Tribhuvan University

Faculty of Management

Office of the Dean



Course detail of
BITM (Bachelor of Information Technology Management) 3rd Semester

Effective from the Admission Batch 2025 AD Onwards

ITM 201: Microprocessor and Computer Architecture	3 Cr. Hrs.
ITM 202: Web Programming – I	3 Cr. Hrs.
ITM 203: Database Management System	3 Cr. Hrs.
ACC 205: Financial Accounting and Analysis	3 Cr. Hrs.
STT 204: Business Statistics	3 Cr. Hrs.

May 2026

ACC 205: FINANCIAL ACCOUNTING AND ANALYSIS

BITM 3rd Semester

Course: Core

Credit Hour: 3 **LHs: 48**

Course Objectives

This course aims to provide basic concepts of financial accounting for fundamental level of knowledge to prepare financial reports for foundation of specialization related theories. This also focuses on how to keep in right track of financial transactions so as to maintain corporate governance and sustainable development of the organizations and also focuses with a view to developing the skill of the students in preparing and analyzing financial statements.

Course Description

Initially, this course discusses conceptual foundation of financial accounting and analysis ; processing and recording business transactions; accrual accounting concept; adjusting entries, closing entries. and so on. Then after, it includes preparation and presentation of basic financial statements like income statement, statement of retained earnings, balance sheet and cash flow statement to measure the performance of the organizations as well as their analysis.

Course Details

Unit I: Introduction to Financial Accounting **3 LHs**

Fundamental knowledge of financial accounting; Concepts, Meaning, Objectives, Importance , Needs and Limitations. Financial Information: Accounting as a language and as an information system; Need and users of financial information; Sources of Accounting Information; Branches of Accounting; Forms of Business Organizations; Nature of business activity: Financing, investing and operating activities; Concept, objectives and branches of Accounting,Role of Accountants .

Unit 2: Conceptual Framework and Accounting Process of Transactions **6 LHs**

Fundamental Accounting Concepts and Assumptions: Business Entry Concept, Money Measurement Concept, Dual Aspect Concept, Going Concern Concept, Accounting Period Concept, Historical Cost Concept, Revenue Realization Concept, Accrual Concept, and Matching Concept; Generally Accepted Accounting Principles (GAAP); Accounting Conventions; Conservatism, Consistency, Full disclosure, and Materiality. Difference between a Concept and Convention.

Unit 3: Processing of Financial Reporting and Control **10 LHs**

Understanding the Concept of Accounting Process; Recording, Classifying and Summarizing; Concept of Double Entry Book Keeping System, Concept and Preparation of Accounting Equation, Transactions and their effects on Accounting Equation; Accounting events : Concept, external and internal events, cash and credit events; Source documents: Concept and role in

recording financial transactions; Cash Basis and Accrual Basis Debit, Credit and their rules; Recording process of accounting events: Journal, Ledger and Trail Balance (Undusted and Adjusted) , Computers in Accounting.

Unit 4: Processing of Accounting Reports **12 LHs**

Objectives of Accounting Reports, Characteristics of Financial Statements, Concepts of Depreciation and Methods: Straight Line and Diminishing Balance Methods; Concepts of Inventory Valuation: FIFO and LIFO Methods and weighted average cost method, Capital and Revenue : Classification of Income; Classification of Expenditures ; Classification of Receipts . Preparation of Financial Statements and Adjustment Entries: Multiple Steps of Income Statement, Statement of Retained Earning and Balance Sheet (Vertical Format) as per NFRS; Concept of Marshalling of Assets and Liabilities with consideration of IFRS; Limitations of Financial Statements.

Unit 5: Reporting and Interpreting of Cash Flow Statement **9 LHs**

Meaning and Objectives of Cash Flow Statement, Preparing of Cash Flow Statement: Direct and Indirect Method, Interpreting Cash Flow Statement; Difference between Funds Flow Statement and Cash Flow Statement

Unit 6 :Analyzing Financial Statements **8 LHs**

Concept, purpose and usefulness of financial statement analysis; Analyzing techniques: Horizontal, vertical ,trend and ratio analysis to measure and interpret the liquidity, solvency, efficiency, profitability, productivity and possibility, their uses.

Suggested Readings

Kimmel. P.. Weygandt J., & Kieso, D. (2011). *Financial Accounting: Tools for Business Decission Making (5th Edition)*

Narayanaswamy. R. (2011). *Financial Accounting: A Managerial Perspective (Fourth Edition)*. New Delhi: PHI Learning Private Limited.

Lal J. Srivastava S.(2015) *Financial Accounting: Principles and Practices*.S.Chanda

Porter. G. A... Norton, C. L. (9th Edition) . *Introduction to Financial Accounting*

Nepal accounting Standards(NAS).

Nepal Financial Reporting Standards(NFRS) and International Financial Reporting standards(IFRS)

ITM 201: Microprocessor and Computer Architecture

BITM 3rd semester

Nature of the Course: Theory + Practical **Credit Value:** 3

Total Hours: 48 (Th) +16 (Pr)

Course Description

This course introduces the organization and architecture of digital computers and the working of microprocessors. Students begin with the basics and generations of microprocessors, then study the 8085 microprocessor and its assembly language programming, basic computer organization and design, microprogrammed control, the central processing unit, pipeline and vector processing, computer arithmetic, input-output organization, memory organization and multiprocessors.

The course combines theory with laboratory work. Students write and run 8085 assembly programs on a simulator and use logic and programming simulators to model registers, micro operations, arithmetic algorithms, pipelines and cache memory. It prepares them for advanced courses in operating systems, embedded systems and computer networks, and for industry roles that need an understanding of how computer hardware works.

Course Objectives

By the end of this course, students will be able to

- Explain the components, generations and bus organization of a microprocessor system.
- Describe the architecture, instruction set, addressing modes and timing of the 8085 microprocessor and write assembly programs for it.
- Explain the organization and design of a basic computer, including micro operations, instruction cycle and interrupts.
- Compare hardwired and microprogrammed control units and describe address sequencing and microinstruction formats.
- Describe CPU organization, stack, instruction formats, addressing modes, program control and the RISC and CISC characteristics.
- Explain parallel processing, pipelining, pipeline hazards and vector processing.
- Perform hardware-level arithmetic for signed numbers, including addition, subtraction and Booth multiplication.
- Explain I/O interfacing, data transfer techniques, priority interrupts and DMA.
- Describe the memory hierarchy, cache mapping, and multiprocessor interconnection and cache coherence.

Course Details

Unit 1: Introduction to Microprocessor **3 LHs**

Introduction to microprocessor; Components of a microprocessor; Basic characteristics of different generations of microprocessors (1st to 5th generation); microprocessor system with bus organization, Application of Microprocessor.

Learning Outcomes: Describe the components of a microprocessor, the characteristics of its generations and the bus organization of a microprocessor system (Knowledge); Identify the components of a microprocessor system and draw its bus organization (Skills); Appreciate the evolution and role of microprocessors in modern computing systems (Competency)

Unit 2: 8085 Microprocessor **10 LHs**

Basic features of the 8085 microprocessor; Functional block diagram of 8085 Microprocessor with description of each block and Pin diagram of the 8085; Types of instructions: data transfer, arithmetic, logic and branching; Instruction format: 1-byte, 2-byte and 3-byte; addressing modes; instruction cycle; machine cycle; timing diagram (MOV, LDA, STA, MVI, LXI, IN, OUT); basic assembly language programming using the 8085 instruction set to perform arithmetic, logical and data transfer operations, sequential programs and simple branching operations.

Learning Outcomes: Explain the architecture, pin functions, instruction set, addressing modes, machine cycles and timing diagrams of the 8085 microprocessor (Knowledge); Draw timing diagrams of basic instructions and write, assemble and run 8085 assembly programs for arithmetic, logical, data transfer and branching operations (Skills); Apply the 8085 instruction set to develop correct and efficient assembly programs (Competency)

Unit 3: Basic Computer Organization and Design **5 LHs**

Microoperations: arithmetic, logic and shift microoperations; Instruction codes and stored program organization; Computer registers and Common bus system; computer instructions: memory-reference, register-reference and input-output; Timing and Control; Instruction cycle and its stages; Input-Output and Interrupt.

Learning Outcomes: Explain microoperations, instruction codes, computer registers, the common bus system, timing and control, the instruction cycle and interrupts of a basic computer (Knowledge); Perform register transfer and microoperations and trace the execution of memory-reference, register-reference and input-output instructions through the instruction cycle (Skills); Apply the concepts of basic computer design to understand how a stored-program computer executes instructions (Competency)

Unit 4: Microprogrammed Control **4 LHs**

Limitation of hardwired control organization; introduction to microprogrammed control organization; control memory; address sequencing; microprogram example and microinstruction format; symbolic microinstructions; symbolic microprograms; binary microprogram; example of microprogram control organization for fetch, add and indirect addressing routines.

Learning Outcomes: Explain the limitations of hardwired control, control memory, address sequencing and microinstruction formats (Knowledge); Write symbolic and binary microprograms for simple routines such as fetch, add and indirect addressing (Skills); Appreciate the flexibility of microprogrammed control in the design of control units (Competency)

Unit 5: Central Processing Unit **7 LHs**

Components of CPU; general register organization; stack organization (register stack and memory stack); instruction formats (zero, one, two and three address); addressing modes; data transfer instructions with examples, data manipulation instructions with examples and program control instructions: status bit conditions, conditional branch instructions, unconditional branch instructions, subroutine call and return, and program interrupt (external, internal and software interrupt); basic characteristics of RISC and CISC; overlapped register windows.

Learning Outcomes: Explain CPU components, register and stack organization, instruction formats, addressing modes, program control instructions and the characteristics of RISC and CISC (Knowledge); Evaluate arithmetic expressions using stack and different address instruction formats, and identify addressing modes in given instructions (Skills); Compare RISC and CISC designs and choose suitable instruction formats and addressing modes for given requirements (Competency)

Unit 6: Pipeline and Vector Processing **5 LHs**

Introduction to parallel processing; Flynn's classification of computers; introduction to pipelining; general consideration of a pipeline system with speedup ratio; types of pipeline: arithmetic pipeline for floating-point addition and subtraction, instruction pipeline with four segments; pipeline hazards (structural, data and control) with their solutions; concept of vector processing.

Learning Outcomes: Explain Flynn's classification, pipelining, types of pipelines, pipeline hazards and their solutions, and vector processing (Knowledge); Draw space-time diagrams, calculate the speedup of a pipeline and identify and resolve hazards in an instruction sequence (Skills); Appreciate the role of pipelining and parallel processing in improving processor performance (Competency)

Unit 7: Computer Arithmetic **4 LHs**

Representation of unsigned and signed numbers; hardware organization, algorithm and example of signed-magnitude addition and subtraction and 2's complement addition and subtraction; hardware organization, algorithm and example of signed-magnitude multiplication and 2's complement multiplication (Booth algorithm).

Learning Outcomes: Explain the representation of signed numbers and the hardware organization and algorithms for addition, subtraction and multiplication (Knowledge); Perform signed-magnitude and 2's complement addition and subtraction and multiply numbers using the Booth algorithm (Skills); Apply arithmetic algorithms to understand how processors perform calculations in hardware (Competency)

Unit 8: Input-Output Organization **4 LHs**

Peripheral devices and I/O interface; asynchronous data transfer: strobe and handshaking; modes of transfer: programmed I/O and interrupt-initiated I/O; priority interrupt: daisy chaining and parallel priority; direct memory access (DMA); input-output processor (IOP); CPU-IOP communication.

Learning Outcomes: Describe I/O interfaces, asynchronous data transfer, modes of transfer, priority interrupts, DMA and the input-output processor (Knowledge); Compare programmed, interrupt-initiated and DMA transfer and explain the working of daisy-chain and parallel priority schemes (Skills); Select suitable data transfer techniques for different I/O devices and applications (Competency)

Unit 9: Memory Organization and Multiprocessor System **6 LHs**

Memory hierarchy; main memory: RAM and ROM chips, memory address map, memory connection to CPU; associative memory; cache memory: mapping techniques and write policies; Multiprocessor system; Characteristics of multiprocessors; interconnection structures: bus, crossbar switch and multistage network; interprocessor arbitration and communication; cache coherence.

Learning Outcomes: Explain the memory hierarchy, RAM and ROM organization, associative and cache memory, multiprocessor characteristics, interconnection structures and cache coherence (Knowledge); Design a memory address map, connect memory chips to the CPU, and work out cache address formats and hit or miss behaviour for different mapping techniques (Skills); Apply memory organization and multiprocessor concepts to evaluate the performance of computer systems (Competency)

Pedagogical Strategies

- Lectures with diagrams and worked numerical examples
- Hands-on laboratory sessions on an 8085 simulator and logic simulator with guided and open-ended exercises
- Problem-based learning using real-world scenarios
- Group discussions and peer review of programs and designs
- Multimedia presentations and animations to visualize CPU operation, pipelines and cache behaviour
- Mini project with iterative design and feedback cycles
- Continuous assessment and formative feedback

Mode of delivery

- Lecture sessions (Theory)
- Demonstration
- Laboratory work (Practical)
- Mini project

Internal assessment methods and types (40%)

Assessment Type	Weightage	Details
Class participation & attendance	10%	Contribution to discussions, engagement in class activities
Quizzes/short tests	15%	Periodic quizzes to assess comprehension
Practical/Project	20%	Lab sessions and mini project
Mid-term examination	25%	Written test
Pre-board examination	30%	Comprehensive written test covering all units

External assessment methods and types (60%)

Out of the total 60% allocated for final assessment, 40% will be assigned to the written/board examination to evaluate students' abilities in remembering, understanding, applying, analyzing, evaluating, and creating**.** The remaining 20% will be assigned to the final practical examination to assess hands-on programming skills and competency.

Mapping course: Learning outcomes and program learning

Course Learning Objective (CLO) Dimensions	Knowledge (K)	Skills (S)	Competence (C)
Total Learning	35%	40%	25%

Laboratory work

Students are required to complete hands-on exercises for every unit. The laboratory work should be done using 8085 Microprocessor kit and a programming language such as C . The following practical tasks are prescribed:

1. Write an 8085 assembly language program to add two 8-bit numbers stored in memory locations 2050H and 2051H. Store the result in memory location 2052H.
2. Write an 8085 assembly language program to subtract two 8-bit numbers stored in memory locations 2050H and 2051H. Store the result in memory location 2052H.
3. Write an 8085 assembly language program to multiply two 8-bit numbers using repeated addition.
4. Write an 8085 assembly language program to divide two 8-bit numbers using repeated subtraction.
5. Write an 8085 assembly language program to find the largest number among three 8-bit numbers stored in consecutive memory locations.
6. Write an 8085 assembly language program to find the smallest number among three 8-bit numbers stored in consecutive memory locations.
7. Write an 8085 assembly language program to arrange five 8-bit numbers in ascending order.
8. Write an 8085 assembly language program to arrange five 8-bit numbers in descending order.
9. Write an 8085 assembly language program to find the sum of N 8-bit numbers stored in consecutive memory locations.

10. Write an 8085 assembly language program to count the number of positive and negative numbers in a given set of 8-bit signed numbers.
11. Write an 8085 assembly language program to find the 1's complement and 2's complement of an 8-bit number.
12. Write an 8085 assembly language program to transfer a block of data from one memory location to another.
13. Write an 8085 assembly language program to count the number of even and odd numbers in a given array of 8-bit numbers.
14. Write an 8085 assembly language program to find the factorial of a given 8-bit number using repeated multiplication.
15. Write an 8085 assembly language program to perform different Logical microoperation
16. Write a program in C language to perform different computer arithmetic operation like Signed magnitude addition and subtraction,2’s Complements addition and subtraction, Signed magnitude and 2’s complement multiplication.

Suggested Readings

Mano, M. M. (2007). Computer system architecture (3rd ed.). Pearson Education.

Gaonkar, R. S. (2013). Microprocessor architecture, programming, and applications with the 8085 (6th ed.). Penram/Pearson.

Stallings, W. (2016). Computer organization and architecture: Designing for performance (10th ed.). Pearson.

Hayes, J. P. (1998). Computer architecture and organization (3rd ed.). McGraw-Hill.

Hamacher, C., Vranesic, Z., & Zaky, S. (2002). Computer organization (5th ed.). McGraw-Hill.

Brey, B. B. (2009). The Intel microprocessors: 8086/8088 and later (8th ed.). Pearson.

TRIBHUVAN UNIVERSITY
FACULTY OF MANAGEMENT
Office of the Dean
Model Question 2026

Full Marks: 60
Pass Marks: 30
Credit Hour: 3
Time: 3 Hrs

BITM / Third Semester / ITM 201: Microprocessor and Computer Architecture

Candidates are required to answer all the questions in their own words as far as practicable. Figures in brackets indicate full marks.

Group "A"

Concept Based Brief Answer Questions

Answer the following questions: (Attempt All) ***[5 × 2 = 10]***

1) What are the functions of the address bus and the data bus in a microprocessor system?
2) State any two limitations of hardwired control organization.
3) Write any two differences between RISC and CISC.
4) What is a pipeline hazard? List its types.
5) What is the cache coherence problem in a multiprocessor system?

Group "B"

Problem Solving / Short Answer Questions

Answer any SIX Questions: ***[6 × 5 = 30]***

6) Write an 8085 assembly program to add two 8-bit numbers stored at memory locations 2050H and 2051H and store the sum, including the carry, at 2052H and 2053H. Add a comment to each instruction.
7) Explain the stages of the instruction cycle of the basic computer with a flowchart. How is an indirect-address memory-reference instruction handled?
8) Explain the microprogrammed control organization with a neat block diagram. Describe the role of the control memory and the control address register.
9) Write the instructions to evaluate X = (A + B) × (C + D) using (i) three-address, (ii) two-address, (iii) one-address and (iv) zero-address instruction formats.
10) Explain a four-segment instruction pipeline with a space-time diagram. Calculate the speedup of a 4-segment pipeline for 100 tasks.
11) Multiply (+7) and (−3) using the Booth algorithm. Use 4-bit registers and show each step of the hardware algorithm in a table.
12) What is asynchronous data transfer? Explain the source-initiated strobe and handshaking methods with block diagrams and timing diagrams.

Group "C"

Comprehensive / Extended Problem / Case / Situation Analysis / Long Answer Questions

Answer any TWO Questions: ***[2 × 10 = 20]***

13) A block of 10 bytes is stored in memory starting at 2100H.
    a. Write an 8085 assembly program to find the largest number in the block and store it at 2200H. (5)
    b. State the addressing mode of each instruction used in your program. (2)
    c. Draw the timing diagram of the instruction MVI B, 05H and state the number of machine cycles and T-states it needs. (3)

14) A computer has a main memory of 64K words and a cache memory of 1K words. The block size is 8 words.
    a. Show the address format (tag, index and word fields) for direct mapping, and explain how a memory address is checked for a hit. (4)
    b. Show the address format for 2-way set-associative mapping and compare it with direct mapping. (3)
    c. Explain the write-through and write-back policies with one advantage and one disadvantage of each. (3)

15) A computer system has several I/O devices that need to transfer data to memory.
    a. Explain how the daisy-chaining priority interrupt method decides which device is served first, with a neat diagram. (5)
    b. Explain how a DMA controller transfers a block of data between an I/O device and memory with a block diagram, and state why DMA is better than interrupt-initiated I/O for large data transfers. (5)

ITM 202: Web Programming I

BITM 3rd Semester

Nature of the Course: Theory + Practical **Credits:** 3

Total Hours: 48(Th) +16(Pr)

Course Description

This course introduces the technologies and design practices used to build the front end of modern websites. Students begin with how the web works, then learn to structure pages with HTML5, style and lay them out with CSS3, apply user interface and user experience (UI/UX) design principles, and add interactivity with JavaScript.

The course combines theory with extensive laboratory work. By the end, students design, build, validate and publish a responsive multi-page website. It prepares them for server-side web development courses and for industry roles in web design and development.

Course Objectives

By the end of this course, students will be able to

- Explain how the Internet, HTTP and the client/server model deliver web pages.
- Create structured, semantic and accessible web pages using HTML5 elements, links, media, tables and forms.
- Style and lay out web pages using CSS3 selectors, the box model, Flexbox and Grid.
- Build responsive web pages that adapt to different screen sizes using media queries and a mobile-first approach.
- Apply UI/UX design principles to plan, wireframe, prototype and evaluate a website.
- Write JavaScript programs using variables, control statements, functions, arrays and objects.
- Manipulate the DOM, handle events and validate form data using JavaScript.
- Fetch and display JSON data, and publish a website using version control and a hosting service.
- Develop a complete, attractive and user-friendly website using good web development practices.

Course Details

Unit 1: Web Essentials and Tools 7 LHs

Introduction to the Internet and the World Wide Web: brief history of the Internet and the web; Internet vs WWW; Internet service providers; web page, web site and web application; static and dynamic web sites; Web 1.0, Web 2.0 and Web 3.0; career paths in web design and development; Internet protocols: TCP/IP model and its layers; IP addressing (IPv4 and IPv6); TCP vs UDP; ports; DNS and domain names; the domain name resolution process; higher-level protocols: FTP, SMTP, POP3, IMAP and HTTPS with TLS basics; HTTP request-response cycle; request message (method, headers, body) and response message (status line, headers, body); HTTP methods (GET, POST, PUT, DELETE); status code groups (1xx to 5xx); HTTP vs HTTPS; URL, URI and URN, URL structure (scheme, host, port, path, query string, fragment); absolute and relative

URLs; Web architecture: client/server architecture; 1-Tier, 2-Tier and 3-Tier models; web client, roles of the browser, web server, application server and database server; web page rendering; web hosting basics; web standards bodies (W3C and WHATWG); Development environment: code editors (VS Code), useful extensions and Live Server; project folder structure and file naming conventions; browser developer tools (Elements, Console, Network and Sources panels); validating pages with the W3C validator; cross-browser testing basics; Version control: need for version control; Git concepts (repository, commit, branch, merge); basic commands (init, add, commit, status, log, push, pull); creating a GitHub repository and README; the .gitignore file; collaboration with pull requests; introduction to web security awareness (HTTPS, same-origin policy, safe handling of user input)

Learning Outcomes: Explain how the Internet, TCP/IP, DNS, HTTP and client/server architecture work together to deliver a web page, and describe HTTP methods, status codes and basic Git concepts (Knowledge); Set up VS Code and Live Server, inspect pages using browser developer tools, validate pages with the W3C validator, and manage and publish a project using Git and GitHub (Skills); Appreciate the role of web standards, development tools and secure practices in building reliable websites (Competency)

Unit 2: HTMLand HTML5: Structuring Documents **8 LHs**

Introduction to HTML and HTML5; Structure of an HTML document: doctype, html, head, title and body; Elements and attributes; Writing comments; Headings, paragraphs, line breaks and horizontal rules; Text formatting and phrase elements (strong, em, abbr, blockquote, q, cite, code, kbd, var, samp, address, sub, sup); Block and inline elements; Grouping with div and span; Lists: ordered, unordered, definition and nested; Hyperlinks: absolute and relative paths, email links and page anchors; Images: img element, src, alt, width and height attributes, image formats (JPEG, PNG, GIF, SVG, WebP), images as links; Tables: table, tr, th, td, caption, thead, tbody, tfoot, colspan and rowspan; Forms and form controls: form element, action and method attributes, enctype attribute, text, password, radio buttons, checkboxes, select boxes, textarea, file select, hidden controls and buttons(summit and reset); HTML5 input types (email, date, number, range, url); label, fieldset and legend; placeholder, required and pattern attributes; Audio and video elements; iframe and its attributes; Head and meta tags; HTML5 semantic elements: header, nav, main, section, article, aside and footer; class and id attributes; Accessibility basics: alt text, form labels and keyboard focus.

Learning Outcomes: Describe the structure of an HTML5 document and the purpose of its elements, attributes, semantic tags and form controls (Knowledge); Create web pages with text, lists, links, images, tables, audio, video and forms using HTML5 elements and attributes (Skills); Apply semantic and accessible markup (alt text, form labels, keyboard focus) to build well-structured web pages (Competency)

Unit 3: CSS3: Styling, Layout and Responsive Design **10 LHs**

Introduction to CSS; CSS syntax; Inserting CSS: inline, internal and external; Writing comments; Cascade, inheritance and specificity; Selectors: universal, type, class, id, descendant, child, adjacent sibling, general sibling, attribute, pseudo-class and pseudo-element selectors; using different color values and backgrounds: background-color,

image, repeat, position, attachment and shorthand; Text and fonts: font-family, font-size, font-weight, font-style, text-align, text-decoration, text-transform, letter and word spacing, web fonts; Styling links, lists and tables; Units: px, em, rem, %, vw and vh; The box model: content, padding, border and margin; Height, width, max-width and overflow; The display and visibility properties; Positioning element with: static, relative, absolute, fixed and sticky, z-index; float and clear; Flexbox layout; CSS Grid layout; Transitions and simple animations; Responsive web design: viewport meta tag, fluid layouts, responsive images, media queries, mobile-first approach; Introduction to CSS frameworks (Bootstrap)

Learning Outcomes: Explain CSS selectors, the cascade, specificity, the box model, positioning, Flexbox, Grid and responsive design techniques (Knowledge); Style web pages and build responsive layouts using CSS3, Flexbox, Grid and media queries (Skills); Apply CSS best practices to develop attractive and consistent websites that work on all screen sizes (Competency)

Unit 4: UI/UX Design for the Web **8 LHs**

Introduction to UI and UX; Human centred design process; Understanding the audience: who will visit, why they visit and what goal they need to achieve; User research basics, personas and scenarios; Information architecture: prioritizing and grouping content, card sorting, site maps; Navigation patterns and user flows; Visual design principles: layout, grid, visual hierarchy, color theory and contrast, typography, spacing and imagery; Usability principles: Nielsen’s ten heuristics, Gestalt principles, Fitts’s law and Hick’s law; Accessibility: WCAG principles (perceivable, operable, understandable, robust), color contrast and keyboard navigation; Interface patterns: navigation bars, cards, forms, buttons, feedback and error messages, microcopy; Introduction to Golden Ration, Its importance in UIUX design; Mobile UX and touch-friendly design; Sketching and wireframing for desktop and mobile; Low-fidelity and high-fidelity prototyping using Figma; Reusable components and styles in Figma (introduction to design systems and design tokens); Peer feedback and iterating on a design.

Learning Outcomes: Describe the human-centred design process, information architecture, visual design principles, usability heuristics, Gestalt principles, Fitts's and Hick's laws, and the WCAG accessibility principles (Knowledge); Create user personas, scenarios, site maps, user flows, paper wireframes and Figma prototypes, apply interface patterns and touch-friendly design, and evaluate an interface using usability heuristics (Skills); Apply user-centred design thinking, accessibility and peer feedback to plan and improve usable, accessible and visually consistent websites centred design thinking to plan and improve usable, accessible and visually consistent websites (Competency)

Unit 5: JavaScript and the DOM **10 LHs**

Introduction to JavaScript; Adding JavaScript to a page: inline, internal and external scripts; Output methods; Comments; Variables (var, let, const) and data types; Type conversion; Operators: arithmetic, assignment, comparison, logical and string; Control statements: if, if else, switch; Loops: for, while, do while, for of, for in, break and continue; Functions: declaration, expression, arrow functions, parameters, return values

and scope; Arrays and array methods (push, pop, forEach, map, filter, every); Objects, methods and classes; Built-in objects: String, Number, Math, Date and Array; Document Object Model (DOM): selecting, creating, changing and removing elements; Browser Object Model (BOM): window, location, history and navigator; JavaScript Events: event listener, Event Object, preventDefault(), event propagation, phases of events(capture, target and bubbling); Event handling: click, change, submit, keyboard and mouse events, event listener; Form validation: when to validate, checking text fields, string length, email format, select box options, radio buttons and checkboxes, displaying error messages; Error handling with try, catch, finally and throw; Cookies, localStorage and sessionStorage; Modern JavaScript (ES6): template literals, destructuring, spread operator and modules; Introduction and implementation of Jquery.

Learning Outcomes: Explain JavaScript syntax, data types, control statements, functions, arrays, objects, the DOM and the event model (Knowledge); Write JavaScript programs that manipulate the DOM, respond to events, validate form input and store data in the browser (Skills); Apply JavaScript and error handling to develop interactive, error-free and user-friendly web pages (Competency)

Unit 6: Web Data and Publishing **5 LHs**

JSON and data handling: JSON syntax and data types, objects, arrays and nesting, JSON.parse() and JSON.stringify(); Asynchronous programming in JavaScript and APIs: promises, async and await, using Fetch for GET and POST requests, REST endpoints, query parameters, request headers, HTTP status codes, CORS basics and inspecting requests in browser DevTools; Dynamic interfaces: displaying API data with the DOM and handling loading, empty and error states; Performance, accessibility, SEO and security: image optimization, lazy loading, Lighthouse, semantic HTML, page titles and descriptions, keyboard accessibility and safely rendering untrusted content to prevent cross-site scripting (XSS); Deployment and final project: deploying with web server, deploying with github pages and firebase hosting, testing across browsers and screen sizes, checking accessibility and broken links, and presenting the finished project.

Learning Outcomes: Describe JSON, asynchronous JavaScript, the Fetch API, SEO, performance and basic web security (Knowledge); Fetch and display data from an API with loading, empty and error states, optimize a site using Lighthouse and publish it online(Skills); Apply data handling, performance, security and deployment practices to deliver a complete, production-ready website (Competency).

Pedagogical Strategies

- Lectures with live coding demonstrations
- Hands-on laboratory sessions with guided and open-ended exercises
- Problem-based learning using real-world scenarios
- Pair programming and peer code review activities
- Design studio sessions with sketching, wireframing and peer critique
- Mini project with iterative design and feedback cycles
- Guest lectures from web design and development professionals
- Continuous assessment and formative feedback

Mode of delivery

- Lecture sessions (Theory)
- Demonstration
- Laboratory work (Practical)
- Mini project

Internal assessment methods and types (40%)

Assessment Type	Weightage	Details
Class participation & attendance	10%	Contribution to discussions, engagement in class activities
Quizzes/short tests	15%	Periodic quizzes to assess comprehension
Practical/Project	20%	Lab sessions and mini project
Mid-term examination	25%	Written test
Pre-board examination	30%	Comprehensive written test covering all units

External assessment methods and types (60%)

Out of the total 60% allocated for final assessment, 40% will be assigned to the written/board examination to evaluate students' abilities in remembering, understanding, applying, analyzing, evaluating, and creating**.** The remaining 20% will be assigned to the final practical examination to assess hands-on programming skills and competency.

Mapping course: Learning outcomes and program learning

Course Learning Objective (CLO) Dimensions	Knowledge (K)	Skills (S)	Competence (C)
Total Learning	35%	40%	25%

Laboratory work

Students are required to complete hands-on exercises for every unit. The following practical tasks are prescribed:

- Unit 1 Lab: Set up a code editor; inspect pages and network requests with browser developer tools; create a Git repository on GitHub and publish a first page
- Unit 2 Lab: Create pages with lists and navigation menus, pages with internal and external links, a photo gallery, a page with audio and video, and a data entry form with different controls. Creating a table with all its attributes.
- Unit 3 Lab: Style the same HTML pages with different looks using external CSS; build page layouts with Flexbox and Grid; create a fully responsive page using media queries
- Unit 4 Lab: Prepare a user brief, site map, wireframes and a Figma prototype for the mini project; perform a heuristic evaluation of a peer's design

- Unit 5 Lab: Programs using variables, control statements, functions, arrays and objects; DOM manipulation and event handling; form validation (empty fields, specific string length, email format)
- Unit 6 Lab: Fetch JSON data from a public API and display it on a page; store user preferences in localStorage; test performance with Lighthouse; publish the final website online
- Mini Project: Develop a multi-page, responsive website with an attractive look and feel and a data entry form with JavaScript validation, based on the student's own UI/UX design

Suggested Readings

Duckett, J. (2011). *HTML & CSS: Design and build websites*. Wiley.

Duckett, J. (2014). *JavaScript and jQuery: Interactive front-end web development*. Wiley.

Robbins, J. N. (2018). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics* (5th ed.). O'Reilly Media.

Jackson, J. C. (2006). *Web technologies: A computer science perspective*. Pearson Prentice Hall.

Krug, S. (2014). *Don't make me think, revisited: A common sense approach to web usability* (3rd ed.). New Riders.

Norman, D. (2013). *The design of everyday things* (Rev. ed.). Basic Books.

Mozilla. (n.d.). *MDN web docs*. https://developer.mozilla.org

World Wide Web Consortium. (2023). *Web content accessibility guidelines (WCAG) 2.2*. https://www.w3.org/TR/WCAG22/

TRIBHUVAN UNIVERSITY
FACULTY OF MANAGEMENT
Office of the Dean
Model Question 2026

Full Marks: 60
Pass Marks: 30
Credit Hour: 3
Time: 3 Hrs

BITM / Third Semester / ITM 203: Web Programming I

Candidates are required to answer all the questions in their own words as far as practicable. Figures in brackets indicate full marks.

Group "A"

Concept Based Brief Answer Questions

Answer the following questions: (Attempt All) ***[5× 2 = 10]***

1) What is the difference between the Internet and the World Wide Web?
2) What is the purpose of the alt attribute in the img element? Why is it important?
3) List the differences between margin and padding in CSS.
4) State Fitts's law.
5) What is the difference between localStorage and sessionStorage?

Group "B"

Problem Solving / Short Answer Questions

Answer any SIX Questions: ***[6 × 5 = 30]***

6) Explain the HTTP request and response cycle with a suitable diagram.
7) Write HTML code for a web page that uses the semantic elements header, nav, main, section, aside and footer.
8) Differentiate between Flexbox and CSS Grid. Write CSS code to create a three-column layout using Grid.
9) What is information architecture? Explain card sorting and site maps with suitable examples.
10) What is a clickable prototype? Explain the steps to build a low-fidelity wireframe and turn it into a prototype in Figma.
11) Write a JavaScript program that stores a list of numbers in an array and displays only the numbers greater than 50 using the filter() and forEach() methods.
12) Write JavaScript code to fetch a list of users from a JSON API using fetch() and display their names in a list. Show a message while loading and when an error occurs.

Group "C"

Comprehensive / Extended Problem / Case / Situation Analysis / Long Answer Questions

Answer any TWO Questions: ***[2 × 10 = 20]***

13) A local café wants a new website that shows its menu, opening hours, location and an online table reservation form. Using the human-centred design process, explain how you would (a) identify the target audience and create one persona, (b) prepare a site map and user flow for table reservation, (c) design a mobile wireframe for the home page.

14) Create a to-do list web page with an input box, an "Add" button and a list area. Write HTML, CSS and JavaScript so that a new task is added to the list when the button is clicked, an empty task is rejected with an error message, a task is removed when it is clicked, and the tasks are saved in localStorage and restored when the page is reloaded.

15) Write HTML and CSS code to create a responsive gallery of cards using CSS Grid. Each card should contain an image, a title, a short description and a button. The gallery should show three cards per row on large screens, two cards per row on screens up to 900px wide, and one card per row on screens up to 600px wide. Also explain how a mobile-first approach would change the way you write the media queries.

ITM 203: Database Management System

BITM 3rd Semester

Nature of the Course: Theory + Practical **Credit Value:** 3
Total Hours: 48(Th) +16(Pr)

Course Description

This course provides a comprehensive introduction to database management systems, covering both relational and non-relational technologies. Students learn database concepts and architecture, conceptual design with Entity-Relationship (ER) and Enhanced ER models, the relational model, normalization, Structured Query Language (SQL), transaction processing, concurrency control and recovery, and NoSQL databases using MongoDB.

The course connects database theory with practice. Equal emphasis is placed on theoretical understanding and hands-on laboratory work in SQL and NoSQL, and students complete a project that designs and implements a database using both a relational and a NoSQL component, preparing them for advanced data-oriented courses and industry roles.

Course Objectives

By the end of this course, students will be able to

- Understand database concepts, the three-schema architecture, data independence and the roles of database users.
- Design conceptual models using ER and Enhanced ER diagrams and convert them into relational tables.
- Explain the relational model, keys and integrity constraints.
- Apply functional dependencies and normal forms (1NF to BCNF) to remove redundancy and anomalies.
- Write SQL statements for defining, querying and modifying data, and create views, stored procedures, triggers and indexes.
- Explain transaction properties, serializability, locking, timestamp ordering and recovery techniques.
- Compare relational and NoSQL databases and select a suitable data model for an application.
- Create, query, index, aggregate and model data in MongoDB, and explain replication and sharding.
- Describe database security, performance tuning, distributed databases, data warehousing, data mining and big data.
- Develop a complete database application using good design practices.

Course Details

Unit 1: Database Concepts and Architecture **4 LHs**

Data, database and Database Management System (DBMS); file-processing system vs database approach; advantages of databases; database users: database administrator, designers, application programmers and end users; schemas and instances; categories of data models (conceptual, logical, physical); three-schema architecture; logical and physical data independence; database languages (DDL, DML, DCL, TCL) and interfaces; database system environment and components; centralized and client/server (1-, 2- and 3-tier) architectures; Data models and its classification: relational, object-oriented, hierarchical, network and NoSQL.

Learning Outcomes: Describe database concepts, the three-schema architecture, data independence and DBMS components (Knowledge); Distinguish the file-processing and database approaches and identify suitable DBMS architectures and users for a given system (Skills); Appreciate the role of a DBMS in managing data for modern applications (Competency)

Unit 2: Data Modelling Using the ER/EER Model and the Relational Model **10 LHs**

High level conceptual models and the database design process; entity types and entity sets; attributes (simple, composite, multivalued, derived, identifier/key); keys (super key, candidate key and primary key); relationship types and sets, roles and degree; mapping cardinalities and participation constraints; weak entity types; relationship types of degree higher than two; ER diagrams and naming conventions; ER design issues; Enhanced ER (EER) concepts: specialization and generalization; disjoint and overlapping constraints; total and partial specialization constraints; aggregation; relational model concepts: relation, tuple, attribute, domain, degree and cardinality; characteristics of relations; relational schema and relational database state; relational keys (super key, candidate key, primary key, alternate key, composite key and foreign key); integrity constraints: domain, key, entity integrity and referential integrity; handling constraint violations during insert, delete and update; mapping ER/EER diagrams into relational tables, including strong and weak entities, simple, composite and multivalued attributes, 1:1, 1:N and M:N relationships, higher-degree relationships, and specialization and generalization; case study: designing an ER/EER schema for a real-world system and converting it into relational tables.

Learning Outcomes: Explain the concepts of ER/EER modelling, relational models, entities, attributes, relationships, keys and integrity constraints (Knowledge); Design ER and EER diagrams based on given requirements and convert them into relational tables by applying appropriate mapping rules (Skills); Apply conceptual modelling, relational database principles and integrity constraints to design sound, consistent and reliable database schemas for real-world problems (Competency).

Unit 3: Database Normalization **5 LHs**

Informal design guidelines for relation schemas; update, insertion and deletion anomalies; functional dependencies; closure of a set of functional dependencies and of attribute sets; properties of relational decomposition: lossless-join and dependency preservation; normal forms based on primary keys: First, Second and Third Normal Forms; Boyce-Codd Normal Form (BCNF) and comparison of BCNF and 3NF; multivalued dependency and Fourth Normal Form

Learning Outcomes: Explain anomalies, functional dependencies, normal forms and decomposition properties (Knowledge); Compute attribute closures and normalize relations up to BCNF with lossless, dependency-preserving decomposition (Skills); Apply normalization to produce efficient, redundancy-free database designs (Competency)

Unit 4: Structured Query Language (SQL) **10 LHs**

Data Definition Language: data and domain types, create, alter , drop, rename, truncate, schema definition; specifying constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT and referential actions; Data Manipulation Language: SELECT, INSERT, UPDATE and DELETE; FROM and WHERE clauses, SQL alias , tuple variables, string operations (LIKE), ordering the display of tuples (ORDER BY), duplicate tuples (DISTINCT); set operations (UNION, INTERSECT, EXCEPT); aggregate functions, GROUP BY and HAVING; null values; joined relations: INNER, OUTER, CROSS and NATURAL joins and join conditions; nested subqueries: set membership, set comparison, test for empty relations and test

for absence of duplicate tuples; views: creating, querying and updating views; Transaction Control Language (Start transaction/ begin, commit, rollback, savepoint, rollback to) and Data Control Language (GRANT, REVOKE); Introduction to stored procedures and functions; DML triggers(insert, update, delete); indexing: Introduction and example of clustered and non-clustered indexes; Introduction to embedded and dynamic SQL

Learning Outcomes: Describe SQL data types, constraints, query constructs, views, stored procedures, triggers and indexes (Knowledge); Write SQL statements to create schemas, manipulate data, join tables, write nested queries and create views, basic triggers (Skills); Apply SQL to solve data management problems in a relational DBMS (Competency)

Unit 5: Transaction Processing, Concurrency Control and Recovery **6 LHs**

Introduction to transaction processing; transaction and system concepts; transaction states; desirable properties of transactions (ACID); schedules: serial, serializable, conflict-serializable and recoverable schedules; concurrency problems: lost update, dirty read, unrepeatable read and phantom; lock-based protocols and two-phase locking and its type; timestamp-ordering protocol; recovery concepts; log-based recovery; deferred update (NO-UNDO/REDO); immediate update (UNDO/REDO); shadow paging; database backup and recovery from failures

Learning Outcomes: Explain ACID properties, serializability, concurrency control protocols and recovery techniques (Knowledge); Test schedules for serializability and apply locking, timestamp ordering and log-based recovery to given scenarios (Skills); Apply transaction management concepts to keep databases consistent during concurrent access and failures (Competency)

Unit 6: NoSQL Databases **8 LHs**

Introduction to NoSQL Databases: Limitations of relational databases at web scale; vertical and horizontal scaling; schema flexibility and join overhead in distributed environments; definition and historical development of NoSQL; RDBMS vs. NoSQL; ACID vs. BASE (Basically Available, Soft State, Eventual Consistency); CAP theorem and its consistency, availability, and partition-tolerance trade-offs; eventual consistency. NoSQL Data Models and Selection Criteria: Key-value, document, wide-column (column-family), and graph data models; representative NoSQL database systems and their applications; advantages, limitations, and criteria for selecting and avoiding NoSQL databases. Document Model and MongoDB Fundamentals: Documents and collections; JSON and BSON data representation; flexible schema; overview of the _id field and ObjectId; document modeling using embedding and referencing;Basic Operations and Indexing in document model: Insertion: insertOne(), insertMany(); Querying: find(), findOne(), comparison operators ($gt, $lt, $gte, $in) and logical operators ($and, $or); Projection and Sorting: Field inclusion and exclusion, sort(), limit(), skip();Updates: updateOne(), updateMany(), and update operators ($set, $inc, $push); Deletion: deleteOne(), deleteMany(); Basic Indexing: Purpose of indexes and their impact on query performance.); Database backup and restore.

Learning Outcomes: Describe NoSQL data models, the CAP theorem, BASE properties (Knowledge); Create collections and perform CRUD, aggregation, indexing and data modelling operations in MongoDB (Skills); Apply the appropriate relational or NoSQL data model for a given application (Competency)

Unit 7: Advanced Topics **5 LHs**

Database security: access control, authentication and authorization(privilege and roles); Introduction to SQL injection and encryption; database performance tuning: query tuning, index selection and execution plans; concept of parallel database and its advantages and disadvantages;

Distributed Databases: Introduction, Characteristics, Advantages, Disadvantages, Components, Distributed Database Design, Fragmentation (horizontal and vertical fragmentation), Replication and Allocation; Introduction to big data (volume, velocity, variety) and its link to NoSQL

Learning Outcomes: Describe database security, performance tuning, distributed databases, data warehousing, data mining and big data (Knowledge); Identify security threats and use indexes and execution plans to improve query performance (Skills); Appreciate emerging database technologies and their role in data-driven organizations (Competency)

Pedagogical Strategies

- Lectures with live demonstrations of SQL and MongoDB commands
- Hands-on laboratory sessions with guided and open-ended exercises
- Problem-based learning using real-world case studies
- Group discussions and peer review of ER diagrams and database designs
- Multimedia presentations to visualize database architecture and transactions
- Mini project with iterative design and feedback cycles
- Continuous assessment and formative feedback

Mode of delivery

- Lecture sessions (Theory)
- Demonstration
- Laboratory work (Practical)
- Mini project

Internal assessment methods and types (40%)

Assessment Type	Weightage	Details
Class participation & attendance	10%	Contribution to discussions, engagement in class activities
Quizzes/short tests	15%	Periodic quizzes to assess comprehension
Practical/Project	20%	Lab sessions (SQL and NoSQL) and mini project
Mid-term examination	25%	Written test
Pre-board examination	30%	Comprehensive written test covering all units

External assessment methods and types (60%)

Out of the total 60% allocated for final assessment, 40% will be assigned to the written/board examination to evaluate students' abilities in remembering, understanding, applying, analyzing, evaluating, and creating. The remaining 20% will be assigned to the final practical examination to assess hands-on SQL and NoSQL skills and competency.

Mapping course: Learning outcomes and program learning

Course Learning Objective (CLO) Dimensions	Knowledge (K)	Skills (S)	Competence (C)
Total Learning	40%	35%	25%

Laboratory work

Students are required to complete hands-on exercises for every unit. The following practical tasks are prescribed:

- **Unit 1 Lab**: Install a DBMS; explore its tools and client interface; identify database components
- **Unit 2 Lab**: Draw ER and EER diagrams for case studies using a diagramming tool and convert them to relational schemas. Create tables with primary key, foreign key, CHECK and DEFAULT constraints; test integrity constraint violations
- **Unit 3 Lab**: Normalize sample relations to 3NF/BCNF and implement the normalized tables
- **Unit 4 Lab**: Write DDL and DML statements; queries with WHERE, ORDER BY, DISTINCT, aggregates, GROUP BY and HAVING; set operations; joins and nested subqueries; views; COMMIT, ROLLBACK and SAVEPOINT; GRANT and REVOKE; stored procedures, functions, triggers and indexes
- **Unit 5 Lab**: Demonstrate transactions with COMMIT and ROLLBACK, lock behaviour and isolation levels using two concurrent sessions; backup and restore a database
- **Unit 6 Lab**: Install MongoDB; create databases and collections; insertOne, insertMany and mongoimport; find() with query operators, projection, sort and limit; update and delete operations; embedded vs referenced data models and $lookup; aggregation pipeline; single and compound indexes with explain(); replica set or Atlas demonstration; mongodump and mongorestore
- **Mini project**: Individually or in groups of at most 3 students, design a database for a real-world problem using an ER diagram, normalized relational tables with SQL, and a MongoDB component with a short justification; submit an original report of at least 4 pages containing the project description, entities, ER diagram and normalization steps, sample SQL and MongoDB queries with outputs, and a conclusion

Suggested Readings

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Pearson Education.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Database system concepts* (6th ed.). McGraw-Hill.

Ramakrishnan, R., & Gehrke, J. (2003). *Database management systems* (3rd ed.). McGraw-Hill.

Ullman, J. D., & Widom, J. (2008). *A first course in database systems* (3rd ed.). Pearson Education.

Sadalage, P. J., & Fowler, M. (2012). *NoSQL distilled: A brief guide to the emerging world of polyglot persistence*. Addison-Wesley.

Chodorow, K. (2013). *MongoDB: The definitive guide* (2nd ed.). O'Reilly Media.

Fowler, A. (2015). *NoSQL for dummies*. John Wiley & Sons.

Özsu, M. T., & Valduriez, P. (2020). *Principles of distributed database systems* (4th ed.). Springer.

MongoDB, Inc. (n.d.). *MongoDB documentation*. https://www.mongodb.com/docs/

TRIBHUVAN UNIVERSITY
FACULTY OF MANAGEMENT
Office of the Dean
Model Question 2026

Full Marks: 60
Pass Marks: 30
Credit Hour: 3
Time: 3 Hrs

BIM / Third Semester / ITM 203: Database Management System

Candidates are required to answer all the questions in their own words as far as practicable. Figures in brackets indicate full marks.

Group "A"
Concept Based Brief Answer Questions

Answer the following questions: (Attempt All) ***[5× 2 = 10]***

1) Mention any two advantages of the database approach over the file-processing system.
2) What is a weak entity type? Give an example.
3) Define functional dependency with an example.
4) Write the difference between the WHERE clause and the HAVING clause in SQL.
5) Write any two advantages of using parallel databases.

Group "B"
Problem Solving / Short Answer Questions

Answer any SIX Questions: ***[6 × 5 = 30]***

6) Explain the three-schema architecture of a DBMS with a neat diagram. How does it support logical and physical data independence?
7) What is the difference between specialization and generalization in the EER model? Explain disjoint/overlapping and total/partial constraints with suitable examples.
8) Consider a relation R(A, B, C, D, E) with the set of functional dependencies F = {A → B, BC → D, D → E}. Compute the closure of {A, C} And find the candidate key of R.
9) How does the two-phase locking protocol ensures serializability? Explain.
10) Explain the techniques used to improve database performance.
11) Explain the ACID properties of a transaction with a suitable example.
12) Explain replication and sharding in NoSQL databases with example.

Group "C"
Comprehensive / Extended Problem / Case / Situation Analysis / Long Answer Questions

Answer any TWO Questions: ***[2 × 10 = 20]***

13) A university wants to develop a database to manage students, courses, instructors and departments. Students have an ID, name, address and one or more phone numbers. Courses have a code, title and credit hours. Instructors have an ID and name. A department has a

name and location. A student can enroll in many courses and a course can have many students; each enrollment records the semester and grade. Each course is taught by one instructor, and each instructor works in exactly one department.

a) Draw an ER diagram showing entities, attributes, keys, relationships and cardinalities.

b) Convert the ER diagram into relational tables by applying the mapping rules.

14) Consider the Following relation

Employee (EmpID, Name, Salary, DeptID) and Department (DeptID, DeptName, Location), where DeptID in Employee is a foreign key referencing Department.

Write SQL queries to:

a) display the names of employees whose salary is greater than 50000 in ascending order of name.

b) display the DeptID and number of employees of each department that has more than 3 employees.

c) display each employee's name along with the name of his/her department using a join.

d) display the employees who receive the highest salary using a subquery.

15) An online store keeps its customer orders in a MongoDB collection named orders. A sample document is: { id: 1, customer: "Ram", city: "Kathmandu", product: "Laptop", qty: 2, price: 85000}.

a) Write a command to insert a new order document into orders.

b) Write a query to display only the customer and product of orders from Kathmandu with qty greater than 10

c) Write a command to update the price of all "Laptop" orders to 90000

d) Write a command to delete the product “Typewriter”

STT 204: Business Statistics

BITM 3rd Semester

Credits: *3*
Lecture Hours: *48*

Nature of the course: Theory + Practical

Course Description

This course introduces fundamental statistical concepts and analytical techniques essential for effective decision-making in business and management. Students will learn how to organize, summarize, and visualize raw data using descriptive statistics, tables, and graphs. The curriculum also covers bivariate analysis through simple linear correlation and regression, along with foundational probability theory, discrete and continuous probability distributions, and sampling methods. Finally, the course introduces statistical inference—specifically point and interval estimation and large-sample hypothesis testing—with a strong practical emphasis on solving real-world business problems using Microsoft Excel.

Course Objectives

Upon successful completion of this course, students will be able to:

• Organize and Visualize Data: Construct frequency distributions, charts, and diagrams (such as histograms, ogives, and box plots) to effectively summarize and present business data.

• Compute Descriptive Measures: Calculate and interpret measures of central tendency, dispersion, partition values, skewness, and kurtosis to analyze the shape and spread of datasets.

• Analyze Bivariate Relationships: Measure the strength and direction of relationships between two variables using correlation coefficients and build predictive models through simple linear regression.

• Apply Probability Concepts: Utilize basic laws of probability and theoretical distributions (Binomial, Poisson, and Normal) to model uncertainty and assess risk in business scenarios.

• Design Sampling Plans: Distinguish between different sampling techniques, understand sampling errors, and apply the Central Limit Theorem to sampling distributions.

• Perform Statistical Inference: Construct confidence intervals for population parameters, determine appropriate sample sizes, and conduct large-sample hypothesis tests (Z-tests) to make data-driven managerial decisions.

• Utilize Spreadsheet Software: Apply Microsoft Excel functions and data analysis tools to perform statistical calculations, generate visualizations, and interpret outputs accurately.

Course Details:

Unit 1: Describing Data using Graphs and Tables **4 LHs**

Uses and scope of statistics in business and management, Frequency distribution, Stem-and-leaf plots, Diagrams (Simple bar diagram, Sub-divided bar diagram, Multiple bar diagram, and Pie-chart) and graphical presentation of frequency distribution – Histogram, Ogive curve, Problems using Excel.

Learning Outcomes: Explain the uses and scope of statistics in business and management, organize raw data into frequency distributions and stem-and-leaf plots, construct bar diagrams (simple, sub-divided, multiple) and pie charts for categorical data, plot histograms and ogive curves to visualize frequency distributions, utilize Microsoft Excel to generate tables, charts, and graphs for business data analysis.

Unit 2: Describing Data Using Numerical Measures **9 LHs**

Measures of central tendency (Mean, Median and Mode), Partition values (Quartiles, Deciles and Percentiles); Measures of variation (Range, inter quartile Range, Quartile deviations, Standard deviation); Variance and Coefficient of Variation; Measurement of skewness (Karl Pearson coefficient of skewness and Bowley coefficient of skewness); Measurement of kurtosis (Percentile coefficient of kurtosis); Five number summery, Box- and -Whisker plot, Problems using Excel.

Learning Outcomes: Calculate Mean, Median, Mode, and partition values (Quartiles, Deciles, Percentiles); Measure variability using Range, IQR, Quartile Deviation, Variance, Standard Deviation, and Coefficient of Variation; Evaluate symmetry and peakedness using skewness (Karl Pearson, Bowley) and kurtosis (Percentile coefficient); Construct Five-Number Summaries, draw Box-and-Whisker plots, and perform these analyses in Excel.

Unit 3: Simple Linear Correlation Analysis **5 LHs**

Introduction, Scatter plot, Karl Pearson's correlation coefficient, Properties of correlation coefficient, Coefficient of determination, Test of significance of sample correlation coefficient using probable error, Spearman's rank correlation coefficient, Merits and limitation of rank correlation. Problems using Excel.

Learning Outcomes: Define linear correlation and use scatter plots to identify the direction and strength of relationships between two variables. Compute & Interpret Karl Pearson's correlation coefficient, apply its properties (such as limits between -1 and +1), and explain variance using the coefficient of determination (r^2). Evaluate the statistical significance of r using Probable Error and compute Spearman's rank correlation for ordinal or ranked data. Apply Software for generating scatter plots and calculating correlation coefficients efficiently using Microsoft Excel.

Unit 4: Simple Linear Regression Analysis **4 LHs**

Introduction, Simple linear regression models, Assumptions of linear regression model, Line of best fit, Linear regression model by least-squares method, Interpretation of regression coefficients, Properties of regression coefficient, Uses of regression analysis, Difference between correlation analysis and regression analysis, Problems using Excel.

Learning Outcomes: Define simple linear regression and identify dependent and independent variables. Formulate the simple linear regression model and explain its underlying assumptions. Calculate the line of best fit using the Ordinary Least Squares (OLS) method. Interpret the slope and y-intercept coefficients in a real-world context. Apply the mathematical properties of regression coefficients. Execute regression analysis and solve practical problems using Microsoft Excel.

Unit 5: Probability **4 LHs**

Introduction, Sample space and events, Basic concept of probability, Laws of probability, Conditional probability, Baye's theorem, Problems using Excel.

Learning Outcomes: Define random experiments, sample spaces, and different types of events. Calculate basic probabilities using classical and empirical approaches. Apply the addition and multiplication laws of probability to compound events. Compute conditional probabilities and test for event independence. Solve real-world problems by updating probabilities using Bayes' theorem. Use Excel functions (like combination, permutation and probability) to automate probability calculations.

Unit 6: Probability Distributions **6 LHs**

Introduction, Discrete probability distribution (Binomial distribution and Poisson distribution), Continuous probability distribution (Normal distribution), Problems using Excel.

Learning Outcomes: Differentiate between discrete and continuous probability distributions and identify their real-world applications. Calculate probabilities, means, and variances for discrete events using Binomial and Poisson distributions. Determine probabilities and z-scores for continuous data using the Normal distribution. Solve probability distribution problems efficiently using Microsoft Excel functions.

Unit 7: Sampling Theory **3 LHs**

Introduction, Population and sample, Objectives of sampling, Sampling techniques, Sampling and non-sampling errors, Standard error, Concept of central limit theorem.

Learning Outcomes: Distinguish between a population and a sample, and explain the core objectives of sampling. Select and apply appropriate probability and non-probability sampling techniques. Differentiate between sampling and non-sampling errors and identify ways to minimize them. Interpret standard error to assess the precision of sample estimates. Apply the Central Limit Theorem to describe the behavior of sampling distributions.

Unit 8: Estimation **6 LHs**

Introduction, Properties of good estimator (Consistency, Unbiasedness, Efficiency and Sufficiency), Point and interval estimates, Level of confidence, Confidence interval estimates for mean and proportion, Determination of sample size for mean and proportion, Problems using Excel.

Learning Outcomes: Define statistical estimation and distinguish between point and interval estimates. Evaluate estimators based on unbiasedness, consistency, efficiency, and sufficiency. Interpret confidence levels and construct confidence intervals for population means and proportions. Calculate the minimum sample size required for estimating means and proportions. Solve practical estimation problems using Excel.

Unit 9: Introduction to Hypothesis Testing **7 LHs**

Introduction, steps of hypothesis testing, Level of significance, Critical region, one tailed test and two tailed test, Hypothesis testing using critical value and p-value approaches, Test of significance for large samples (Z-test): Test of significance of a single mean and difference between two means, Test of significance of a single proportion and difference between two proportions, Problems using Excel.

Learning Outcomes: Define null and alternative hypotheses and select between one-tailed and two-tailed tests. Interpret the level of significance and critical regions to make decisions using both critical value and p-value approaches. Execute large-sample tests of significance for single and two-sample means and proportions. Compute test statistics, p-values, and interpret hypothesis test results using Microsoft Excel.

Pedagogical strategies

- Lectures: Lessons with clear examples and step-by-step problem solving.
- Lab Work: Hands-on practice building circuits with guidance.
- Real-World Problem Solving: Learning by solving actual business problems.
- Group Activities: Team discussions and role-playing based on business and management situations.
- Visual Presentations: Using slides, videos, and graphics to make ideas easier to see and understand.
- Regular Check-ins: Weekly quizzes and practice exercises to track students' progress.

Mode of delivery

- Lectures: Learning the main ideas and theory.
- Demonstrations: Watching how things are done.
- Tutorials: Small group discussions and extra help.
- Lab work: Hands-on practice.

Internal assessment methods and types

Assessment types	Weightage	Details
Class participant & attendance	10%	Contribution to discussions, engagement in class activities
Quizzes/Short tests	15%	Periodic quizzes to asses comprehension
Practical/ Project	20%	Lab sessions and mini project
Mid-term examination	25%	Written test
Pre-board examination	30%	Comprehensive written test covering all units

External Assessment Method and Types

The final assessment makes up 60% of the total grade and is divided into two parts:

- 40% for the written/board exam: This tests how well students can remember, understand, apply, analyze, evaluate, and create information.
- 20% for practical and competency work: This assesses hands-on skills and real-world abilities.

Mapping Course: Learning Outcomes and Program

Course Learning Objective (CLO) Dimensions	Knowledge (K)	Skill (S)	Competence (C)
Total Learning	30%	35%	35%

Laboratory Works (16 hours)

This lab uses Microsoft Excel for business data analysis, covering:

- Data visualization: Creating tables, charts, and graphs.
- Basic statistics: Calculating averages, spread, skewness, and kurtosis.
- Advanced analysis: Running correlation, regression, probability, estimation, and hypothesis tests.

Suggested Reading

Davis, G. and B. Pecar. *Business Statistics using EXCEL*, Oxford University Press.

Levine, D.V., D.F. Stephan and K. A. Szabat. *Statistics for managers using MS excel***,** Pearson.

Linda, H. and F. John. *Applied Statistics for Business and Management using Microsoft Excel*, Springer.

Trevor***,*** *W. Applied Statistics: Methods and Excel-Based Application*, JUTA Academics.