Tribhuvan University

Faculty of Management

Office of the Dean

Course detail of
BITM (Bachelor of Information Technology Management) 2nd Semester

Effective from the Admission Batch 2025 AD Onwards

ITM 151: Digital Logic	3 Cr. Hrs.
ITM 152: Object Oriented Programming in Java	3 Cr. Hrs.
ITM 153: Discrete Structure	3 Cr. Hrs.
ENG 153: Business Communications	3 Cr. Hrs.
MGT 153: Organizational Behavior & Human Resource Management	3 Cr. Hrs.

May 2026

ITM 152: Object Oriented Programming in Java

BITM 2nd

Nature of the Course: Theory + Practical

Credits: *3*
Lecture Hours: *48(Th) +16(Pr)*

Course Description

This course provides a comprehensive introduction to Object Oriented Programming (OOP) using Java as the primary implementation language. Building upon the foundations of Structured Programming in C, students will transition to the object-oriented paradigm and develop proficiency in Java's type system, class hierarchy, inheritance, interfaces, exception handling, file handling, generics, and essential Java library classes.

The course bridges fundamental programming knowledge with modern software design principles, equipping students to design modular, reusable, and robust applications. Equal emphasis is placed on theoretical understanding and hands on laboratory work, preparing students for advanced software development courses and industry roles.

Course Objectives

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

- Understand the history, features, and architecture of Java (JVM, JRE, JDK) and the object-oriented paradigm.
- Demonstrate knowledge of Java fundamentals including data types, variables, operators, and control statements.
- Define and use classes, objects, constructors, methods, and access control to implement encapsulation.
- Apply inheritance, method overriding, abstract classes, and interfaces to build extensible class hierarchies.
- Utilize arrays, strings, and essential Java library classes for solving computational problems.
- Handle exceptions effectively using Java's built-in and user-defined exception mechanisms.
- Implement file handling operations (text, binary, serialization) for data persistence.
- Apply generic programming concepts and use the Java Collections Framework.
- Write multi-threaded programs and manage concurrency issues.
- Develop complete Java applications using good software engineering practices

Course Details

Unit 1: Introduction to Java 5 LHs

The Creation of Java; Java's lineage and its influence on modern programming; Java's impact on the Internet; Java Buzzwords (Simple, Object-Oriented, Robust, Multithreaded, Architecture Neutral, Interpreted, High Performance, Distributed, Dynamic); Java Virtual Machine (JVM), Java Runtime Environment (JRE), and Java Development Kit (JDK) architecture and roles; The Bytecode; compilation model vs. interpretation model; Platform independence; Writing, Compiling, and Running Simple Java Programs using Command Line and IDE; Object-Oriented Programming Paradigm: Overview of OOP principles (Encapsulation, Inheritance, Polymorphism, Abstraction);

Using Command Line Arguments; Reading Input with Scanner; Writing Output with System.out

Learning Outcomes: Describe the history, features, and architecture of Java including JVM, JRE, and JDK (Knowledge); Write, compile, and execute simple Java programs using command line tools and IDE (Skills); Appreciate the significance of platform independence and OOP in Java application development (Competency).

Unit 2: Fundamental Programming Structures **6 LHs**

Writing Comments; Primitive Data Types (byte, short, int, long, float, double, char, boolean); Variables and Constants; Type Conversion and Casting; Automatic Type Promotion Rules; Operators: Arithmetic, Bitwise, Relational, Logical (short-circuit), Assignment, Conditional (?:), sizeof equivalent; Operator Precedence and Associativity; Using Parentheses; Control Statements: if, if else, nested if else, else if ladder, switch; Iteration Statements: while, do while, for, enhanced for each loop, nested loops; Jump Statements: break, continue, return; Working with Big Numbers (BigInteger, BigDecimal overview); Declaration, Initialization, and Processing of One Dimensional and Multidimensional Arrya; Passing Arrays to Methods.

Learning Outcomes: Explain Java's data types, operators, and control structures with proper syntax (Knowledge); Write Java programs using decision making, iteration, and array processing (Skills); Apply fundamental programming structures to develop logical and efficient Java programs (Competency).

Unit 3: Classes and Objects **8 LHs**

Object Oriented Principles: Encapsulation, Abstraction, Inheritance, Polymorphism; Defining Classes: fields, methods, and the general form of a class; Creating Objects and Accessing Class Members; Assigning Object Reference Variables; Methods: Adding Methods, Returning Values, Method Parameters and Return Types; Method Overloading; Constructors: Default and Parameterized; Constructor Overloading; the this Keyword; Variable Length Arguments; Access Control: public, private, protected, default; JavaBeans convention; static Fields and Methods; final Variables, Methods, and Classes; The Object Class; Nested and Inner Classes: static nested, inner, local, and anonymous classes; Garbage Collection and Finalization; Object Lifecycle; Creating and Using Packages; import statements; CLASSPATH.

Learning Outcomes: Describe the concepts of classes, objects, constructors, methods, and access control in Java (Knowledge); Implement well-structured Java classes with proper encapsulation, method overloading, and package organization (Skills); Apply OOP principles to design modular and reusable Java components (Competency).

Unit 4: Inheritance and Interfaces **7 LHs**

Inheritance Basics: extends keyword, member access, IS-A relationship; Using super: calling superclass constructors and overridden methods; Method Overriding: rules, @Override annotation, covariant return types; Dynamic Method Dispatch: runtime polymorphism; why overridden methods matter; Abstract Classes: abstract methods and the template method concept; inheritance types; Using final with Inheritance(preventing overriding and subclassing); Interfaces: defining, implementing, and extending interfaces;

Default and Static Methods in Interfaces (Java 8+); multiple interface inheritance; Functional Interfaces and the @FunctionalInterface annotation; Sealed Classes overview; Liskov Substitution Principle.

Learning Outcomes: Explain inheritance, method overriding, abstract classes, and interfaces in Java (Knowledge); Implement class hierarchies using inheritance, polymorphism, and interfaces effectively (Skills); Apply inheritance and interface-based design to build extensible and loosely coupled Java applications (Competency).

Unit 5: Exception Handling **4 LHs**

Exception Handling Fundamentals; Exception Hierarchy: Throwable, Error, Exception, Runtime Exception; Checked vs. Unchecked Exceptions; understanding uncaught exceptions and stack traces; Using try, catch, and finally blocks; multiple catch clauses; nested try statements; Multi catch (Java 7+); try with resources for automatic resource management; throw and throws keywords; exception chaining (initCause, getCause); Java's Built in Exceptions overview; Creating Custom Exception Subclasses; best practices for exception design

Learning Outcomes: Describe the exception hierarchy and types of exceptions in Java (Knowledge); Implement robust exception handling using try catch finally, custom exceptions, and try with resources (Skills); Apply exception handling to build fault tolerant programs that handle errors gracefully (Competency).

Unit 6: Generics and Collections **5 LHs**

Importance of Generic Programming; defining Generic Classes and Methods; type parameters; Bounded Wildcards (extends/super); type erasure and heap pollution; Restrictions and Limitations of Generics; Inheritance Rules with Generics; The Java Collections Framework: List, Set, Map, Queue, and Deque interfaces; Iterators and the Iterable interface; enhanced for loop with collections; Type Wrappers: Character, Boolean, numeric wrappers; Autoboxing and Unboxing; Autoboxing in expressions; autoboxing with Boolean and Character; preventing errors.

Learning Outcomes: Describe generic programming concepts, type parameters, wildcards, and the Collections Framework (Knowledge); Implement generic classes and methods (Skills); Apply generics and collections to write type-safe, reusable, and efficient data management code (Competency).

Unit 7: Multithreaded Programming **5 LHs**

The Java Thread Model: threads vs. processes; benefits of multithreading; Thread Priorities; synchronization; messaging between threads; The Thread Class and the Runnable Interface; the main thread; Creating Threads: implementing Runnable; extending Thread class; Creating Multiple Threads; using isAlive() and join(); Thread Priorities and scheduling; Synchronization: synchronized methods and synchronized statements.

Learning Outcomes: Explain the Java thread model, thread lifecycle, synchronization, and enumeration concepts (Knowledge); Create and manage threads using Runnable and Thread; apply synchronized mechanisms to prevent race conditions (Skills); Apply multithreading techniques to develop concurrent, efficient, and thread safe Java applications (Competency).

Unit 8: Essential Java Classes **3 LHs**

String Class: constructors, length, special operators (concatenation, conversion, toString()); Character extraction: charAt() and getChars(); String comparison methods; Modifying Strings: substring(), concat(), replace(), trim(); StringBuffer and StringBuilder: mutable strings, performance considerations; String Handling Library Functions overview; Math Class: commonly used mathematical methods; Primitive Type Wrappers: Character, Boolean, Integer, Double methods and conversions; Introduction to Lambda Expressions and Stream API (overview for modern Java awareness).

Learning Outcomes: Describe the functionality of essential Java classes including String, StringBuilder, Math, and wrapper classes (Knowledge); Use string manipulation methods, math functions, and wrapper class utilities to solve common programming problems (Skills); Apply essential Java library classes to write cleaner, more expressive, and efficient programs (Competency).

Unit 9: File Handling **5 LHs**

Introduction to Files; File class: creating, deleting, listing files and directories; File Modes and I/O Exceptions; AutoCloseable, Closeable, and Flushable Interfaces; Byte Streams: InputStream, OutputStream, FileInputStream, FileOutputStream; PrintStream, DataOutputStream, DataInputStream; RandomAccessFile for random access; Character Streams: Reader, Writer; FileReader, FileWriter, PrintWriter, BufferedReader, BufferedWriter; The Console Class; reading from standard input and writing formatted output; NIO.2 API (java.nio.file): Path, Files, Paths modern file operations; Sequential and Random Access to Files; Serialization: Serializable interface, ObjectOutputStream, ObjectInputStream; I/O Classes and Interfaces review.

Learning Outcomes: Describe Java file I/O concepts including file types, stream hierarchies, serialization, and NIO.2 (Knowledge); Implement file handling operations opening, reading, writing, appending, closing for both text and binary files using byte and character streams (Skills); Apply file management and serialization techniques to develop programs with persistent data storage and retrieval capabilities (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
- Multimedia presentations to visualize OOP and concurrency concepts
- Mini project with iterative design and feedback cycles
- Guest lectures from software industry 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 programming exercises for every unit. The following practical tasks are prescribed:

- Unit 1 Lab: Write and execute basic Java programs; explore JDK tools (javac, java); use Scanner for input
- Unit 2 Lab: Programs using operators, type casting, control statements, and arrays
- Unit 3 Lab: Design and implement classes with constructors, method overloading, static members, and packages
- Unit 4 Lab: Build class hierarchies using inheritance; demonstrate polymorphism and interface implementation
- Unit 5 Lab: Programs with try-catch-finally, custom exceptions, and try-with-resources
- Unit 6 Lab: Implement generic classes/methods; use ArrayList; and sorting with Comparator
- Unit 7 Lab: Create multithreaded programs; demonstrate synchronization and deadlock prevention; use Enumerations
- Unit 8 Lab: String processing programs; use StringBuilder, Math class, and wrapper classes
- Unit 9 Lab: File I/O programs — text file read/write, binary file handling, random access, and object serialization; develop a record-keeping system with search and manipulation features

Suggested Readings

Horstmann, C. S. (2022). *Core Java: Volume I—fundamentals* (12th ed.). Pearson Education.

Schildt, H. (2022). *Java: The complete reference* (12th ed.). McGraw Hill.

Bloch, J. (2018). *Effective Java* (3rd ed.). Addison-Wesley.

Deitel, P., & Deitel, H. (2020). *Java: How to program* (11th ed.). Pearson.

Loy, M., Niemeyer, P., & Leuck, D. (2020). *Learning Java: An introduction to real-world programming with Java*. O'Reilly Media.

Eckel, B. (2006). *Thinking in Java* (4th ed.). Prentice Hall.

Oracle Corporation. (2023). *Java SE 21 documentation*. https://docs.oracle.com/en/java/javase/21/

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 / Second Semester / ITM152: Object Oriented Programming in Java

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 purpose of this keyword in Java?
2) Which interfaces and classes are used to serialize and deserialize an object?
3) List the difference between StringBuffer, and StringBuilder in Java.
4) List the Key Aspects of Java's Impact on the Internet.
5) What is the difference between throw and throws in Java exception handling?

Group "B"
Problem Solving / Short Answer Questions

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

6) Describe the architecture of JVM, JRE, and JDK.
7) Write a Java program to demonstrate type conversion and casting between different data types.
8) Explain method overloading and variable length arguments (varargs) in Java.
9) Explain the concept of Sealed classes with example
10) Write a Java program to demonstrate the use of nested try blocks.
11) What are wrapper classes in Java? Describe the concepts of autoboxing and unboxing.
12) Write a Java program using StringBuffer to demonstrate string manipulation by performing operations such as appending, inserting, and deleting characters.

Group "C"

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

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

13) Write a Java program to demonstrate file handling by performing read and write operations on a file.

14) Define polymorphism in Java and explain its types with suitable examples.

15) Write a Java program that simulates a ticket booking system in which multiple threads attempt to book tickets from a limited number of available seats. The program should include a class named TicketCounter with a method for booking tickets, and the total number of tickets should be initialized as10 tickets. Create multiple threads using the Runnable interface that try to book tickets at the same time. Apply proper synchronization to ensure that tickets are not overbooked. The program should display which thread successfully books tickets and the number of tickets remaining after each transaction. Use the Thread class to create and manage the threads.

ITM 151: Digital Logic

2nd Semester

Nature of the Course: Theory + Practical

Credits: *3*
Lecture Hours: *48(Th) +16(Pr)*

Course Description

This course provides a comprehensive introduction to Digital Logic, covering the fundamental principles, analysis techniques, and design methodologies of digital circuits and systems. Beginning with number systems and digital codes, the course progresses through combinational logic design, sequential logic design, register and counter architectures, sequential machine design, and the characteristics of digital integrated circuits. Students develop a solid understanding of how digital systems are conceived, analyzed, and constructed from basic logic gates up to complex integrated circuit devices.

The course bridges foundational IT knowledge with the hardware principles that underpin all modern computing technology, equipping students of the Bachelor of Information Technology and Management program to understand and reason about the digital systems they will develop, manage, and deploy throughout their careers. Equal emphasis is placed on theoretical understanding and hands-on laboratory work, preparing students for advanced courses in computer organization, microprocessors, and embedded systems.

Course Objectives

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

Knowledge

- Describe the fundamental principles of digital versus analog signals and explain the structure and application of digital number systems including binary, octal, hexadecimal, BCD, Gray code, and alphanumeric codes.
- Explain the operation of basic, universal, and exclusive logic gates and apply De'Morgan's laws and Boolean algebra to analyze and simplify digital expressions.
- Identify the design procedures and functional characteristics of combinational logic circuits including adders, subtractors, multiplexers, demultiplexers, encoders, decoders, and comparators.
- Describe the operation of latches, flip-flops (SR, D, T, JK), registers, and counters, and explain their role in sequential logic systems.
- Recall the principles of synchronous sequential machine design including state diagrams, transition tables, excitation maps, and circuit realization.
- Identify the electrical characteristics and circuit configurations of digital integrated circuit .

Skills

- Convert numbers between decimal, binary, octal, hexadecimal, BCD, Excess-3, and Gray code representations and perform binary arithmetic using 1's and 2's complement.
- Design and simplify combinational logic circuits using Karnaugh maps, Boolean algebra, and sum-of-product / product-of-sum methods.
- Implement and analyze combinational circuits such as adders, multiplexers, encoders, and decoders from problem specifications to gate-level schematics.
- Design synchronous and asynchronous sequential circuits including flip-flop conversions, registers, mod-n counters, and sequence detectors.
- Construct and test digital circuits in the laboratory using ICs, breadboards, and measuring instruments.

Competency

- Apply digital logic design methodologies to solve problems involving data encoding, computation, control, and communication relevant to information technology systems.
- Appreciate the significance of digital logic as the foundational building block of all modern computing and information technology systems, connecting hardware principles to software and system-level IT practice.

Course Details

Unit 1: Introduction **6 LHs**

Digital versus analog signals; Digital integrated circuits (Advantage, Characteristics and Level of Integration); Digital system applications; Digital codes and conversions; Decimal, binary, octal and hexadecimal codes; BCD code; Excess-3 code; Gray code; Alphanumeric codes: ASCII code and EBCDIC code; Binary addition and subtraction; 1's complement and 2's complement; Signed number representation.

Learning Outcomes: Differentiate between digital and analog signals and justify the advantages of digital integrated circuits , integration levels; Convert numbers accurately among decimal, binary, octal, hexadecimal, BCD, Excess-3, and Gray code representations, and perform binary arithmetic operations including addition, subtraction, and 1's and 2's complement methods ; Represent signed integers using sign-magnitude and complement notations and explain how alphanumeric codes such as ASCII and EBCDIC enable text data to be encoded in digital systems; Recognize the pervasive role of number systems and digital codes as the foundational language underlying all data storage, processing, and communication in modern computing infrastructure

Unit 2: Logic Gates **3 LHs**

Basic gates, Derived gates and Universal gates (NAND, NOR) and their equivalents; Realization of different gates using Universal gates; Applications of logic gates

Learning Outcomes: Identify and compare the types, symbols, truth tables, and Boolean expressions of basic gates (AND, OR, NOT), derived gates (NAND, NOR, XOR, XNOR), and classify NAND and NOR as universal gates capable of realizing any logic

function ; Construct equivalent gate circuits using only NAND or NOR gates; Analyze practical logic gate circuits to determine their function, and design simple gate networks that implement specified logical operations in real digital systems

Unit 3: Boolean Algebra and Logic Simplification **8 LHs**

Boolean algebra and its laws; Demorgan's Theorem; Simplification of Boolean expressions; Minterms and maxterms; Sum-of-product (SOP) and product-of-sum (POS) methods; Truth tables and Karnaugh map (Up to Four variables); Don't care conditions; Boolean Function simplification using Quine-McCluskey Method/Tabulation Method.

Learning Outcomes: State the postulates, theorems, and De'Morgan's laws of Boolean algebra and translate any logic function between truth table, SOP canonical form, POS canonical form, minterm, and maxterm representations ; Simplify Boolean expressions systematically using algebraic manipulation, Karnaugh maps for up to four variables with don't care conditions, and the Quine-McCluskey tabular method to obtain minimal SOP and POS forms ; Evaluate competing simplification strategies to select the minimized expression that reduces gate count and propagation delay, thereby lowering cost and improving performance in practical digital circuit implementations.

Unit 4: Combinational Logic Circuits **12 LHs**

Design procedures; Half-adder and full-adder design; Half-subtractor and full-subtractor design; Binary adder design; BCD adder design; Encoder designs; Decoder designs; Multiplexers design; Boolean function implementation using multiplexer; Demultiplexer design; Seven-segment decoder; Magnitude comparators; Code Converter; Programmable Logic Devices (PLA and PAL design).

Learning Outcomes: Explain the structured design procedure for combinational circuits and describe the internal operation of half-adder, full-adder, half-subtractor, full-subtractor, BCD adder, binary adder, encoder, decoder, multiplexer, demultiplexer, seven-segment decoder, magnitude comparator, and code converter circuits ; Design combinational logic circuits by systematically deriving truth tables, writing minimized Boolean expressions, and producing gate-level schematics for each specified functional block ; Construct and troubleshoot combinational circuit implementations, and evaluate their suitability for arithmetic processing, data routing, and code conversion tasks in IT hardware systems.

Unit 5: Sequential Logic Circuits **6 LHs**

Latches and flip-flops: SR, D, T and JK; Characteristics Table; Excitation tables and characteristic equations of all flip flops; Master-slave flip-flops; Flip-flop applications

Learning Outcomes: Distinguish between latches and edge-triggered flip-flops, and compare the operation, characteristic equations, excitation tables, and triggering behavior of SR, D, T, and JK flip-flop types, including master-slave configurations ; Design

simple flip-flop application circuits and analyze their role as the elementary memory and state-holding elements that underpin all sequential digital systems.

Unit 6: Registers and Counters **10 LHs**

Register; Shift Register (SISO, SIPO, PISO, PIPO and Bidirectional Registers); Data transfer timing diagrams; Asynchronous counters (Up, Down, Mod-N and UP/Down asynchronous counters); Synchronous counters (Up, down, Mod-N, UP/Down synchronous counters); Shift Register Counter; Register and counter applications.

Learning Outcomes: Classify and compare the four shift register transfer modes (SISO, SIPO, PISO, PIPO) and bidirectional operation, and differentiate between asynchronous (ripple) and synchronous counter architectures in terms of propagation delay and modulus flexibility ; Design shift registers and Mod-N asynchronous and synchronous up/down counter circuits by selecting appropriate flip-flop types, deriving timing diagrams, and realizing the circuit from given count-sequence specifications ; Apply register and counter circuits to solve practical digital system problems including serial data buffering, parallel-to-serial conversion, frequency division, ring and Johnson counter sequences, and event timing in computing hardware.

Unit 7: Sequential Machine Designs **3 LHs**

Sequential Machine design procedures: state diagram (introduction and example only); Transition/ State tables; Reduction of Redundant states; binary assignment tables; Excitation Table and output table; derivation of flip-flop input and Circuit output function, design of a circuit diagram.

Learning Outcomes: Explain the complete synchronous sequential machine design procedure, state reduction using redundant-state elimination, pure binary state assignment, transition and excitation map derivation, and circuit realization. Students are expected to learn the design procedure from the state diagram provided.

Pedagogical strategies

- Lectures with worked circuit design examples and step-by-step problem-solving demonstrations
- Hands-on laboratory sessions with guided and exploratory circuit construction exercises using standard IC families
- Problem-based learning using real-world IT and computing scenarios that connect digital logic concepts to practical hardware and system applications
- Collaborative problem solving through tutorial group work on K-map simplification and sequential machine design
- Multimedia presentations to visualize waveform timing, state machine transitions, and IC internal architectures
- Design assignments with iterative feedback cycles to reinforce the theory-to-implementation workflow

- Continuous formative assessment through quizzes and tutorial exercises aligned with weekly topics

Mode of delivery

- Lecture sessions (Theory)
- Demonstration
- Tutorial sessions
- Laboratory work (Practical)

Internal assessment methods and types (40%)

Assessment Type	Weightage	Details
Class participation & attendance	10%	Engagement in class discussions and laboratory activities
Quizzes / Short Tests	15%	Periodic quizzes to assess theoretical comprehension
Practical / Project	20%	Laboratory sessions and circuit design assignments
Mid-term examination	25%	Written test covering Units 1–4
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 circuit design and laboratory skills.

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 circuit construction and verification exercises for each unit. The following practical tasks are prescribed (45 Practical Hours):

- Lab 1: Verify the truth tables of basic gates (AND, OR, NOT), universal gates (NAND, NOR), and exclusive gates (XOR, XNOR) using ICs on a breadboard or Digital Circuit Trainer Board
- Lab 2: Demonstrate De 'Morgan's theorem experimentally using NAND and NOR gates; verify equivalent circuit implementations using ICs on a breadboard or Digital Circuit Trainer Board
- Lab 3: Design and construct half-adder, full-adder, half-subtractor, and full-subtractor using ICs on a breadboard or Digital Circuit Trainer Board
- Lab 4: Design and construct encoder (4-to-2, 8-to-3) and decoder (2-to-4, 3-to-8, BCD decoder) and verify their operations using ICs on a breadboard or Digital Circuit Trainer Board

- Lab 5: Design and construct multiplexers (4-to-1, 8-to-1) and demultiplexer (1-to-4, 1-to-8) circuits using ICs on a breadboard or Digital Circuit Trainer Board

- Lab 6: Design and construct any three-code converter using ICs on a breadboard or Digital Circuit Trainer Board
- Lab 7: Build and analyze SR latch, gated SR latch, and T flip-flop circuits; observe triggering behavior using ICs on a breadboard or Digital Circuit Trainer Board or available simulator tools.
- Lab 8: Construct D flip-flop and JK flip-flop circuits; implement a master-slave JK flip-flop and observe its behavior using ICs on a breadboard or Digital Circuit Trainer Board or available simulator tools.
- Lab 9: Design and observe SISO, SIPO shift register using ICs on a breadboard or Digital Circuit Trainer Board or available simulator tools.
- Lab 10: Design and observe PISO, and PIPO shift register using ICs on a breadboard or Digital Circuit Trainer Board or available simulator tools.
- Lab 11: Design and observe bidirectional shift register configurations using ICs on a breadboard or Digital Circuit Trainer Board or available simulator tools.
- Lab 12: Design and test a 3-bit and a 4-bit asynchronous (ripple) counter; observe modulus and up/down counting behavior on LEDs using ICs on a breadboard or Digital Circuit Trainer Board or available simulator tools.
- Lab 13: Design and test a 3-bit and a 4-bit synchronous counter; observe modulus and up/down counting behavior on LEDs using ICs on a breadboard or Digital Circuit Trainer Board or available simulator tools.

Suggested Readings

Floyd, T. L. (2015). *Digital fundamentals*. Pearson Education.

Mano, M. M. (1995). *Digital logic and computer design*(Latest Edition). Prentice Hall.

Leach, D. P., Malvino, A. P., & Saha, G. (2012). *Digital principles and applications* (8th ed.). Tata McGraw-Hill Education.

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 / Second Semester / ITM151: DIGITAL LOGIC

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) Simplify xy+x'z+yz to minimum number of literals using Boolean Algebra.
2) Implement OR and AND gate using NAND gates only.
3) We can't have both input 1 in a basic flip flop constructed with NOR gates Justify.
4) In what case is it economical to use PLA than PAL? Explain
5) List out the major characteristics of IC.

Group "B"
Problem Solving / Short Answer Questions

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

6) Differentiate between T and JK flip-flop along with its Truth Table and logic diagram.
7) Implement the function F(W,X,Y,Z)=∑(0,2,4,6,10,12,14) using multiplexer.
8) Design a 3-bit synchronous binary counter using any flip-flop.
9) What is Shift Register? You are provide with a bit sequence 1101, and you are asked to extract those bits after right shifting. Which shift register would you use? Discuss with necessary block diagrams and timing diagrams.
10) Design 8X1 multiplexer
11) Design a 3-bit synchronous counter.
12) Mention the design procedure of a combinational circuit.

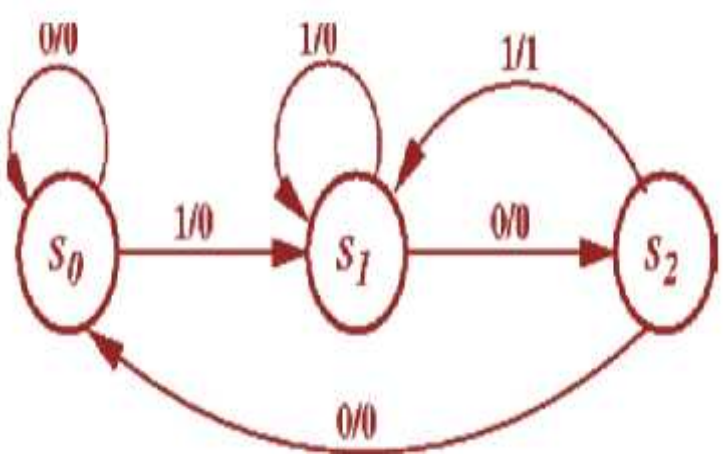
Group "C"

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

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

13) Design 4X16 decoder using suitable number of 2X4 along with its truth table and explain its operating mechanism.

14) Design a circuit diagram for a given state diagram following standard design procedure



15) Design a MOD-16 synchronous Up/Down Counter.

ENG 153: Business Communication

BITM 2nd

Credit: 3
Lecture Hours: 48

Course Description

The course aims to develop students' professional communication competence and leadership skills necessary for success in the contemporary digital and corporate workplace. It integrates business writing, oral communication, intercultural collaboration, and digital professionalism. It provides an understanding of communication practices and prepares students for their assignments in the corporate world. Students learn how to plan, compose, and deliver effective messages across media and organizational contexts while developing interpersonal, persuasive, and ethical communication skills. Emphasis is placed on job readiness, teamwork, report writing, business presentations, and communication technology applications that enhance employability and organizational performance.

Course Objectives

- Understand the role of communication in professional, managerial, and intercultural contexts.

- Compose effective business documents—emails, memos, proposals, and reports—using standard formats and professional tone.

- Demonstrate oral communication competence in meetings, interviews, and presentations.

- Apply digital and visual tools for information design and workplace communication.

- Collaborate effectively in diverse teams through ethical and empathetic communication.

- Integrate strategic communication skills that enhance employability and entrepreneurial innovation.

Course Details

Unit 1: Fundamentals of Business Communication **8 LHs**

a) Purpose and scope of business communication
b) Barriers and strategies for effective communication
c) Listening, feedback, and interpersonal skills
d) Communication ethics in multicultural workplaces

Learning Outcomes: Identify key elements, functions, and ethics of business communication (knowledge); apply strategies to overcome communication barriers through clarity, empathy, and persuasion (skills); and demonstrate professional and ethical communication behavior in simulated workplace interactions (competency).

Unit 2: Business Writing in the Digital Workplace **8 LHs**

a) Email, memos, and instant business messages

b) Writing for collaboration platforms (Teams, Slack, etc.)
c) Documenting information using AI and digital tools
d) Etiquette and security in e-communication

Learning Outcomes: Understand structure, tone, and format of digital business messages (knowledge); draft clear and concise professional correspondence using digital media (skills); and exhibit responsible, efficient, and secure digital communication practices (competency).

Unit 3 : Business Reports and Proposals **8 LHs**

a) Types of business proposals and reports
b) Planning, researching, and organizing content
c) Visual design and data presentation
d) Executive summaries and recommendations

Learning Outcomes: Understand components and formats of formal business reports and proposals (knowledge); plan, research, and draft data-supported documents with effective visuals (skills); and present evidence-based recommendations to solve organizational problems (competency).

Unit 4: Oral and Interpersonal Communication **8 LHs**

a) Structuring and delivering business presentations
b) Conducting and participating in meetings
c) Interview communication strategies
d) Verbal and nonverbal elements of persuasion

Learning Outcomes: Recognize principles of effective speaking and listening (knowledge); prepare and deliver professional presentations and participate in meetings (skills); and demonstrate confidence, clarity, and leadership in the formal business settings (competency).

Unit 5: Cross-Cultural and Interpersonal Communication **8 LHs**

a) Communication across cultures
b) Managing diversity and inclusion
c) Conflict resolution and negotiation
d) Social intelligence and empathy in professional life

Learning Outcomes: Identify dimensions of cultural communication differences (knowledge); employ empathy and diplomacy in cross-cultural communication (skills); and use intercultural competence and teamwork skills for local and global business contexts (competency).

Unit 6: Employment Communication and Professional Branding **8 LHs**

a) Writing résumés, cover letters, and portfolios
b) Virtual job interviews and LinkedIn networking
c) Personal branding and professional ethics
d) Reflective career communication

Learning Outcomes: Understand modern hiring processes and branding strategies (knowledge); prepare job-related documents and perform effectively in interviews (skills); and build and communicate a professional identity aligned with market demands (competency).

Pedagogical Strategies

- Interactive lectures and problem-based learning

- Business simulations, role-playing, and mock interviews

- Group presentations, collaborative assignments, and case study analysis

- Use of digital tools

- Peer feedback and workshops on presentation and digital communication

- Guest lectures from professors, professionals, and entrepreneurs

Mode of Delivery

- In-person and blended learning

- Digital collaboration using online platforms

- simulations aligned with professional scenarios.

Internal Assessment Methods

Type	Weightage	Details
Class participation & attendance	10%	Engagement and contribution
Quizzes / short tests	15%	Application-based quizzes
Assignments / case study reports	20%	Business writing & presentation tasks
Mid-term examination	25%	Analytical written test
Pre-board examination	30%	Comprehensive applied writing test

External Assessment: Final written examination assessing remembering, understanding, applying, analyzing, and creating business communication outputs.

Mapping of Learning Outcomes

Dimensions	Weightage
Knowledge (K)	30%

Skills (S)	35%
Competency (C)	35%

Suggested Readings

Cardon, P. W. (2023). *Business communication: Developing leaders for a networked world (5th ed.)*. McGraw-Hill Education.

Guffey, M. E., Loewy, D., & Almonte, R. (2022). *Essentials of business communication (12th ed.)*. Cengage Learning.

Reardon, T. H., Sage, T., & Brizee, A. (2022). *Open technical communication (3rd ed.).* LibreTexts.

Locker, K. O., & Kaczmarek, S. K. (2014). *Business communication: Building critical skills (6th ed.)*. McGraw-Hill Education.

World Bank. (2021). *Developing skills for the modern labor market in South Asia: Trends and policy lessons*. Washington, DC: World Bank Group.

ITM 153: Discrete Structure

Semester: II

Nature of the Course: Theory + Practical

***Credits**: 3*
***Lecture Hours**: 48(Th) + 16(Pr)*

Course Description

This course introduces students to the concepts of mathematical structures that are fundamentally countable. It provides the toolkit to handle discrete data, things that cannot be divided. Students learn to know about the mathematical background of the pure computing like logic, number theory, set theory, counting techniques, proof techniques, graphs and tree.

Course Objectives

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

- Design truth tables and logical connectives in business contracts
- Design Venn diagram to show market segmentation
- Design implicative business rules and identifying logical fallacies in marketing
- Use counting principles for product bundling
- Use binomial theorem in quality control and discrete probability in financial forecasting
- Use graph theories in modeling supply chain codes, delivery cost reduction, identifying critical paths in project timelines, analyzing influence within a market

Course Details

Unit 1: Logic and Proofs **10 LHs**

Propositional logic, Logical Operators (AND, OR, NOT, IMPLICATION with its variants, BICONDITIONAL), Laws of logical equivalences, Translating English sentences, Predicate and Quantifiers, Nested and Order in quantifiers, Translating English sentences using quantifiers, Rules of inferences for propositional logic, Valid arguments in propositional logic, Fallacies, Rules of inferences for quantified statements, Valid arguments in quantified statements, Methods of Proving Theorems (Direct Proof, Indirect Proof, Proof by Contradiction, Vacuous and Trivial Proof, Proof of Equivalence, Exhaustive Proof, Proof by Cases, Existence Proof, Uniqueness Proof, Counter Example), Mistakes in Proof.

Learning Outcome: Use connectives like AND ($\wedge$), OR ($\vee$), NOT ($\neg$), and implication ($\rightarrow$) to evaluate the truth value of complex statements (Apply). Describe the different rules of inferences for propositional logic and quantified statements (Understand). Identify the techniques of proof (Remember).

Unit 2: Set Theory and Functions **2 LHs**

Sets, Ways to describe the sets, Venn diagram, Subset, Size of a set, Power set, Cartesian product, Set operations, Set identities, Computer representations of set, Functions, One to one function, Onto function, Bijection, Inverse functions, Composition of functions, Graph of functions, Floor functions, Ceiling functions, Sequence and Summations, Boolean matrices, Meet and Join operation on Boolean matrices.

Learning Outcome: Describe the sets and Venn diagram (Understand). Use the different representation techniques of set (Apply). Identify the types of functions (Remember).

Unit 3: Number Theory **5 LHs**

The division algorithm, Modular Arithmetic, Representation of Integers, Algorithms for Integer Operations, Primes, Greatest Common Divisors, Linear Congruences, The Chinese Remainder Theorem, Computer Arithmetic with Large Integers.

Learning Outcome: Explain the Chinese Remainder Theorem (Understand). Perform the modular arithmetic operations (Apply). Identify the prime (Remember).

Unit 4: Mathematical Induction and Recursion **4 LHs**

Introduction to Mathematical Induction, Proof by Mathematical Induction, Strong Induction, Well Ordering Property, Recursively Defined Functions and Sets, Structural Induction, Generalized Induction, Recursive Algorithms, Proving the Correctness of Recursive Algorithms.

Learning Outcome: Describe the steps of mathematical induction (Understand). Use it to proof the inequalities (Apply). Writing of the recursive algorithms (Remember).

Unit 5: Basics of Counting **4 LHs**

Sum Rule, Product Rule, Principle of Inclusion – Exclusion, The Pigeonhole Principle (Generalized as well), Permutations and Combinations, Binomial Theorem, Pascal's Identity and Triangle, Generalized Permutations and Combinations, Permutations with Repetitions, Combinations with Repetitions, Permutations with Indistinguishable Objects

Learning Outcome: Explain the generalized Pigeonhole principle (Understand). Generates permutations and combinations (Apply). What is Pascal's triangle? (Remember).

Unit 6: Discrete Probability **4 LHs**

Introduction, Finite Probability, Probabilities of Complements and Unions of Events, Assigning Probabilities, Conditional Probability, Independence, Random Variable, The Birthday Problem, Expected Value.

Learning Outcome: What is probability (Understand)? What are the uses of Conditional Probability (Apply)? What is random variable (Remember)?

Unit 7: Advanced Counting Techniques **5 LHs**

Recurrence Relations, Modeling with Recurrence Relations, Solving Linear Homogeneous Recurrence Relations with Constant Coefficients (*Without Proving the Theorem*), the Degree of Two Case (*Two Distinct or Equal Characteristic Roots*), the General Case (the Degree may be Greater than Two, where the Characteristic Equation has Distinct Roots or Repeated Roots).

Learning Outcome: Describe the recurrence solution (Understand). Solve the recurrence relations (Apply). Why do we need to solve it (Remember)?

Unit 8: Relations **3 LHs**

Relation and its Properties, *n – ary* Relations, Representing Relations (using Matrix and Digraphs), Closure of Relations (Reflexive, Symmetric, Transitive), Warshall's

Algorithm to Compute the Transitive Closure of a Relation, Equivalence Relations, Equivalence Classes, Partial Ordering.

Learning Outcome: Describe the properties of relation (Understand). Represent the relations using matrix and directed graph (Apply). What is Partial Ordering (Remember)?

Unit 9: Graph Theory **8 LHs**

Graph Models, Types of Graphs (*Simple Graph, Multigraph, Pseduograph, Directed Graph, Null Graph, Bipartite Graph*), Graph Terminologies (*Adjacent Vertices, Degree of a Vertex, Isolated Vertex, Pendant Vertex*), Handshaking Theorem, Representation of Graphs (*Adjacency List, Adjacency Matrix, Incidence Matrix*), Graph Isomorphism, Graph Connectivity, Euler and Hamilton Path, Necessary and Sufficient Conditions for Euler and Hamilton Path and Circuits (*Without Proof*), Shortest Path Algorithm (*Dijkstra's Algorithm*), Planar Graph, Graph Coloring

Learning Outcome: Explain the different types of graphs (Understand). Know the theorem related to graph (Apply). What is the use of Dijkstra's Algorithm (Remember)?

Unit 10: Trees **3 LHs**

Introduction to Trees, Rooted Tree, Terminologies of a Tree (Parent, Child, Sibling, Ancestors, Descendants, Leaf, Internal Nodes), *M – ary* Tree, Binary Search Tree, Decision Tree, Prefix Codes, Tree Traversal, Spanning Tree, Minimum Spanning Tree, Kruskal's Algorithm.

Learning Outcome: Describe the terminologies of a tree (Understand). Find the MST (Apply). What is *M – ary* tree (Remember)?

Pedagogical Strategies

- Lectures with demonstration
- Hands-on lab sessions
- Problem-based learning
- Guest lectures from tech industry experts
- Continuous assessment and feedback
- Multimedia presentations to visualize concepts
- Mini project

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 Works (16 hours)

The laboratory work includes writing programs for implementing the concepts of logic gates, boolean matrix operations, floor and ceiling value of given positive as well as negative real numbers, set theory, number theory and graph theory using a programming language like C.

Suggested Reading

Kenneth H. Rosen (2019), *Discrete Mathematics and Its Applications*, Eighth Edition, McGraw Hill Education.

Susanna S. Epp (2020), *Discrete Mathematics with Applications*, Fifth Edition