen.104246

Mycoskie, B. (2012). *Start Something That Matters.* Random House Inc.

Nambisan, S. (2017). Digital entrepreneurship: Toward a digital technology perspective of entrepreneurship. *Entrepreneurship Theory and Practice, 41*(6), 1029–1055. https://doi.org/10.1111/etap.12254

Nurul, M., Reza, H., Hosseini, E., & Dafali, S. (2023). *Entrepreneurship New Insights.* https://mts.intechopen.com/storage/books/12140/authors_book/authors_book.pdf

Osterwalder, A., & Pigneur, Y. (2010). *Business model generation: A handbook for visionaries, game changers and challengers*.

https://www.researchgate.net/publication/236650466_Business_Model_Generation_A_Handbook_for_Visionaries_Game_Changers_and_Challengers

Ries, E. (2017). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.*: Currency.

Royal Government of Bhutan. (2007). *Labour and Employment Act of Bhutan.* National Assembly of Bhutan.

Royal Government of Bhutan. (2019). *Cottage Small and Medium Industry policy.* Royal Government of Bhutan.

Royal Government of Bhutan. (2016). *Companies Act of Bhutan.* National Assembly of Bhutan.

Royal Government of Bhutan. (2024). *13th Five-Year Plan.* Royal Government of Bhutan.

Royal Government of Bhutan. (2017). *Economic Development Policy of Bhutan.* Ministry of Economic Affairs.

Royal Government of Bhutan. (n.d.). *BAS for SMEs.* Royal Government of Bhutan.

Royal Government of Bhutan. (n.d.). *Cottage Small and Medium Industry Development Strategies (2012-2020). Royal* Government of Bhutan

Scarborough, N.M. (2014). *Entrepreneurship and effective small business management* (11th ed.). Pearson.

Thiel, P., & Masters, B. (2014). *Zero to one: Notes on startups, or how to build the future*. Crown Business.

Zacharakis, A., Corbett, A. C., & Bygrave, W. D. (2020). *Entrepreneurship*. Wiley.

Updated: April 21, 2025

Year 4 Semester I

Module Code and Title:	**EMT403 Blockchain Technology and Applications**
Programme:	BSc in Applied Computing and Information Technology
Credit:	12
Module Tutor:	Phuntsho Wangdi

General Objective
This module introduces students to the fundamental concepts of blockchain technology and develops an understanding of how blockchain systems operate and are applied across various sectors. The module also provides an introductory overview of smart contracts, common blockchain platforms and real-world applications.

Learning Outcomes
On completion of the module, students will be able to:

i. explain the fundamental purpose and benefits of blockchain
ii. describe how blockchain protects and secures data
iii. differentiate between blockchain network types

iv. illustrate the lifecycle of a blockchain transaction
v. apply smart contracts to solve simple problems
vi. evaluate blockchain applications and their practical challenges
vii. analyse the ethical and social implications of blockchain adoption
viii. design a basic blockchain solution to address a given problem.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	2	60
	Computer Laboratory Session	2	
Independent Study	Assignment	2	60
	Self-study	2	
Total		**8**	**120**

Assessment Approach

A. Laboratory Portfolio (15%)
Students must submit a laboratory portfolio that includes several hands-on tasks completed during the semester. The portfolio should show a basic practical understanding of blockchain concepts through simple activities such as setting up a blockchain environment, creating and checking blocks and transactions, simulating transaction validation, exploring how consensus works, and, where applicable, demonstrating a simple smart contract in a safe learning environment. The submission should include screenshots or log outputs as evidence, along with brief explanations of each task, and all materials should be clearly organized. The following criteria will be followed for marking:

5 Marks Practical Implementation (Successful completion of hands-on tasks)

4 Marks Conceptual Understanding (Use of concept, Demonstration of the assigned task)

2 Marks Evidence and Documentation (Clear screenshots, log outputs, and relevant artefacts provided)

2 Marks Explanation and Reflection (Short, clear explanations of tasks, observations, and outcomes)

2 Marks Organisation (Well-organised portfolio with proper structure and clarity)

B. Case Study (20%)
Students in groups of 4–5 members will complete a written case study that analyzes a real-world blockchain application scenario. The case study should focus on understanding the given scenario, identifying key problems or requirements, and proposing a basic blockchain-based solution outline. Students are expected to explain how blockchain can be applied to the scenario, including the type of blockchain, key features used, and expected benefits. The submission should be well-structured, clearly written, and demonstrate a practical understanding of blockchain concepts. The case study will be marked based on following criteria:

5 Marks Understanding of Scenario (Clear understanding of the real-world blockchain application and valid justification)

5 Marks Problem Analysis and Requirements (Identification and analysis of key problems, needs, and requirements)

5 Marks Proposed Blockchain Solution (Appropriate blockchain type, basic solution outline, and feasibility)
3 Marks Application of Blockchain Concepts (Correct use of blockchain features, concepts, and terminology)
2 Marks Organization and Clarity (Logical structure, clarity of writing, and presentation quality)

C. Mid-Semester Examination (25%)
The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours

D. Semester-End Examination (40%)
The Semester-End Examination will be conducted for 100 marks over a duration of three hours.

Overview of assessment approaches and weighting

Areas of Assignments	Quantity	Weighting (%)
A. Laboratory Portfolio	1	15
B. Case Study	1	20
C. Mid-Semester Examination	1	25
D. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: NET201 Networking Fundamentals, CYS301 Introduction to Cyber Security

Subject Matter

Unit I: Introduction to Blockchain Technology

1.1. Definition and key concepts
1.2. Brief history and evolution
  1.2.1. Origins of blockchain: Bitcoin
  1.2.2. Evolution beyond Bitcoin: Broader applications
1.3. Features of blockchain
  1.3.1. Decentralisation
  1.3.2. Transparency
  1.3.3. Immutability
  1.3.4. Auditability
1.4. Types of blockchain
  1.4.1. Public
  1.4.2. Private
  1.4.3. Consortium
1.5. Components of a blockchain
  1.5.1. Blocks: Header and body overview
  1.5.2. Chains: Linking blocks
  1.5.3. Nodes: Role in the network
  1.5.4. Ledger: Distributed record keeping

Unit II: Blockchain Architecture

2.1 Overview of blockchain architecture
2.2 Layered architecture of blockchain
2.3 Data layer
  2.3.1 Network layer
  2.3.2 Consensus layer
  2.3.3 Application layer
2.4 Blockchain network architecture

- 2.4.1 Peer-to-Peer (P2P) network model
- 2.4.2 Node types: Full nodes, light nodes, miners/validators

2.5 Data propagation and communication
- 2.5.1 Transaction broadcasting
- 2.5.2 Block propagation

2.6 Security aspects of blockchain architecture
- 2.6.1 Cryptography overview
- 2.6.2 Trust and fault tolerance

Unit III: How Blockchain Works

3.1 Structure of a block
- 3.1.1 Conceptual understanding of block header and body

3.2 Hashing concepts for integrity

3.3 Transactions
- 3.3.1 Definition of a transaction
- 3.3.2 Transaction lifecycle

3.4 Consensus mechanisms
- 3.4.1 Purpose of consensus in a blockchain
- 3.4.2 Proof of Work and Proof of Stake

3.5 Adding blocks
- 3.5.1 Overview of block validation
- 3.5.2 Chain growth and forks
- 3.5.3 Concept of finality

Unit IV: Blockchain Platforms and Smart Contracts

4.1 Blockchain platforms overview
- 4.1.1 Bitcoin: Conceptual overview
- 4.1.2 Ethereum: Conceptual overview
- 4.1.3 Differences and use cases

4.2 Smart contracts
- 4.2.1 Definition and purpose
- 4.2.2 Role in the blockchain ecosystem
- 4.2.3 Simple conceptual use cases

4.3 Smart contract lifecycle
- 4.3.1 Deployment
- 4.3.2 Invocation
- 4.3.3 Observing outcomes

Unit V: Blockchain Applications

5.1 Cryptocurrency applications
- 5.1.1 Bitcoin and basic concepts

5.2 Non-currency applications
- 5.2.1 Supply chain management
- 5.2.2 Digital identity
- 5.2.3 Record keeping
- 5.2.4 Verification and audit trails

Unit VI: Benefits, Challenges, Limitations, and Future Trends

6.1 Benefits
- 6.1.1 Transparency and data integrity
- 6.1.2 Auditability, tamper resistance, and shared trust

6.2 Challenges
- 6.2.1 Scalability, performance issues, and energy consumption
- 6.2.2 Governance and interoperability

6.3 Regulatory and adoption considerations

6.3.1 Legal and regulatory overview
6.3.2 Adoption barriers
6.4 Myths and misconceptions
6.5 Ethical and social implications
6.5.1 Privacy considerations
6.5.2 Responsible use
6.5.3 Governance and accountability
6.6 Emerging trends
6.6.1 NFTs
6.6.2 Decentralized Finance (DeFi)
6.6.3 Central Bank Digital Currencies
6.6.4 Blockchain in government and education
6.6.5 Blockchain combined with IoT
6.7 Career opportunities and skill pathways

List of Practicals

i. Setting up a simple blockchain learning environment (local/simulated).
ii. Creating blocks and simulating transactions in a controlled environment.
iii. Inspecting blocks and transaction history (block explorer concepts where applicable).
iv. Demonstrating transaction validation behaviour (guided exercise).
v. Introductory smart contract demonstration (deploy and observe a simple contract behaviour in a safe learning setup).
vi. Short reflection notes on application suitability and limitations for selected scenarios.

Reading List

Essential Reading

Antonopoulos, A. M. (2017). *Mastering Bitcoin: Unlocking digital cryptocurrencies.* O'Reilly Media.

Davis, E. (2021). *Coding for Beginners: Blockchain Development: A Step-By-Step Guide to Create Your Own Blockchains, Cryptocurrencies and NFTs*. Code Connections.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies.* Princeton University Press.

Tapscott, D., & Tapscott, A. (2016). *Blockchain revolution: How the technology behind Bitcoin is changing money, business, and the world*. Penguin.

Singhal, B., Dhameja, G., & Panda, P.S. (2018). *Beginning Blockchain. A Beginner's Guide to Building Blockchain Solutions*. Apress.

Additional Reading

Singhal, B., Dhameja, G., & Panda, P.S. (2018). *Beginning Blockchain. A Beginner's Guide to Building Blockchain Solutions*. Apress

IBM Blockchain. (n.d.). Blockchain for beginners. Retrieved from https://www.ibm.com/blockchain

Date: May 2026

Module Code and Title: **BIS402 Enterprise Systems and Business Context**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Sonam

General Objective
This module aims to introduce enterprise information systems and their role in supporting organisational operations and business processes. It will cover the integration of core functions through systems such as ERP, CRM and SCM while developing understanding of business process modelling, governance and service management principles.

Learning Outcomes
On completion of the module, students will be able to:

i. describe the business functions and their information needs
ii. explain the architecture and core modules of enterprise systems support organisational processes
iii. analyse and model business processes, highlighting opportunities for integration and improvement
iv. evaluate enterprise system solutions against business requirements
v. design high-level enterprise system architectures including master data and integration approaches
vi. apply IT governance and service management frameworks to ensure alignment and control with organisational objectives
vii. assess compliance, privacy, and risk considerations for enterprise systems
viii. present findings and recommendations clearly through professional artefacts.

Learning and Teaching Approach:

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures	3	60
	Laboratory Sessions	1	
Independent study	Assignments, Project work	2	60
	Reading and review of class materials	2	
Total		**8**	**120**

Assessment Approach:

A. Case Study Analysis (15%)
Students in groups of 4-5 will complete a written case study analysis with 1200-1500 words focusing on evaluating an organisational challenge and proposing system-based solutions. The analysis should include process mapping and structured recommendations. The assessment will be graded based on the following marking criteria:

5 Marks Understanding the requirements (Demonstrate clear grasp of organisational needs and solution options)
5 Marks Process analysis (Effective use of process modelling to identify improvements)
3 Marks Risk and compliance consideration (Addresses potential risks and governance issues)
2 Marks Clarity of recommendation (Provide actionable, well-structured conclusions)

B. Assignment – Process Integration Analysis (15%)
Individually students will complete a written assignment professionally within 800-1000 words covering the contents from Unit I-III, showing how organisations use information and enterprise systems to improve their processes. Students have to explain the main functions in an organisation, the kind of information system required for each function to work together using simple process models making the organisation more efficient and integrated. The assessment will be graded based on the criteria;

4 Marks Understanding the Business functions (Explaining clearly the roles and information need of organisational functions)

3 Marks Enterprise systems integration (Describing how enterprise systems support integration and efficiency across functions)
2 Marks Process modelling and improvement (Provide clear processes of “as-is” and “to-be”)
4 Marks Critical reflection (Critical reflection identifying inefficiencies, risks, or opportunities for improvement in processes and systems)
2 Marks Structure and professionalism (Present ideas in a logical, well-formatted and clarity of language and use of diagrams where appropriate)

C. Project – Digital Transformation Roadmap (20%)

In group of 4-5 students will develop a Digital Transformation Roadmap and write as a report within 1800-2000 words -10% and roadmap deck (presentation) – 10%. The group project is designed to help students apply their knowledge of enterprise systems and business process management to a practical organisational scenario. It encourages teamwork, strategic thinking, and professional communication. The aim is for students to demonstrate how digital transformation can be planned, structured, and communicated effectively. The assessment will be assessed based on the criteria;

i. **Roadmap Report (10%)**

This will be out of 20 mark and converted to 10%.

6 Marks Strategic alignment and scope (Roadmap align with organisational goals and defines scope, phases, and dependencies clearly)
5 Marks Architecture and integration (Provide an appropriate architecture showing system, data and process integration)
4 Marks Benefits and risk management (Identifies measurable benefits, ROI, and risks with appropriate mitigation strategies)
3 Marks Change Plan (Outlines realistic implementation steps showing detailed change plans)
2 Marks Structure and Quality (Well organised, visually clear and professionally formatted)

ii. **Presentation (10%)** – (10-12 slides) for 15 mins. This will be out of 20 mark and converted to 10%.

7 Marks Clarity of Content (Key ideas are communicated clearly and logically)
5 Marks Engagement and Delivery (presentation style is confident, engaging and appropriate)
3 Marks Visual Support (Slides/visual are clear, professional and support the spoken content effectively)
3 Marks Team Collaboration (Participation showing teamwork and coordination)
3 Marks Stakeholder Communication (Presentation demonstrates awareness of need and addresses them persuasively)

D. Mid-Semester Examination (20%)

The Mid-semester examination will be conducted for 50 marks over a duration of one and half hours. It will be converted to 20%

E. Semester-End Examination (30%)

The semester-end examination will be conducted for 100 marks over a duration of three hours. It will be converted out of 30%.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting
A. Case Study Analysis	1	15
B. Assignment	1	15

C. Group Project	1	20
D. Mid-Semester Examination	1	15
E. Semester-End Examination (SEE)	1	35
Total		**100**

Pre-requisites: SDP201 System Analysis and Design

Subject Matter

Unit I: Understanding Business Functions

1.1 Overview of organisational structure and roles
1.2 Core functions:
  1.2.1 Management
  1.2.2 Administration
  1.2.3 Finance
  1.2.4 HR
  1.2.5 Operations
  1.2.6 Marketing
1.3 Information needs and typical workflows for each function
1.4 How IS supports decision-making and efficiency

Unit II: Enterprise Systems Overview

2.1 Introduction to Enterprise Systems
  2.1.1 Definition
  2.1.2 Purpose of ERP, CRM, and SCM systems
  2.1.3 Role of enterprise systems in business integration and process automation
2.2 ERP Architecture and Core Modules
  2.2.1 Finance, HR, Procurement, Inventory, Manufacturing modules
  2.2.2 Data flows and master data concepts
2.3 CRM and SCM Systems
  2.3.1 CRM functions
    2.3.1.1 customer data
    2.3.1.2 sales
    2.3.1.3 marketing automation
  2.3.2 SCM functions
    2.3.2.1 procurement
    2.3.2.2 logistics
    2.3.2.3 supplier collaboration
2.4 Deployment Models
  2.4.1 On-premise vs. cloud/SaaS solutions
  2.4.2 Hybrid architectures and scalability considerations

Unit III: Business Process Management

3.1 Principles of BPM
  3.1.1 Process orientation and continuous improvement
  3.1.2 Relationship between BPM and enterprise systems
3.2 Process Modelling Techniques
  3.2.1 BPMN notation
    3.2.1.1 events, tasks
    3.2.1.2 gateways
    3.2.1.3 swimlanes
  3.2.2 SIPOC diagrams and workflow mapping
3.3 Process Analysis
  3.1.1 Identifying bottlenecks and inefficiencies;
  3.1.2 Linking “as-is” and “to-be” models to ERP/CRM functionality
3.4 Performance Metrics
  3.1.3 Cycle time

3.1.4 throughput
3.1.5 error rates
3.1.6 KPI design for process improvement

Unit IV: Enterprise Architecture and Integration

4.1 Architecture Layers
4.1.1 Application
4.1.2 Data
4.1.3 Integration layers in enterprise systems
4.2 Master Data Management
4.2.1 Concepts of master data and data governance
4.3 Integration Approaches
4.3.1 APIs
4.3.2 ETL processes
4.3.3 Middleware
4.3.4 message queues
4.4 Reporting and Analytics
4.4.1 Operational vs. strategic dashboards;
4.4.2 Data aggregation for decision-making

Unit V: Governance and Service Management

5.1 IT Governance Frameworks
5.1.1 Overview of COBIT principles and domains;
5.1.2 Decision rights and accountability structures
5.2 Service Management Practices
5.2.1 ITIL concepts:
5.2.1.1 incident, change
5.2.1.2 release management
5.2.2 Service level agreements (SLAs) and performance indicators
5.3 Compliance and Audit
5.3.1 Internal controls and audit trails for enterprise systems

Unit VI: Security, Privacy, and Risk

6.1 Access Control
6.1.1 Role-based access and segregation of duties
6.2 Data Privacy - Regulatory requirements (e.g., GDPR overview)
6.3 Risk Management
6.3.1 Identifying risks in enterprise system implementation
6.3.2 Mitigation strategies and contingency planning

Reading List

Essential Reading

Harmon, P. (2019). *Business process change: A business process management guide for managers and process professionals* (4th ed.). Morgan Kaufmann.

ISACA. (2019). COBIT 2019 framework: Introduction and methodology. https://www.isaca.org/resources/cobit

Open Textbook Library. (n.d.). Information systems for business and beyond. https://open.umn.edu/opentextbooks/textbooks/information-systems-for-business-and-beyond

The Open Group. (2018). TOGAF® standard, version 9.2. https://pubs.opengroup.org/architecture/togaf9-doc/arch/

Turban, E., Pollard, C., & Wood, G. (2018). Information technology for management: On-demand strategies for performance, growth and sustainability (11th ed.). Wiley.

Additional Reading

Object Management Group. (2014). *Business process model and notation (BPMN) version 2.0*. https://www.omg.org/spec/BPMN/2.0/

Danby, S. (2025, January). *ITIL 4 management practices explained: Full list and purposes*. ITSM.tools. https://itsm.tools/34-itil-4-management-practices/

Selected industry case studies (ERP/CRM/SCM implementation, benefits realisation, process mining examples). [Unpublished manuscript].

Date: May 2026

Module Code and Title: **HCI403 IT Project Management**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Sonam

General Objective

This module develops students' ability to manage IT projects effectively by applying structured methodologies, tools, and professional practices. It emphasises planning, execution, monitoring, and closure of projects, enabling students to balance scope, time, cost, quality, and risk while ensuring ethical and collaborative delivery.

Learning Outcomes

On completion of the module, students will be able to:

i. analyse the core project management concepts and their relevance to managing IT projects
ii. apply suitable project management methodologies to IT project contexts
iii. develop project plans covering scope, schedule, cost, resources, and quality
iv. use project management tools to track progress and communicate status
v. assess project risks while managing project changes
vi. manage stakeholder communication and team collaboration effectively
vii. apply ethical and professional practices in IT project delivery
viii. develop project closure and lessons-learned artefacts that demonstrate reflective practice and readiness for larger projects.

Learning and Teaching Approach

Type	Approach	Hours per week	Total Credit Hours
Contact	Interactive Lectures and Laboratory Sessions	4	60
Independent Study	Assignment; Reading; Practice exercises, PM artefacts development	4	60
Total		**8**	**120**

Assessment Approach

A. Class Test (10%)

Students will take a written class test during regular class hours for a duration of 35 minutes, carrying a maximum of 25 marks and converted into 10%. The test will cover topics from Units I and II, focusing on project management principles, methodologies, and terminology.

B. Assignment – Planning Lab (10%)

Student in groups of 3 – 4 will complete one written assignment on topics related with the application of a project management ensuring students will be creating project plans, risk

registers, and stakeholder communication strategies covering from Units II–IV. It will be assessed based on following criteria:

3 Marks Concepts (Accurate and consistent use of PM terms, frameworks and approaches)

3 Marks Feasibility and realism (Plans are practical and achievable, Risk register includes relevant risks with mitigation strategies and appropriate communication strategies)

2 Marks Connection with the content (linked frameworks, lifecycle artefacts, and HRM concepts and clear alignment between theoretical concepts and practical outputs)

2 Marks Teamwork and output (All group members contribute, Work is coherent, consistent, and professionally presented or no inconsistencies in sections)

C. Project Management Portfolio (40%)

In a group of 4-5 students will submit a consolidated Project Management Portfolio that demonstrates their ability to plan, monitor, and close an IT project using professional project management practices. The assessment is divided into two parts: The Project Management Portfolio report writing within 2800 – 3000 words worth 25%, and they should produce a professional project management artefact. The second part of the assessment is a presentation worth 15%. The students will prepare a presentation within 12-15 slides and there will be a Q&A session to assess the ability to defend the project decisions. The assessment will be graded based on following criteria:

i. **Project Management Portfolio Report (25%)**

5 Marks Scope and charter quality (clarity of objectives, deliverables, constraints, success criteria)
5 Marks Schedule and resourcing quality (WBS and schedule logic; realistic sequencing and milestones)
5 Marks Cost/effort reasoning (appropriate level estimate; assumptions clear; feasibility)
4 Marks Risk management quality (risk identification, mitigation, monitoring; active use during project)
3 Marks Stakeholder and communication management (stakeholder map, communication plan, conflict handling evidence)
3 Marks Monitoring and change control (status tracking; change log; justified adjustments; tool evidence)

ii. **Presentation (15%)**

4 Marks Clarity of Content (Communicates key artefacts and decisions logically and effectively)
3 Marks Engagement and Delivery (Confident, professional delivery with team participation).
2 Marks Visual and Tool Evidence (Clear, professional slides and artefact/tool evidence supporting the narrative).
4 Marks Critical Reflection (Insightful discussion of risks, changes, and considerations).
2 Marks Response (Able to respond accurately and thoughtfully to questions, demonstrating ownership of work).

D. Semester-End Examination (40%)

The semester-end examination is a comprehensive written assessment conducted for three hours for a total of 100 marks and the marks obtained will be converted into 40%.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)

A. Class Test	1	10
B. Assignment	1	10
C. Project Management Portfolio	1	40
D. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: SDP201 Systems Analysis and Design

Subject Matter

Unit I: Fundamental of Project Management and IT Project Management

1.1. Fundamentals of project management
  1.1.1. Definition
  1.1.2. Scope, and
  1.1.3. Importance
1.2. Distinction between projects and operations
1.3. Key concepts
  1.3.1. Project scope
  1.3.2. Quality
  1.3.3. Schedule
  1.3.4. Cost, and
  1.3.5. Risk
1.4. Program management and portfolio management – relationship to projects
1.5. Roles and responsibilities of project managers
1.6. Specific roles in IT project management
  1.6.1. Technical coordination
  1.6.2. Stakeholder engagement
  1.6.3. Governance and ethical practice
1.7. Challenges unique to IT projects
  1.7.1. Scope creep
  1.7.2. Technology changes
  1.7.3. Integration issues
  1.7.4. Data privacy concerns

Unit II: Project Management Frameworks and Approaches

2.1. Overview of project management frameworks
  2.1.1. PMBOK knowledge areas
  2.1.2. PRINCE2 overview
2.2. Project management approaches
  2.2.1. Waterfall
  2.2.2. Agile
  2.2.3. Scrum
  2.2.4. Kanban
  2.2.5. Lean
  2.2.6. Hybrid models
2.3. Specifications of each approach
  2.3.1. Scope definition
  2.3.2. Scheduling
  2.3.3. Quality assurance, and
  2.3.4. Validation methods
2.4. Criteria for selecting appropriate approaches in IT project contexts

Unit III: Project Lifecycle and Artefacts in IT Project Management

3.1 Phases of the project lifecycle
  3.1.1 Initiation
  3.1.2 Planning

- 3.1.3 Execution
- 3.1.4 Monitoring
- 3.1.5 Closure

3.2 Key artefacts aligned to IT project management

- 3.2.1 Project charter and scope statement
- 3.2.2 Work Breakdown Structure (WBS) and Gantt chart baseline
- 3.2.3 Effort and cost estimation
- 3.2.4 Quality planning and acceptance criteria
- 3.2.5 Risk register and monitoring plan
- 3.2.6 Communication plan and stakeholder map
- 3.2.7 Change control log
- 3.2.8 Closure checklist and lessons learned report

3.3 Project design, scoping, and validation as critical lifecycle activities

Unit IV: Human Resource Management in IT Projects

4.1 Resource allocation and team roles
4.2 Skills mapping and competency planning
4.3 Team collaboration strategies
4.4 Conflict resolution and negotiation approaches
4.5 Leadership styles in IT project management
4.6 Ethical considerations in HRM (fairness, inclusivity, professional conduct)

Unit V: Project Execution and Monitoring

5.1 Execution strategies and monitoring mechanisms
5.2 Tracking progress using tools (MS Project, Excel, Jira, Trello)
5.3 Performance metrics

- 5.3.1 Schedule variance
- 5.3.2 Cost variance
- 5.3.3 Earned value analysis

5.4 Change management and governance controls
5.5 Monitoring artefacts

- 5.5.1 Status reports
- 5.5.2 Dashboards
- 5.5.3 Change logs

5.6 Ethical decision-making in monitoring and change control

Unit VI: Risk Management

6.1 Identifying risks in IT projects
6.2 Qualitative and quantitative risk analysis
6.3 Risk mitigation strategies and monitoring
6.4 Risk registers and escalation procedures
6.5 Ethical decision-making in risk management

- 6.5.1 Responsible trade-offs,
- 6.5.2 Data privacy implications

Unit VII: Stakeholder and Communication Management

7.1 Stakeholder identification and analysis
7.2 Communication planning and tools
7.3 Managing stakeholder expectations and conflicts
7.4 Negotiation and collaboration strategies
7.5 Ethical decision-making in stakeholder communication

Unit VIII: Project Closure and Evaluation

8.1 Deliverable acceptance and handover

8.2 Post-project review and lessons learned/retrospectives
8.3 Knowledge management and continuous improvement
8.4 Validation of project outcomes against scope, quality, and success criteria
8.5 Ethical reflection on project outcomes and professional responsibility

Reading List

Essential Reading

Kerzner, H. (2017). *Project management: A systems approach to planning, scheduling, and controlling* (12th ed.). Wiley.

Schwalbe, K. (2018). *Information technology project management* (9th ed.). Cengage Learning.

Additional Reading

Berkun, S. (2008). *Making things happen: Mastering project management*. O'Reilly Media.

Cohn, M. (2005). *Agile estimating and planning*. Prentice Hall.

Institute of Electrical and Electronics Engineers. (2011). IEEE standard for project management (IEEE Std 1490-2011). IEEE.

Project Management Institute. (2021). *A guide to the project management body of knowledge (PMBOK® guide)* (7th ed.). Project Management Institute.

Date: May 2026

Module Code and Title: **HCI404 IT Laws and AI Ethics**
Programme: BSc in Applied Computing and Information Technology
Credit: 12
Module Tutor: Tashi Wangmo

General Objective

This module aims to develop students' understanding of the legal, ethical and governance aspects of modern computing. Through applied ad reflective learning, students will evaluate emerging technologies, apply professional responsibility and contribute to ethical and sustainable digital practices.

Learning Outcomes

On completion of the module, students will be able to:

i. analyse national and international IT laws, including data privacy, cybercrime, intellectual property, and digital evidence
ii. demonstrate professional accountability by engaging with established codes of conduct (ACM, IEEE, Software Engineering Code)
iii. apply codes of conduct to analyse and resolve real-world ethical dilemmas in computing contexts
iv. critically assess data protection and surveillance practices, balancing privacy rights with governance and security imperatives
v. use structured ethical decision-making models to evaluate and resolve professional dilemmas beyond codes of conduct
vi. evaluate the ethical implications of emerging technologies, particularly Artificial Intelligence, with respect to bias, fairness, transparency, and accountability
vii. apply intellectual property principles to ensure ethical and legal use of digital assets

viii. interpret GNH alignment in AI governance, assessing impacts on employment, equity, and wellbeing
ix. interpret Bhutanese IT legislation and official cybersecurity policy documents, situating them within global frameworks such as GDPR and cybercrime conventions
x. synthesise knowledge of IT law, ethical AI, and governance principles to prepare for civic responsibility and global citizenship in technology leadership.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Interactive Lectures, case law discussion, ethical dilemma workshops, and guest speakers	4	60
Independent study	Reading legal statutes and ethical guidelines, case studies, drafting compliance reports, and self-reflection	3	60
	Reading, reviewing materials, analysing case studies	1	
Total			**120**

Assessment Approach

A. VLE Quiz (20%)
Quizzes will account for 20% and each quiz carries 10 marks and must be completed within 30 minutes. They will be conducted both through the Virtual Learning Environment (VLE) and in offline classroom sessions, so students should be prepared for either format.

The quizzes will cover material from Units I–II and V, focusing on fundamental concepts such as legal definitions, ethical theories, and components of AI bias.

B. Comparative Analysis of Data Protection Frameworks (20%)

Students will work in groups of 4 – 5 members, engaging with material from Units II–IV and VII.

Each group is required to complete two tasks: first, a case study analysis report where they examine a national or international example such as a data breach, copyright infringement, or privacy violation, and provide legal and ethical reasoning; second, an AI ethics critique in which they evaluate the ethical implications of an AI system, focusing on bias, transparency, accountability, and alignment with GNH values. This will be evaluated based on the following criteria:

The total marks from the two tasks will be converted to 20%.

Task 1: Case Study Analysis Report (20 Marks)

6 Marks	Legal and Ethical Analysis (Accurate interpretation and application of relevant IT laws, policies, and ethical principles; depth of analysis; critical evaluation of the selected case.)
5 Marks	Recommendations and Practical Application (Feasible, realistic, and well-justified recommendations that address the identified legal and ethical issues.)
4 Marks	Use of Evidence and Argumentation (Appropriate use of case evidence, logical development of arguments, integration of relevant theories or legal frameworks, and critical reasoning.)

3 Marks	Organisation and Language (Logical organisation, coherent presentation, clear academic language, and.)
2 Marks	Referencing (Accurate application of APA referencing)

Task 2: AI Ethics Critique (20 Marks)

6 Marks	Ethical Evaluation of the AI System (Critical evaluation of bias, transparency, accountability, privacy, fairness, and other ethical issues using appropriate ethical frameworks.)
4 Marks	Application to GNH and Societal Context (Meaningful analysis of the AI system's implications for individuals, society, and alignment with Gross National Happiness values.)
5 Marks	Recommendations and Critical Reflection (Practical and well-justified recommendations for improving the ethical use and governance of the AI system, supported by critical reflection.)
3 Marks	Organisation and Language (Logical organisation, coherent presentation, clear academic language)
2 Marks	Referencing (accurate application of APA referencing)

C. Seminar / Debate (20%)

The seminar/debate component will account for 20% of the overall grade.
Students will participate in structured group seminars and debates, engaging with material from Units V–VIII, which cover cybercrime, intellectual property, policy drafting, and professional responsibility.

Each group of 4-5 students will select a national or international ethical or legal issue, such as cybercrime prosecution, intellectual property disputes, AI governance, or privacy versus surveillance.

They will present policy recommendations or ethical positions that integrate GNH governance values, while also critically evaluating opposing viewpoints. Students are expected to defend their positions using theoretical frameworks and draw on relevant case studies, demonstrating both depth of research and the ability to engage in constructive, evidence-based debate. The marking criteria will be as follows:

Option A: Seminar (20%)

7 Marks	Depth of Research and Analysis (Comprehensive understanding of the selected issue through the application of relevant IT laws, policies, ethical frameworks, and national and international case studies.)
5 Marks	Critical Evaluation and Recommendations (Critical analysis of legal and ethical issues; well-reasoned and feasible recommendations supported by evidence.)
5 Marks	Presentation and Organisation (Logical organisation, coherent presentation, effective use of visual aids, appropriate time management, and audience engagement.)

3 Marks	Language (Clear, fluent, and professional oral communication using appropriate academic and technical terminology with accurate grammar and pronunciation.)

Option B: Debate (20%)

7 Marks	Argument Development and Evidence (Develops logical, persuasive, and well-supported arguments using relevant legal, ethical, and theoretical frameworks together with appropriate evidence.)
5 Marks	Rebuttal and Critical Thinking (Effectively responds to opposing arguments, evaluates alternative perspectives, and defends positions using reasoned analysis.)
5 Marks	Teamwork and Debate Conduct (Balanced participation, effective coordination, respectful engagement, adherence to debate procedures, and professional interaction.)
3 Marks	Language (Clear, confident, and persuasive oral communication using appropriate technical vocabulary, accurate grammar, and effective delivery.)

D. Semester-End Examination (40%)

The semester-end examination is a comprehensive written assessment conducted for three hours for a total of 100 marks. It will be converted to 40%

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Quizzes / Short Tests	2	20
B. Group work	2	20
C. Seminar and Debate	1	20
D. Semester-End Examination (SEE)	1	40
Total		**100**

Pre-requisites: HCI101 Mindful Technology and Digital Wellbeing

Subject Matter

Unit I: Evolution of Technology, Ethics, and IT Law

- 1.1 Evolution of technology and ethical concerns
- 1.2 International and national IT laws
  - 1.2.1 Jurisdiction in cyberspace and international dimensions
  - 1.2.2 Overview of major global IT law frameworks: GDPR, cybercrime conventions
  - 1.2.3 Bhutanese IT law and regulatory context
- 1.3 Professional Codes of Conduct
  - 1.3.1 ACM Code
  - 1.3.2 IEEE Code
  - 1.3.3 IEEE/ACM Software Engineering Code
- 1.4 Ethical Theories applied to computing
  - 1.4.1 Deontology
  - 1.4.2 Consequentialism
  - 1.4.3 Virtue ethics

Unit II: Data Protection, Privacy, and Surveillance

- 2.1 Privacy concepts: personal data, Personally Identifiable Information (PII)
- 2.2 Data protection laws: global and Bhutanese context
- 2.3 Data breaches: notification and liability

2.4 Digital surveillance and ethical implications

Unit III: Ethical Frameworks and Professional Dilemmas

3.1 Decision-making models for ethical conflicts
3.2 Case studies: confidentiality, whistleblowing, conflicts of interest, dilemmas
3.3 Accountability and responsibility in systems development

Unit IV: Artificial Intelligence Ethics and Governance

4.1 AI bias and fairness: sources, mitigation
4.2 Transparency and explainability (XAI)
4.3 AI accountability: liability for autonomous systems
4.4 GNH alignment: AI's impact on employment and equity

Unit V: Cybercrime, Digital Evidence, and Security

5.1 Types of cybercrime
5.1.1 Hacking
5.1.2 Malware
5.1.3 Phishing
5.2 Legislation addressing cybercrime: local context
5.3 Digital forensics
5.3.1 Chain of custody
5.3.2 Admissibility
5.3.3 Vulnerabilities

Unit VI: Intellectual Property in the Digital Age

6.1 Copyrights, patents, and trademarks in software
6.2 Open source vs. proprietary software
6.3 Licensing agreements and compliance
6.4 Creative Commons and ethical use of digital assets

Unit VII: Course Synthesis and Professional Responsibility

8.1 Consolidation of IT law, ethical AI, and governance principles
8.2 Reflection: preparing for professional accountability in computing careers
8.3 Civic responsibility and global citizenship in technology leadership

Reading List

Essential Reading

Reynolds, G. W. (2018). *Ethics in information technology* (6th ed.). Cengage Learning.

Solove, D. J. (2008). *Understanding privacy*. Harvard University Press.

Additional Books

Association for Computing Machinery (ACM). (2018). *ACM code of ethics and professional conduct*. ACM.

Jones, A. C. (2017). *The tech-wise family: Everyday steps for putting technology in its proper place*. Zondervan.

Royal Government of Bhutan - Official IT and cybersecurity legislation and policy documents.

Date: May 2026

Module Code and Title: **ITP401 Project**
Programme: BSc in Applied Computing and Information Technology

Credit: 12
Module Tutor(s): Arun Kumar Singh, Phuntsho Wangdi, Mark Justin, Ngawang Tenzin
Module Coordinator: Arun Kumar Singh

General Objective
This project module builds on the RIT301 Research Methodology module by enabling students to transform research-based concepts into a tangible and functional product, demonstrating independent design, practical implementation, and responsible computing practices, while preparing them for real-world, industry-level deployment.

Learning Outcomes
On completion of the module, students will be able to:

i. refine a problem context based on the prototype proposal developed in research methodology module
ii. translate research findings into clear functional and non-functional product requirements
iii. design a product architecture informed by literature, feasibility analysis, and prototype insights
iv. develop a working product using appropriate tools, frameworks, and disciplined engineering practices
v. evaluate the product through systematic testing, usability studies, and performance checks
vi. apply ethical, security, privacy, and mindfulness considerations throughout product development and evaluation
vii. produce professional documentation suitable for technical teams, end-users, and industry partners
viii. collaborate effectively in a team using professional workflows, demonstrating both collective output and individual contribution
ix. present the product in a professional format, showcasing clarity, technical depth, and audience engagement
x. evaluate the transition from prototype to product, teamwork dynamics, limitations, and readiness for Capstone Internship.

Learning and Teaching Approach

Type	**Approach**	**Hours per week**	**Total credit hours**
Contact	Project Studio / Workshop (scaffolded delivery) Applied coaching	3	60
	Supervision / Milestone Clinics (team meetings)	1	
Independent study	Development, testing, documentation, reflection	4	60
Total		**8**	**120**

Assessment Approach
Progress Supervision and Viva Checks
Throughout the module, each group will present work progress during scheduled supervision clinics. At these checkpoints, we will conduct on-the-spot viva sessions to confirm authorship, individual contribution, and understanding. Viva responses form part of the assessment and are considered alongside submitted materials.

Contribution Logs

Every submission must be accompanied by a team member contribution log (repository commits, task tracking, code reviews, workflow notes). Based on viva performance and contribution evidence, team members may receive differentiated marks to ensure fairness and accountability.

Integrated Assessment

- **ITP401 Project (Product Development):** Assesses the technical build, product functionality, testing, evaluation, and documentation outcomes of the project.
- **HCI403 IT Project Management (Project Management and Professional Practice):** Assesses project management artefacts, governance structures, workflow discipline, and professional delivery practices used to manage the product development process.

Final Deliverables

At the end of the module, groups submit a complete product package (technical build, documentation, evaluation report, and handover checklist) and deliver a professional presentation/demo. Individual viva sessions confirm authorship, contribution, and reflective maturity, ensuring readiness for Capstone continuation.

Every group should be assigned a project mentor to guide and supervise the students. The tutor who will be assigned this module will be the overall project coordinator.

Phase 1 – Proposal Refinement and Planning Presentation (20%)

Students present an improved proposal based on the RIT301 module, incorporating deeper research, clearer scope, and feasibility considerations.

A. Neutral Panel Assessment (15%)

The assessment will be graded out of 30 marks and will be converted to 15% based on the following marking criteria:

5 Marks	Problem Clarity and Relevance (Clearly defined problem aligned with industry/societal needs)
6 Marks	Scope and Feasibility (Realistic scope, constraints identified, achievable outcomes)
5 Marks	Requirements Definition (Clear functional and non-functional requirements)
7 Marks	Literature and Gap Analysis (Appropriate use of sources, clear research gap identified)
4 Marks	Proposed Solution Approach (Logical, innovative solution with initial design thinking)
3 Marks	Presentation Delivery (Clear structure, visual quality, confident communication)

B. Project Mentor Assessment (5%)

The assessment will be graded out of 15 marks and will be converted to 5% based on the following criteria:

6 Marks	Proposal Refinement and Feasibility (Improvement evidence, refinement depth, practical approach, tool suitability, timeline realism)
5 Marks	Team Preparedness and Contribution (Planning, role clarity, coordination, contribution logs, consistency)
4 Marks	Engagement and Decision Justification (Feedback response, participation, justification, adaptability, communication)

Phase 2: Progress Report Presentation (20%)
Mid-semester presentation demonstrating development progress, initial implementation, and preliminary testing.

A. Neutral Panel Assessment (15%)
The assessment will be graded out of 30 marks and will be converted to 15% based on the following marking criteria:

8 Marks	Progress and Implementation Demonstration (prototype evidence, feature progress, repository commits, working modules)
7 Marks	Design and Technical Approach (system architecture, data models, UI flow, technology choice, development practices)
5 Marks	Testing and Validation (unit testing, integration testing, testing results, bug tracking)
5 Marks	Problem-Solving and Adaptation (challenges handling, solution refinement, flexibility, iterations)
5 Marks	Presentation and Communication (Structure, clarity, teamwork, delivery, visuals)

B. Project Mentor Assessment (5%)
The assessment will be graded out of 20 marks and will be converted to 5% based on the following marking criteria:

8 Marks	Development Consistency and Contribution (Regular progress, commits, task completion, contribution logs, consistency)
4 Marks	Team Coordination and Technical Effort (role clarity, collaboration, communication, initiative, independent learning)
6 Marks	Responsiveness to Feedback (feedback incorporation, adaptation, improvement, participation, justification)

Phase 3: Final Submission/Demonstration and Individual Viva Voce (60%)
This phase evaluates the final product, documentation, and presentation, including improvements made based on earlier feedback.

A. Final Project Report (40%)
Students will be required to submit a comprehensive final report documenting the complete project lifecycle, including problem analysis, requirements, design, implementation, testing, evaluation, and reflection. The report should demonstrate technical competence, professional documentation practices, ethical awareness, and evidence of project outcomes. Assessment will focus on technical quality, completeness, evaluation practices, documentation standards, problem-solving ability, and professional reflection.

5 Marks	Problem definition and objectives (clarity of objectives, contextual understanding, and relevance of the identified problem)
5 Marks	Requirements analysis and planning (quality of requirements specification, planning structure, feasibility, and risk management)
6 Marks	Design and architecture (appropriateness of architecture, design justification, workflows, models, and technical structure)
8 Marks	System/solution implementation and technical quality (technical depth, completeness of implementation, functionality, coding practices, and innovation)
6 Marks	Testing, evaluation, and optimization (testing strategy, evidence of evaluation, performance analysis, optimization, and reliability assessment)

3 Marks Industry adaptation and problem-solving (alignment with industry/project needs, adaptability, and handling of constraints)

3 Marks Documentation and deployment readiness (quality of technical documentation, deployment guidance, and usability of deliverables)

2 Marks Ethical, security, and privacy considerations (ethical awareness, data handling, security practices, and responsible computing considerations)

2 Marks Reflection and professional readiness (critical reflection, lessons learned, teamwork/professional practice insights, and career readiness)

B. Final Presentation and Demonstration (10%)

Students will be required to present and demonstrate the completed solution before academic and industry panels. The presentation should clearly communicate the project objectives, implementation approach, technical functionality, and practical relevance of the solution. Assessment will consider presentation structure, demonstration quality, technical communication, professional delivery, and ability to respond to questions. It will be assessed by a neutral panel out of 20 marks and converted to 10%.

8 Marks Presentation Structure and Delivery (logical flow, organisation, clarity, visual, engagement)

6 Marks Technical demonstration and functionality (working system, features demonstration quality, technical confidence)

2 Marks Team coordination and Communication (Teamwork, coordination, confidence)

4 Marks Impact and Justification (industry relevance, Q&A, decisions, ethics, reflection)

C. Individual Viva (10%)

Students will be required to attend an individual viva voce examination to verify authorship, technical understanding, and personal contribution to the project. They will be expected to explain implementation decisions, problem-solving approaches, technical challenges, and lessons learned during project execution. Assessment will focus on technical understanding, contribution evidence, reflective maturity, communication skills, and professional readiness.

4 Marks Contribution and Technical understanding (contribution, repository, ownership, technical knowledge, architecture)

3 Marks Problem-solving and adaptability (challenges, debugging, solution, adaptation)

3 Marks Reflection and professional Skills (learning, teamwork, growth, readiness, behavior, confidence, clarity)

Overview of assessment approaches and weighting

Areas of Assessments	**Quantity**	**Weighting (%)**
Phase 1. Proposal Refinement and Planning Presentation	1	20
Phase 2. Progress Report Presentation	1	20
Phase 3. Final Submission/Demonstration and Individual Viva Voce (60%)	1	60
Total		**100**

Pre-requisites: RIT301 Research Methodology, WEB303 Back-end Development, DMS202 Database Programming,SDP201 System Analysis and Design, SDP302 Software Engineering, and PGL203 Data Structure and Algorithm

Project Milestone
The module tutor as the overall project coordinator will approve team formation, assign project mentors and coordinate presentations and other activities and review the table accordingly. The milestones should tally with the phases recommended under the assessment approach.

Week/Area	Student Tasks	Tutor Actions
1 Orientation and Setup	Form teams (4–5), allocate roles, select one RIT301 proposal, set up repository and contribution log.	Assign supervisors, approve team formation, check repository setup and contribution log format.
2 Project Brief and Feasibility	Refine problem statement, define scope and success criteria, draft team charter, conduct feasibility analysis.	Review briefs, approve feasibility, conduct viva check, provide feedback on team charters.
3 Requirements and Systems Design	Translate research findings into functional and non-functional product requirements; traceability mapping; usability/accessibility considerations; integrate 8–10 sources; identify gaps.	Run requirements clinic, provide feedback, sign off requirements, check literature integration, viva check.
4–5 Design Studio	Prepare product architecture diagrams, data models, UI wireframes; justify design decisions and trade-offs; incorporate security/privacy by design.	Facilitate design studio, critique artefacts, approve design milestone.
6–13 Product Development and Progress Demonstration	Incremental product coding with tutor feedback; maintain repository commits; conduct unit/integration/system tests; ensure code quality and workflow discipline.	Monitor repo activity, check contribution logs, supervise progress demo, viva check, provide feedback on workflow habits.
14 Evaluation and Ethics	Conduct usability testing, performance checks, defect logging; analyse product reliability; identify security/privacy risks; reflect on ethical and GNH impacts.	Review evaluation evidence, provide feedback on risks and ethics, viva check.
15 Documentation Workshop	Draft technical documentation (architecture, setup, limitations), user guide, and handover checklist for Capstone continuation.	Run documentation workshop, check clarity and completeness of drafts.
16 Final Deliverables Package	Submit product package with documentation and deployment notes; deliver professional presentation/demo; attend individual viva; reflective discussion on teamwork, learning, and Capstone readiness.	Approve final deliverables, check handover readiness, conduct viva, assess presentation quality, confirm authorship and contribution.

Student Responsibilities

i. Define scope and requirements: Refine your RIT301 proposal into a feasible product brief with clear functional and non-functional requirements.
ii. Teamwork: Work collaboratively with clear role allocation and maintain contribution logs to ensure fair workload distribution.
iii. Repository discipline: Use version control, meaningful commits, and task tracking tools to demonstrate workflow discipline.
iv. Design and Development: Prepare architecture diagrams, data models, and UI wireframes; justify design decisions; implement the product incrementally with quality coding practices.
v. Testing and Evaluation: Conduct unit, integration, and system testing; usability and performance checks; defect logging; security/privacy risk analysis; ethical and GNH reflection.
vi. Documentation: Maintain technical notes, user guides, and handover checklists to ensure product readiness for Capstone continuation.
vii. Milestone clinics: Attend weekly supervision clinics, present progress, and respond to viva questions to confirm authorship and contribution.
viii. Final Presentation: Deliver a professional demo and presentation of the product, supported by documentation and evaluation evidence.
ix. Professional conduct: Communicate clearly, respect deadlines, and uphold ethical, security, privacy, and GNH considerations.

Tutor/Supervisor/Mentor Responsibilities (bifurcate the responsibilities of the module tutor who will be the overall project coordinator and the projects mentors)

i. Coaching: Provide applied guidance in weekly studios (design critique, repo discipline, testing strategies, documentation).
ii. Approval: Sign off on scope, feasibility, requirements, design, and milestone readiness.
iii. Supervision: Conduct weekly clinics with viva checks to confirm authorship, progress, and individual contribution.
iv. Feedback: Offer constructive critique on design, implementation, evaluation, and documentation.
v. Monitoring: Check contribution logs and repositories to ensure balanced workload distribution and workflow discipline.
vi. Preparation for Capstone: Ensure products are packaged with documentation, evaluation evidence, and handover notes for continuation into the Capstone Internship.

Structure of the Project Report (4000 - 6000 words)

1. **Cover Pages**
   i. Title page (project title, student names, supervisor/mentor, institution, date).
   ii. Declaration of originality and integrity.
   iii. Acknowledgements (industry partner, supervisor, team members).
   iv. Abstract (concise summary of problem, solution, methods, and outcomes).
   v. Table of contents, list of figures/tables.

2. **Introduction**
   i. Background and context of the industry problem.
   ii. Statement of the problem and its relevance.
   iii. Objectives of the project.
   iv. Scope and limitations.
   v. Significance of the project (academic, industry, community impact).

3. **Requirements & Planning**
    i. Functional and non-functional requirements.
    ii. Project plan (milestones, deliverables, risk analysis).
    iii. Industry/independent constraints considered.
4. **Design and Architecture**
    i. System architecture diagrams.
    ii. Data models and workflows.
    iii. UI/UX design considerations.
    iv. Security and privacy by design.
    v. Justification of design decisions and trade-offs.

5. **Implementation**
    i. Description of tools, frameworks, and technologies used.
    ii. Development process (incremental builds, repository discipline).
    iii. Key features and modules implemented.
    iv. Screenshots or code snippets (where appropriate).

6. **Testing and Evaluation**
    i. Testing methodology (unit, integration, system, usability, performance).
    ii. Test results and defect logs.
    iii. Evaluation of scalability, reliability, and security.
    iv. Industry feedback or independent validation results.

7. **Results and Discussion**
    i. Analysis of outcomes against objectives.
    ii. Adaptation to industry constraints or independent context.
    iii. Strengths and limitations of the solution.
    iv. Ethical, security, privacy, and GNH reflections.

8. **Conclusion and Future Work**
    i. Summary of achievements.
    ii. Lessons learned.
    iii. Recommendations for future improvements or continuation.
9. **Documentation and Handover**
    i. Technical documentation (setup, deployment, limitations).
    ii. User guide/manual.
    iii. Handover checklist for industry partner or future continuation.

10. **Reflection**
    i. Individual and team reflections on professional practice.
    ii. Insights into working with industry or independently.
    iii. Readiness for professional roles or further research.

11. **References and Appendices**
    i. Properly formatted references (APA style).
    ii. Appendices (logbook extracts, contribution logs, raw data, additional diagrams).

Key Preparation Tips

i. Consistency: Use a clear formatting style (fonts, headings, numbering) and 7th APA referencing style.
ii. Evidence: Include repository links, screenshots, and testing evidence.
iii. Professional Tone: Write formally, avoid casual language.
iv. Clarity: Ensure diagrams and tables are labeled and explained.
v. Validation: Cross-check that all milestones and deliverables are documented.

Reading List

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: Elements of reusable object-oriented software. Addison-Wesley Professional.* Pressman,

R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

Sommerville, I. (2015). *Software engineering* (10th ed.). Pearson.

Date: May 2026

Year 4 Semester II

Module Code and Title:	**ITP402 ACIT Capstone Project**
Programme:	BSc in Applied Computing and Information Technology
Credit:	60
Module Tutor:	Arun Kumar Singh, Phuntsho Wangdi, Mark Justin, Ngawang Tenzin
Module Coordinator:	Arun Kumar Singh

General Objectives

The Capstone Project provides students with the opportunity to apply their knowledge and skills in a real-world context by solving an industry problem. Students may work in collaboration with an industry partner on a project of professional relevance. The emphasis is on innovation, technical competence, and professional practice, with students expected to plan, adapt, and execute their projects with professional responsibility and guided autonomy. This structure ensures continuity from academic preparation to industry engagement, allowing students to demonstrate problem-solving, responsible computing, and readiness for professional practice.

Learning Outcomes

On completion of the module, students will be able to:

i. identify an industry-relevant problem in consultation with industry partners.
ii. formulate clear functional and non-functional requirements based on industry needs and feasibility analysis
iii. design a full-scale product or solution using appropriate tools, frameworks, and disciplined engineering practices
iv. implement a full-scale product or solution using appropriate tools, frameworks, and disciplined engineering practices
v. collect technical and user data required to assess solution performance, usability, scalability, and reliability
vi. apply project management practices to deliver project milestones on time and within scope

vii. demonstrate professional collaboration with industry teams while demonstrating accountability, initiative, and responsibility
viii. handle data and information with ethics, security, privacy, and integrity
ix. produce comprehensive documentation including technical reports, deployment manuals, and reflective accounts
x. present the implemented solution and its evaluation to academic and industry panels.
xi. critically assess the process of solving an industry problem, teamwork, project limitations, and professional readiness
xii. apply quality assurance practices throughout the project
xiii. integrate emerging technologies where appropriate to enhance the solution
xiv. deploy the developed solution in an appropriate environment
xv. evaluate stakeholder satisfaction with the developed solution
xvi. demonstrate compliance with relevant legal and regulatory requirements.
xvii. produce user documentation for the developed solution
xviii. apply version control practices during project development
xix. develop a maintenance plan for the implemented solution.

Learning and Teaching Approach

Type	Approach	Hours per week	Total credit hours
Contact	Industry Immersion	3	45
	Academic Supervisor	2	30
Independent study	Industry Engagement / Guided Project Execution	35	525
Total		**40**	**600**

Pedagogical Emphasis

i. Problem-based learning: Students tackle authentic industry problems, applying research and technical skills in real contexts.
ii. Experiential learning: Industry engagement provides hands-on experience with professional workflows, tools, and organisational constraints.
iii. Mentored practice: Academic and industry supervisors guide students while encouraging initiative, accountability, and professional responsibility.
iv. Reflective practice: Students critically evaluate their learning process, teamwork, project limitations, and readiness for professional practice.
v. Responsible computing: Ethical, security, privacy, and GNH considerations are integrated throughout design, implementation, and evaluation.

Assessment Approach

Assessment for the Capstone Project shall be conducted in a phased and continuous manner to ensure fairness, consistent monitoring, verification of authorship, and professional readiness. Assessment shall include supervisor evaluations, neutral panel evaluations, progress monitoring, contribution logs, repository activity, presentations, demonstrations, and viva voce examinations.

Although project development may be conducted in groups, individual marks may vary based on contribution logs, repository evidence, interim viva performance, supervisor observations, and individual participation.

Neutral panel members shall not be the direct supervisors of the assessed group to ensure fairness and objectivity in assessment.

Industry supervisors or industry representatives may participate in assessment panels where applicable to ensure industry relevance and professional alignment.

Progress supervision, interim viva checks, repository monitoring, and consultation records shall form part of the continuous assessment process throughout the semester.

Phase-wise Assessment Structure

Phase 1 Proposal Refinement and Planning Presentation 20%

Phase 2	Mid-Project Progress Demonstration and Evaluation	20%
Phase 3	Final Report, Final Presentation/Demonstration and Individual Viva Voce	60%
Total		100%

Phase 1 – Proposal Refinement and Planning Presentation (20%)
Purpose
Students will be required to refine and justify an industry-relevant project proposal by defining the problem, requirements, project scope, feasibility, and implementation plan. They will present their proposed solution approach and supporting planning documents before an evaluation panel. Assessment will be based on the quality of problem definition, planning, requirements analysis, presentation clarity, and evidence of participation in supervision and repository preparation activities.

Neutral Panel Assessment (15%)
This task will be assessed out of 20 marks and converted to 15%.

5 Marks	Problem Definition and Industry Relevance (Clarity of the industry problem, significance, contextual understanding, and relevance to professional practice.)
4 Marks	Individual Contribution and Question Handling (Ability of individual members to explain their assigned roles and responsibilities, justify problem definition and planning decisions, demonstrate understanding of project requirements and feasibility considerations, respond confidently to panel questions, and provide evidence of participation in proposal preparation, planning activities, documentation, or repository setup.)
4 Marks	Functional and Non-functional Requirements (Clarity, completeness, and appropriateness of the requirements specification.)
3 Marks	Project Planning and Risk Analysis (Project milestones, timelines, deliverables, dependency planning, and risk identification.)
2 Marks	Feasibility and Scope Clarity (Realistic project scope, achievable objectives, and feasibility justification.)
2 Marks	Presentation Clarity and Response to Questions (Clear communication, logical presentation structure, confidence, and justification of decisions during the presentation.)

Supervisor Assessment (5%)
This task will be assessed out of 10 marks and converted to 5%.

4 Marks	Participation in Consultation Sessions (Regular attendance, preparedness, active engagement during supervision, responsiveness to feedback, and evidence of progress between consultations.)
3 Marks	Contribution Logs and Repository Readiness (Evidence of repository setup, version control practices, contribution tracking, and maintenance of project documentation.)
3 Marks	Professional Conduct and Preparedness (Professional behaviour, punctuality, accountability, preparedness for meetings, and timely completion of assigned tasks.)

Phase 2 – Mid-Project Progress Demonstration and Evaluation (20%)
Purpose

Students will be required to demonstrate the progress of system development, including implemented features, testing activities, workflow discipline, and adaptation to technical or project constraints. They will present technical progress and discuss challenges, solutions, and development practices. Assessment will be based on technical progress, design consistency, testing evidence, repository activity, teamwork, and responsiveness to supervisory feedback.

Neutral Panel Assessment (15%)

5 Marks	Technical progress and working features (implementation progress, functional modules, and technical completeness at mid-stage)
4 Marks	Design and architecture consistency (alignment between implementation and approved design/architecture decisions)
3 Marks	Testing and evaluation evidence (unit testing evidence, debugging records, testing workflow, and evaluation practices)
3 Marks	Problem-solving and adaptability (handling technical challenges, adaptation to project constraints, and iterative improvements)
2 Marks	Presentation and question handling (clarity of technical explanation, confidence, and ability to justify technical decisions)
4 Marks	Individual technical contribution and question handling (ability of individual members to explain their assigned tasks, justify technical decisions, demonstrate understanding of implementation work, respond to panel questions confidently, and provide evidence of contribution to project activities, repository work, testing, or documentation)

B. Supervisor Assessment (5%)

2 Marks	Weekly progress and consultation participation (regular progress, responsiveness to supervision, and milestone completion)
3 Marks	Documentation progress (maintenance of technical records, workflow reports, and documentation quality)

Phase 3 – Final Report, Presentation/Demonstration and Individual Viva Voce (60%)

Final Project Report (40%)
Students will be required to submit a comprehensive final report documenting the complete project lifecycle, including problem analysis, requirements, design, implementation, testing, evaluation, and reflection. The report should demonstrate technical competence, professional documentation practices, ethical awareness, and evidence of project outcomes. Assessment will focus on technical quality, completeness, evaluation practices, documentation standards, problem-solving ability, and professional reflection.

5 Marks	Problem definition and objectives (clarity of objectives, contextual understanding, and relevance of the identified problem)
5 Marks	Requirements analysis and planning (quality of requirements specification, planning structure, feasibility, and risk management)
6 Marks	Design and architecture (appropriateness of architecture, design justification, workflows, models, and technical structure)
8 Marks	System/solution implementation and technical quality (technical depth, completeness of implementation, functionality, coding practices, and innovation)
6 Marks	Testing, evaluation, and optimization (testing strategy, evidence of evaluation, performance analysis, optimization, and reliability assessment)

3 Marks	Industry adaptation and problem-solving (alignment with industry/project needs, adaptability, and handling of constraints)
3 Marks	Documentation and deployment readiness (quality of technical documentation, deployment guidance, and usability of deliverables)
2 Marks	Ethical, security, and privacy considerations (ethical awareness, data handling, security practices, and responsible computing considerations)
2 Marks	Reflection and professional readiness (critical reflection, lessons learned, teamwork/professional practice insights, and career readiness)

B. Final Presentation and Demonstration (10%)

Students will be required to present and demonstrate the completed solution before academic and industry panels. The presentation should clearly communicate the project objectives, implementation approach, technical functionality, and practical relevance of the solution. Assessment will consider presentation structure, demonstration quality, technical communication, professional delivery, and ability to respond to questions. This task will be assessed by a neutral panel out of 20 marks and converted to 10%.

5 Marks	Structure and clarity of presentation (logical flow, organisation, communication clarity, and presentation quality)
6 Marks	Technical demonstration and system functionality (working demonstration, technical confidence, feature completeness, and demonstration quality)
4 Marks	Professional delivery and team coordination (professionalism, coordination, confidence, and audience engagement)
5 Marks	Question handling and justification of decisions (ability to answer questions and justify technical/project decisions)

C. Individual Viva Voce (10%)

Students will be required to attend an individual viva voce examination to verify authorship, technical understanding, and personal contribution to the project. They will be expected to explain implementation decisions, problem-solving approaches, technical challenges, and lessons learned during project execution. Assessment will focus on technical understanding, contribution evidence, reflective maturity, communication skills, and professional readiness.

3 Marks	Authorship and individual contribution (understanding of personal contribution, repository evidence, and ownership of work)
2 Marks	Technical understanding (depth of technical knowledge, architecture understanding, and implementation clarity)
2 Marks	Problem-solving and adaptability (ability to explain challenges, debugging strategies, and adaptation during project execution)
3 Marks	Reflective maturity and professional readiness (reflection on learning, teamwork, professional growth, and readiness for practice)

Continuous Assessment Components

The following shall be monitored continuously throughout the semester:

- Weekly supervision and consultation participation
- Repository commits and workflow discipline
- Contribution logs and task allocation
- Interim viva checks
- Documentation progress
- Professional conduct and accountability
- Attendance and milestone completion

Alignment with Project Milestones

Week Assessment Phase
Week 4 Phase 1 – Proposal Refinement and Planning Presentation
Week 9 Phase 2 – Mid-Project Progress Demonstration
Week 16 Phase 3 – Final Report, Presentation/Demonstration and Viva Voce

Key Features

- Balanced Evaluation: Group PPT assesses collective achievement, while viva ensures individual accountability.
- Industry Problem Focus: Both presentation and viva emphasise how the solution addresses a real industry challenge.
- Professional Readiness: Parameters highlight communication, ethical awareness, and reflective maturity — essential for industry engagement.
- Differentiated Marks: Viva and contribution logs allow supervisors to adjust marks fairly across team members.

Pass Requirements

i. Students must secure an overall pass mark of at least 50% in the Capstone Project module.
ii. Students must complete all assessment phases.
iii. Failure to complete any major assessment component may result in an incomplete grade.

Completion Rules

i. Submission of the project report by the scheduled deadline is mandatory.
ii. If a student submits the report but fails to make the presentation or attend the viva-voce, the project will be considered incomplete.
iii. In such cases, the student must contact the Academic Affairs Department for further instructions.

Presentation and Viva-Voce

i. Students will be formally notified of the schedule for presentation and viva-voce through a letter issued by the Programme Leader.
ii. Both the presentation and viva-voce will be conducted on campus at the College.
iii. The viva-voce will be conducted face-to-face only; no alternative methods will be permitted.

Overview of assessment approaches and weighting

Areas of Assessments	Quantity	Weighting (%)
A. Proposal Refinement and Planning Presentation	1	20
B. Mid-Project Progress Demonstration and Evaluation	1	20
C. Final Report, Final Presentation/Demonstration and Individual Viva Voce	1	60
Total		**100**

Capstone Project Milestone (16 Weeks)

Week/Area	Student Tasks	Tutor Actions
1 Orientation and Planning	Select an industry-relevant project topic in consultation with supervisors and industry stakeholders. Refine proposal; finalize scope, objectives, and deliverables; draft detailed project plan with milestones and risk analysis.	Approve project selection and scope; review feasibility; provide feedback on project plan and risk analysis.

2 Environment Setup	Configure development tools and access; adapt architecture to industry infrastructure or project environment; document setup challenges.	Check environment readiness; advise on infrastructure adaptation; confirm documentation of setup.
3 Requirements and Design	Define functional/non-functional requirements; prepare architecture diagrams, data models, and workflows; justify design decisions; incorporate security/privacy by design.	Run design clinic; critique artefacts; sign off requirements and design milestone.
4 Interim Review I	Submit progress report; present requirements and design artefacts; adjust plan based on feedback.	Conduct viva check; review progress report; provide formative feedback.
5–8 Implementation Phase I	Begin product development; incremental coding with repository discipline; conduct unit testing; maintain logbook and documentation.	Monitor repo activity and contribution logs; supervise coding practices; provide feedback on workflow habits.
9 Interim Review II	Present implementation progress; demonstrate working features; discuss challenges and mitigation strategies.	Review implementation demo; conduct viva check; provide feedback on technical quality and problem-solving.
10–12 Implementation Phase II and Testing	Extend product functionality; conduct comprehensive testing (functionality, usability, security, scalability); document test results.	Review testing evidence; advise on defect handling and reliability improvements.
13 Evaluation and Optimisation	Analyse collected data (performance metrics, user feedback); optimise product based on evaluation; prepare draft evaluation section for final report.	Review evaluation evidence; provide feedback on optimization strategies; confirm ethical/security compliance.
14 Documentation Drafting	Compile technical documentation, deployment manual, and user guide; prepare handover checklist.	Run documentation workshop; check clarity, completeness, and handover readiness.
15 Presentation Preparation	Prepare oral defence presentation; rehearse with supervisor; finalise slides and demo materials.	Facilitate rehearsal; provide feedback on presentation clarity and demo flow.
16 Final Defence and Submission	Deliver presentation and defence before academic/industry panel; submit final report, technical deliverables, and reflective account.	Assess final deliverables; conduct viva to confirm authorship and contribution; evaluate presentation quality and reflective maturity.

Roles and Responsibilities of

Students

i. **Project Selection**

Choose an industry problem in consultation with industry partners and academic supervisors, ensuring feasibility and professional relevance.

ii. **Proposal Stage**

Submit a refined proposal outlining the problem, objectives, scope, and planned methodology.

iii. **Planning**
Develop a detailed project plan with milestones, deliverables, and risk management strategies.

iv. **Execution**
Work incrementally, documenting progress in a logbook, and adapt the solution to project and organisational requirements and constraints.

v. **Interim Reporting**
Submit progress reports at agreed checkpoints, incorporating industry mentor or supervisor feedback.

vi. **Final Deliverables**
   a. A fully implemented product or system addressing the industry problem.
   b. A comprehensive report including research background, design, implementation, and evaluation.
   c. A professional presentation and defence before an academic and industry panel.

vii. **Professional Conduct**
Uphold ethical standards, academic integrity, accountability, teamwork, and professional conduct.

viii. **Reflection**
Include a reflective section in the final report discussing how the industry problem was addressed and lessons learned.

Industry Supervisors

i. **Industry Project Supervision**: Supervise students within the industry environment and ensure active participation in real-world project activities, workflows, and organisational practices.
ii. **Problem Definition and Industry Alignment**: Help students identify relevant industry problems, organisational requirements, operational constraints, and expected professional outcomes.
iii. **Technical Mentorship:** Provide practical guidance on tools, technologies, frameworks, workflows, deployment practices, and industry standards relevant to the project.
iv. **Workplace Integration:** Facilitate student engagement with project teams, meetings, operational processes, testing environments, and professional communication practices within the organisation.
v. **Progress Monitoring and Feedback**: Monitor student attendance, task completion, contribution, and professional performance during industry attachment or project engagement, and provide continuous feedback.
vi. **Evaluation of Professional Contribution**: Assess student initiative, technical contribution, collaboration, adaptability, workplace discipline, and readiness for professional practice.
vii. **Industry Validation:** Review the practicality, usability, scalability, and professional relevance of the developed solution or system.
viii. **Support for Testing and Deployment**: Assist students in conducting testing, validation, deployment, and evaluation activities within the organisational or operational environment where applicable.
ix. **Panel and Presentation Participation:** Participate in interim reviews, demonstrations, presentations, and final defence panels where applicable, providing industry-based evaluation and recommendations.

x. **Professional and Ethical Oversight**: Ensure that students comply with organisational policies, confidentiality requirements, ethical standards, security practices, and professional conduct expectations.

Academic Supervisors

i. Academic Guidance: Guide students in project planning, research direction, methodology selection, and alignment with programme learning outcomes.
ii. Progress Monitoring: Meet student groups regularly to review academic progress, provide formative feedback, and ensure milestone completion.
iii. Quality Assurance: Ensure that project work meets academic standards, technical rigor, ethical requirements, and institutional expectations.
ii. Assessment Support: Apply assessment rubrics consistently during proposal reviews, interim evaluations, presentations, reports, and viva voce examinations.
iii. Research and Documentation Support: Guide students in report writing, referencing, technical documentation, reflection writing, and presentation preparation.
iv. Professional Practice Oversight: Monitor academic integrity, responsible computing practices, teamwork, and professional conduct throughout the project lifecycle.
v. Coordination with Industry Supervisors: Maintain communication with industry supervisors to monitor student engagement, project feasibility, and alignment between academic and industry expectations.

Programme Leader

i. oversee the overall implementation and quality assurance of the Capstone Project module.
ii. ensure alignment of projects with programme learning outcomes and institutional standards.
iii. coordinate with academic staff, industry partners, and the Programme Committee regarding project-related matters.
iv. approve project policies, schedules, and assessment arrangements.
v. monitor fairness, consistency, and academic integrity in project supervision and assessment.
vi. address major academic or administrative issues related to project execution.

Programme Committee

i. review and endorse project guidelines, assessment rubrics, and module delivery approaches.
ii. ensure that project activities align with programme objectives, industry relevance, and institutional requirements.
iii. monitor the effectiveness of project implementation and recommend improvements where necessary.
iv. support quality assurance processes related to supervision, assessment, and student outcomes.
v. review feedback from students, supervisors, and industry partners for continuous enhancement of the module.

Project Coordinator

i. coordinate the day-to-day management of Capstone Projects.
ii. assign academic supervisors and facilitate communication between students, supervisors, and industry partners.
iii. maintain project records, schedules, and progress monitoring documentation.
iv. organize proposal reviews, interim evaluations, presentations, and viva-voce sessions.
v. ensure timely submission of reports, presentations, and assessment records.

vi. provide administrative and technical support related to project activities.
vii. monitor student progress and report concerns to the Programme Leader when necessary.

Structure of the Project Report (12000-14000 words)

1. Cover Pages

i. Title page (project title, student names, supervisor/mentor, institution, date).
ii. Declaration of originality and integrity.
iii. Acknowledgements (industry partner, supervisor, team members).
iv. Abstract (concise summary of problem, solution, methods, and outcomes).
v. Table of contents, list of figures/tables.

2. Introduction

i. Background and context of the industry problem.
ii. Statement of the problem and its relevance.
iii. Objectives of the project.
iv. Scope and limitations.
v. Significance of the project (academic, industry, community impact).

3. Literature Review

i. Review of related work, industry practices, and academic studies.
ii. Identification of gaps or challenges in existing solutions.
iii. Justification for the chosen approach.

4. Requirements and Planning

i. Functional and non-functional requirements.
ii. Project plan (milestones, deliverables, risk analysis).
iii. Industry and project constraints considered.

5. Design and Architecture

i. System architecture diagrams.
ii. Data models and workflows.
iii. UI/UX design considerations.
iv. Security and privacy by design.
v. Justification of design decisions and trade-offs.

6. Implementation

i. Description of tools, frameworks, and technologies used.
ii. Development process (incremental builds, repository discipline).
iii. Key features and modules implemented.
iv. Screenshots or code snippets (where appropriate).

7. Testing and Evaluation

i. Testing methodology (unit, integration, system, usability, performance).
ii. Test results and defect logs.
iii. Evaluation of scalability, reliability, and security.
iv. Industry feedback or project validation results.

8. Results and Discussion

i. Analysis of outcomes against objectives.
ii. Adaptation to industry constraints or project constraints.
iii. Strengths and limitations of the solution.

iv. Ethical, security, privacy, and GNH reflections.

9. Conclusion and Future Work

i. Summary of achievements.
ii. Lessons learned
iii. Recommendations for future improvements or continuation.

10. Documentation and Handover

i. Technical documentation (setup, deployment, limitations).
ii. User guide/manual
iii. Handover checklist for industry partner or future continuation.

11. Reflection

i. Individual and team reflections on professional practice.
ii. Insights into working with industry stakeholders and project teams.
iii. Readiness for professional roles or further research.

12. References and Appendices

i. Properly formatted references (APA style).
ii. Appendices (logbook extracts, contribution logs, raw data, additional diagrams).

Key Preparation Tips

i. Consistency: Use a clear formatting style (fonts, headings, numbering) and 7th APA referencing style.
ii. Evidence: Include repository links, screenshots, and testing evidence.
iii. Professional Tone: Write formally, avoid casual language.
iv. Clarity: Ensure diagrams and tables are labeled and explained.
v. Validation: Cross-check that all milestones and deliverables are documented.

Note: Project cover pages and sample letter to the industry will be prepared later.

Reading List

As per the specific requirements of each project.

Date: May 2026