	DATES
PRACTICALS	26/12/24 - 31/12/24
BCU341+FLU344	2/1/25
BSU343	3/1/25
SPECIALISATION	6/1/25
CSE303 - DSC++	7/1/25
CSE302 - PYTHON	8/1/25
CSE304 - DBMS	9/1/25
ECE306	10/1/25
MAT301	13/1/25

MAT301(MATHEMATICS)

MI: Basic Statistics:

Measures of Central Tendency
- Mean, Median, Mode

Moments, Skewness, and Kurtosis
- Calculation of moments
- Skewness and its interpretation
- Kurtosis and its interpretation

Correlation and Regression
- Pearson's correlation coefficient
- Simple linear regression
- Multiple regression

Rank Correlation
- Spearman's rank correlation

Curve Fitting
- Method of least squares
- Fitting of straight lines
- Fitting of second degree parabolas
- Fitting of more general curves

MII: Basic Probability and expectation:

- **Discrete and Continuous Random Variables**
  - Properties of discrete random variables
  - Properties of continuous random variables
- **Dependent and Independent Random Variables**
  - Concept of dependence and independence
- **Probability Spaces**
  - Sample spaces and events
  - Axioms of probability
- **Conditional Probability**
  - Definition and applications
- **Sums of Independent Random Variables**
  - Distribution of sums
- **Expectation of Discrete Random Variables**
  - Expected value and its properties
- **Probability Distributions for Discrete and Continuous Variables**
  - **Discrete distributions:**
    - Binomial distribution
    - Poisson distribution
  - **Continuous distributions:**
    - Normal distribution
  - **Evaluation of Statistical Parameters**
    - Mean, variance, standard deviation, skewness, kurtosis

MIII: Test of significance for Small and large samples:

Large Sample Tests
- Test for a single proportion
- Test for difference of proportions
- Test for a single mean
- Test for difference of means
- Test for difference of standard deviations

Small Sample Tests
- Test for single mean
- Test for difference of means
- Test for correlation coefficients
- Test for ratio of variances

Chi-Square Test
- Goodness of fit
- Test for independence of attributes

MIV: Numerical Methods:

- **Solution of Simultaneous Linear Equations**
  - Jacobi's method
  - Gauss-Seidel method
- **Solution of Algebraic and Transcendental Equations**
  - Bisection method
  - Newton-Raphson method
  - Regula-Falsi method
- **Interpolation Methods**
  - Interpolation using finite differences
  - Relation between operators
  - Newton's forward and backward difference formula
  - Interpolation for unequal intervals:
    - Newton's divided difference formula
    - Lagrange's interpolation formula

MV: Numerical Methods:

- **Numerical Differentiation and Integration**
  - Picard's method
  - Trapezoidal rule
  - Simpson's 1/3rd rule
  - Simpson's 3/8 rule
- **Solution of Ordinary Differential Equations**
  - Taylor's series method
  - Euler's method
  - Modified Euler's method
  - Runge-Kutta method (Fourth order)
  - Milne's method
  - Adam's predictor-corrector method

CSE302(PYTHON)

Module I: Introduction to Python

- **Introduction to Python**
    - **History of Python**
    - **Features of Python Programming**
    - **Applications of Python**
    - **Use of Python in various domains**
    - **Installation and running Python on Windows/Linux**
- **Basic Syntax and Structure**
    - **Keywords and Identifiers**
    - **Statements and Comments**
    - **Python Variables**
    - **Python Data Types**
    - **Python Type Conversion**
    - **Python I/O and Import**
    - **Python Operators**
    - **Python Namespace**
- **Control Flow and Looping**
    - **If-else Statements**
    - **Python for Loop**
    - **while Loop**
    - **Break and Continue Statements**
- **Data Structures and Manipulation**
    - **String Manipulation**
    - **Lists, Tuples, and Dictionaries**
    - **Pass Statement**
    - **Looping Techniques (e.g., List comprehensions, etc.)**
- **Functions in Python**
    - **Functions and Function Arguments**
    - **Recursion**
    - **Anonymous Functions (Lambda functions)**
    - **Python Global, Local, and Nonlocal variables**

Module II: (OOP) in Python

- **Python Modules and Packages**
    - **Python Modules**
    - **Python Packages**
- **File Operations and Directory Handling**
    - **File Operations (Reading, Writing, etc.)**
    - **Python Directory Operations**
- **Exception Handling**
    - **Python Exception Handling**
    - **User-defined Exceptions**

- **Object-Oriented Programming (OOP) Concepts**
  - **Classes and Objects in Python**
  - **Inheritance**
    - **Single Inheritance**
    - **Multiple Inheritance**
  - **Operator Overloading**

Module III: Regular Expressions, CGI, and Database

- **Regular Expressions in Python**
  - **match() function**
  - **search() function**
  - **Matching vs. Searching**
  - **Modifiers in Regular Expressions**
  - **Patterns in Regular Expressions**
- **CGI (Common Gateway Interface) Programming**
  - **Introduction to CGI**
  - **CGI Architecture**
  - **CGI Environment Variables**
  - **GET/POST Methods in CGI**
  - **Working with Cookies**
  - **File Uploads using CGI**
- **Database Programming in Python**
  - **Introduction to Databases**
  - **Database Connections in Python**
  - **Executing SQL Queries**
  - **Transactions in Databases**
  - **Handling Database Errors**

Module IV: GUI Programming and Web Frameworks

- **GUI Programming with Tkinter**
  - **Tkinter Overview**
  - **Tkinter Widgets (e.g., Buttons, Labels, Entry fields, etc.)**
  - **Standard Attributes of Widgets**
- **CGI Programming**
  - **Introduction to CGI Programming (basic concepts)**
- **Web Frameworks: Django**
  - **Introduction to Django Web Framework**
  - **Application Lifecycle in Django**
  - **Creating a Django Project**
  - **Django Admin Interface**
  - **Creating Views in Django**
  - **URL Mapping in Django**
  - **Template System in Django**
  - **Creating Database Models in Django**
  - **Interfacing Databases with Django (e.g., PostgreSQL)**
  - **Page Redirection in Django**

- **Form Processing in Django**

CSE322(PYTHON LAB)

1. Write a python program to demonstrate working of lists.: **(2 Hours)**
2. Write a python program to demonstrate working of tuples.: **(2 Hours)**
3. Write a python program to demonstrate working of dictionaries and conditional statements: **(2 Hours)**
4. Write a python program to demonstrate working of Inheritance and other OOP concepts.: **(2 Hours)**
5. Write a python program to demonstrate regular expressions like match function, search function, pattern search function.: **(2 Hours)**
6. Write a python program for reading data from CSV file.: **(2 Hours)**
7. Write a python program for writing data in CSV file.: **(2 Hours)**
8. Write a python program for reading data from text file.: **(2 Hours)**
9. Write a python program for writing data from text file.: **(01 Hour)**
10. Write a python program for image analysis using open CV.: **(01 Hour)**
11. Write a program to demonstrate connection with postgresql: **(01 Hour)**
12. Develop a dynamic website using Django framework and postgresql as backend.: **(1 Hour)**

CSE202(DS C++)

Module I: Introduction to C++

- **C++ Programming Concepts**
  - Review of C Programming
  - Input and Output in C++
  - Functions in C++
    - Value Parameters
    - Reference Parameters
    - Parameter Passing
    - Function Overloading
    - Function Templates
  - Exceptions in C++
    - Throwing Exceptions
    - Handling Exceptions
  - Arrays in C++
  - Pointers in C++
    - `new` and `delete` operators
  - Classes and Objects in C++
    - Access Specifiers
    - Friend Functions
    - Constructors and Destructors
    - Operator Overloading
    - Class Templates
  - Object-Oriented Concepts
    - Inheritance
    - Polymorphism
- **Basic Concepts and Data Structures**
  - Data Objects and Structures
  - Algorithm Specification
    - Introduction to Algorithms
    - Recursive Algorithms
  - Data Abstraction
  - Performance Analysis
    - Time Complexity
    - Space Complexity
  - Asymptotic Notation
    - Big O Notation
    - Omega Notation
    - Theta Notation
  - Complexity Analysis Examples
  - Introduction to Linear and Non-Linear Data Structures

Module II: Introduction to Data Structures (DS)

- **Arrays and Matrices**
  - Representation of Single and Two-Dimensional Arrays
  - Sparse Matrices: Array and Linked Representations

- **Linear List ADT**
  - Array Representation
  - Linked Representation
- **Linked Lists**
  - Singly Linked Lists
    - Operations: Insertion, Deletion
  - Circularly Linked Lists
    - Operations for Circularly Linked Lists
  - Doubly Linked Lists
    - Operations: Insertion, Deletion
- **Stacks**
  - Stack ADT Definition
  - Array and Linked Implementations
  - Applications
    - Infix to Postfix Conversion
    - Postfix Expression Evaluation
    - Recursion Implementation
- **Queues**
  - Queue ADT Definition
  - Array and Linked Implementations
  - Circular Queues
    - Insertion and Deletion Operations

Module III: Trees

- **Introduction to Trees**
  - Tree Definition and Terminology
  - Binary Trees
    - Definition
    - Properties of Binary Trees
    - Binary Tree ADT
    - Representation of Binary Trees: Array and Linked Representations
    - Binary Tree Traversals
  - Threaded Binary Trees
  - Priority Queues
    - Definition and Applications
    - Max Priority Queue ADT Implementation
    - Max Heap Definition
    - Insertion into a Max Heap
    - Deletion from a Max Heap
- **Minimum Spanning Tree**
  - Prim's Algorithm
  - Kruskal's Algorithm
- **Shortest Path Algorithms**

Module IV: Searching & Sorting

- **Searching Algorithms**
  - Linear Search
  - Binary Search

- ○ Hashing
  - ■ Introduction to Hashing
  - ■ Hash Tables
  - ■ Hash Functions
  - ■ Overflow Handling
- ○ Comparison of Searching Methods
- **Sorting Algorithms**
  - ○ Insertion Sort
  - ○ Selection Sort
  - ○ Radix Sort
  - ○ Quick Sort
  - ○ Heap Sort
  - ○ Merge Sort
  - ○ Comparison of Sorting Methods

Module V: Graphs

- **Introduction to Graphs**
  - ○ Definitions and Terminology
  - ○ Applications of Graphs
  - ○ Properties of Graphs
  - ○ Graph ADT Definition
  - ○ Graph Representations
    - ■ Adjacency Matrix
    - ■ Adjacency Lists
  - ○ Graph Search Methods
    - ■ Depth-First Search (DFS)
    - ■ Breadth-First Search (BFS)
  - ○ Complexity Analysis of Graph Algorithms
- **Search Trees**
  - ○ Binary Search Tree (BST) ADT
    - ■ Definition
    - ■ Operations: Searching, Insertion, Deletion
  - ○ Balanced Search Trees
    - ■ AVL Trees (Definition and Examples)
    - ■ B-Trees (Definition and Examples)
    - ■ Red-Black Trees (Definition and Examples)
  - ○ Comparison of Search Trees

CSE222(DS C++ LAB)

1 Write a C++ programs to implement recursive and non recursive i) Linear search ii) Binary search
2 Write a C++ programs to implement i) Bubble sort ii) Selection sort iii) quick sort iv) insertion sort
3 Write a C++ programs to implement the following using an array.
(a) Stack ADT b) Queue ADT
4 Write a C++ programs to implement list ADT to perform following operations
(a) Insert an element into a list.
(b) Delete an element from list
(c) Search for a key element in list
(d) count number of nodes in list

5 Write C++ programs to implement the following using a singly linked list. Stack ADT b) Queue ADT
6 Write C++ programs to implement the deque (double ended queue) ADT using a doubly linked list and an array.
7 Write a C++ program to perform the following operations:
(a) Insert an element into a binary search tree.
(b) Delete an element from a binary search tree.
(c) Search for a key element in a binary search tree.
8 Write C++ programs for implementing the following sorting methods:Merge sort b) Heap sort
9 Write C++ programs that use recursive functions to traverse the given binary tree in
a) Preorder b) in order c) post order
10 Write a C++ program to perform the following operations a) Insertion into a B-tree b) Deletion from a B-tree

CSE304(DBMS)

Module I: Introduction to DBMS

- **Concept and Goals of DBMS**
  - Definition of DBMS
  - Goals of DBMS (e.g., data independence, consistency, security, etc.)
- **Database Languages**
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Data Control Language (DCL)
  - Transaction Control Language (TCL)
- **Database Users**
  - End users
  - Database administrators (DBAs)
  - Application developers
- **Database Abstraction**
  - Levels of abstraction (physical, logical, and view)
  - Data independence (physical and logical)
- **Entity-Relationship (ER) Model**
  - Basic Concepts of ER Model
    - Entities, Attributes, Relationships
    - Keys: Primary, Foreign, Candidate
  - **Relationship Sets**: One-to-one, One-to-many, Many-to-many
  - **Keys**: Superkey, Candidate key, Primary key, Foreign key
  - **Mapping**: Cardinality and participation constraints
  - **Design of ER Model**
  - **Generalization, Aggregation, and Specialization** in ER Models
  - **Transforming ER Diagram into Tables**
- **Other Data Models**
  - Object-Oriented Data Model
  - Network Data Model
  - Relational Data Model

Module II: Relational Data Models

- **Fundamentals of Relational Model**
  - **Domains**: Set of valid values for an attribute
  - **Tuples**: Rows in a table
  - **Attributes**: Columns in a table
  - **Relations**: Tables
  - **Characteristics of Relations**: Uniqueness, order, etc.
- **Keys and Attributes of Relation**
  - Primary Key
  - Candidate Key

- Foreign Key
- Superkey
- **Relational Database Concepts**
  - **Schemas**: Logical design of the database
  - **Integrity Constraints**
    - Entity Integrity
    - Referential Integrity
  - **Intension and Extension**: Schema vs. actual data
  - **Relational Query Languages**
    - SQL (Structured Query Language)
      - **DDL (Data Definition Language)**: CREATE, ALTER, DROP
      - **DML (Data Manipulation Language)**: SELECT, INSERT, UPDATE, DELETE
      - **Integrity Constraints**: Not Null, Unique, Primary Key, Foreign Key
      - **Complex Queries**
      - **Joins**: Inner join, Outer join (left, right, full)
      - **Indexing**: B-tree indexing, Hash indexing
      - **Triggers**: Event-based actions in response to DML operations
- **Relational Algebra and Relational Calculus**
  - **Relational Algebra Operations**:
    - Select, Project, Join, Union, Difference, Intersection
    - Division Operation
  - **Tuple Relational Calculus**

Module III: Database Design

- **Database Design and Normalization**
  - Introduction to Normalization
  - **Normal Forms**: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF
  - **Functional Dependency**
  - **Decomposition**: Breaking relations into smaller, well-structured relations
  - **Dependency Preservation**: Ensuring all functional dependencies are maintained after decomposition
  - **Lossless Join**: Property of decomposition where original relation can be reconstructed without data loss
  - **Null-Valued and Dangling Tuples**: Issues and handling
  - **Multivalued Dependencies**

Module IV: Transaction Processing Concepts

- **Transaction System**
  - Definition of a Transaction
  - ACID Properties (Atomicity, Consistency, Isolation, Durability)
- **Serializability**
  - Testing of Serializability

- ○ Conflict and View Serializable Schedules
  - ● **Recoverability**
    - ○ Recovery from Transaction Failures
    - ○ **Log-Based Recovery**: Write-Ahead Logging (WAL)
    - ○ **Checkpoints**: Savepoints for recovery
  - ● **Concurrency Control Techniques**
    - ○ **Locking Techniques**: Two-Phase Locking (2PL), Deadlock Detection
    - ○ **Timestamping Protocols**: Ensuring serializability through timestamps
    - ○ **Validation-Based Protocol**: Ensuring transactions are serializable without locks
    - ○ **Multiple Granularity**: Locking at different levels of granularity (e.g., row, page, table)
    - ○ **Multiversion Schemes**: Creating multiple versions of data to allow concurrency
    - ○ **Recovery with Concurrent Transactions**

Module V: Relational Database Management Systems (RDBMS)

  - ● **Introduction to RDBMS**
    - ○ Study of RDBMS (e.g., Oracle, PostgreSQL, MySQL)
    - ○ **RDBMS Architecture**
      - ■ Physical Files
      - ■ Memory Structures
      - ■ Background Processes
  - ● **Database Storage Concepts**
    - ○ **Table Spaces**: Logical storage units in a database
    - ○ **Segments, Extents, and Blocks**: Physical storage units
    - ○ **Dedicated Server vs. Multi-threaded Server**
    - ○ **Distributed Database**: Concepts and architecture
  - ● **Introduction to SQL**
    - ○ **ANSI SQL**: Standard SQL syntax
    - ○ **SQL Commands and Operators**
      - ■ LIKE, ANY, ALL, EXISTS
      - ■ **Views**: Creating and using views in SQL
      - ■ **Special Operators**: IN, BETWEEN, IS NULL, etc.
      - ■ **Hierarchical Queries**: Using CONNECT BY (in Oracle), recursive queries
      - ■ **Inline Queries**: Subqueries within SELECT, INSERT, etc.
      - ■ **Flashback Queries**: Retrieving previous versions of data in databases like Oracle

CSE324(DBMS LAB)

1. Using create command design three specific table and the table structure is given below.

Table name- Book

ISBN	TITLE	PUB_YEAR	UNIT_PRICE	AUTHOR_NAME	PUB_NAME
1001	Oracle	2004	399	Arora	phi
1002	Dbms	2004	400	Basu	technical
2001	Dos	2003	250	Sinha	nirali
2002	Adbms	2004	450	Basu	technical
2003	Unix	2000	300	Kapoor	scitech

Table name- Author

AUTHOR_NAME	COUNTRY
Arora	U.S.A.
Kapoor	Canada
Basu	India
Sinha	India

Table name- Publisher

PUB_NAME	PUB_ADD1
Phi	Delhi
Technical	Pune mainmarket
Nirali	Mumbai
Scitech	Chennai

2. Write the SQL query to find the name of all publisher from Book relation.
3. Write the SQL query to display the name of all publisher using distinct clause.
4. Write the SQL query to find the names of author from the author table where the first two characters of names are 'ba'.
5. Write the SQL query to display the title of books published in year 2004.
6. Write the SQL query to display title of books having price between 300 to 400.
7. Write the SQL query to display title of books having price between 300 to 400 using operators.
8. Write the SQL query to display title of books with author_name and country published in year 2004.
9. Write the SQL query to display all title and (unit_price*10) as an attribute from book table using arithmetic expression
10. Write the SQL query to add the new column in all three tables.
11. Study the concept of Views and their utility in DBMS, write the SQL query to design a view.
12. Write the SQL query to make the attribute ISBN as a primary key in Book relation.
13. Write the SQL query to display the all the titles of Books with price and year in descending order.
14. Write the SQL query to study the use of Delete and Drop command in DBMS.
15. Study the concept of Triggers, cursors and stored procedures in DBMS.

ECE306(DELD)

Module I: Fundamentals of Digital Systems

- **Digital Signals and Circuits**
    - Definition of Digital Signals
    - Overview of Digital Circuits
- **Basic Logic Gates**
    - AND, OR, NOT, NAND, NOR, Exclusive-OR (XOR) Operations
    - Examples of IC Gates (Integrated Circuit gates)
- **Boolean Algebra**
    - Basic Boolean Operations
    - Boolean Expressions and Simplifications
    - Properties of Boolean Algebra
- **Number Systems**
    - Binary Number System
    - Signed Binary Numbers
    - Octal and Hexadecimal Numbers
    - Binary Arithmetic
    - One's and Two's Complements
- **Standard Representation for Logic Functions**
    - Boolean Expressions for Logic Functions
- **K-map Representation and Simplification**
    - Karnaugh Map (K-map) for Function Simplification
    - Minimization of Logical Functions Using K-map
    - Don't Care Conditions in K-map
    - Quine-McCluskey (Q-M) Method of Function Realization

Module II: Combinational Digital Circuits

- **Multiplexers and Demultiplexers**
    - Multiplexer (MUX) Overview
    - Demultiplexer (DEMUX) and Decoders
- **Adders and Subtractors**
    - Half Adder and Full Adder
    - Subtractors (Half Subtractor and Full Subtractor)
    - BCD (Binary-Coded Decimal) Arithmetic
- **Other Combinational Circuits**
    - Digital Comparators
    - Parity Checker/Generator
    - Code Converters
    - Priority Encoders
    - Decoders/Drivers for Display Devices (e.g., 7-segment display)

Module III: Sequential Circuits and Systems (8 Hours)

- **Basic Sequential Circuits**
    - 1-bit Memory (Bi-stable Latch)
    - Properties of Bi-stable Latches

- **Flip Flops**
  - Clocked SR Flip Flop
  - JK Flip Flop
  - T Flip Flop
  - D Flip Flop
  - Applications of Flip Flops
- **Shift Registers**
  - Shift Registers Overview
  - Applications of Shift Registers
  - Serial-to-Parallel Converter
  - Parallel-to-Serial Converter
- **Counters**
  - Ring Counter
  - Ripple (Asynchronous) Counters
  - Synchronous Counters
  - Counter Design Using Flip Flops
  - Applications of Counters

Module IV: A/D and D/A Converters & Logic Families

- **Digital-to-Analog (D/A) Converters**
  - Weighted Resistor (D/A) Converter
  - R-2R Ladder D/A Converter
  - Specifications for D/A Converters
  - Examples of D/A Converter ICs
  - Sample and Hold Circuit
- **Analog-to-Digital (A/D) Converters**
  - Parallel Comparator A/D Converter
  - Successive Approximation A/D Converter
  - Counting A/D Converter
  - Specifications for A/D Converters
- **Logic Families**
  - Characteristics of Digital ICs (Integrated Circuits)
  - Digital Logic Families:
    - TTL (Transistor-Transistor Logic)
    - Schottky TTL
    - CMOS (Complementary Metal-Oxide-Semiconductor)
  - Interfacing CMOS and TTL Logic
  - Tri-state Logic (Three-state Logic)
  - ECL (Emitter Coupled Logic)
  - RTL (Resistor-Transistor Logic)
  - DTL (Diode-Transistor Logic)

Module V: Semiconductor Memories and Programmable Logic Devices

- **Memory Organization and Operation**
  - Overview of Memory Organization
  - Expanding Memory Size
- **Classification and Characteristics of Memories**
  - Sequential Memory
  - Read-Only Memory (ROM)

- ○ Read and Write Memory (RAM)
- ○ Content Addressable Memory (CAM)
- ○ Charge-Coupled Device Memory (CCD)
- **Commonly Used Memory Chips**
  - ○ Examples of Memory Chips in Practice
- **Programmable Logic Devices**
  - ○ ROM as a Programmable Logic Device (PLD)
  - ○ Programmable Logic Array (PLA)
  - ○ Programmable Array Logic (PAL)

ECE326(DELD LAB)

1. To verify the truth tables of NOT, OR, AND, NOR, NAND, XOR, XNOR gates. **(2 Hours)**
2. To obtain half adder, full adder using gates and verify their truth tables. **(2 Hours)**
3. To obtain half subtractor, full subtractor using gates and verify their truth tables. **(2 Hours)**
4. To implement control circuit using multiplexer. **(2 Hours)**
5. To convert BCD code into excess 3 code and verify the truth table. **(2 Hours)**
6. To verify the truth tables of RS, D, JK and T flip- flops. **(2 Hours)**
7. To implement and verify 3-bit bi-directional shift register. **(2 Hours)**
8. To design and study asynchronous/ripple counter. **(2 Hours)**
9. To design and study synchronous counter. **(2 Hours)**
10. To design and study a sequence detector. **(2 Hours)**

BSU343(BEHAVIOUR)

Module I: Thinking as a Tool for Problem Solving

- **What is Thinking?**
  - The Mind, Brain, and Behavior: Understanding the Connection
  - Cognitive Processes in Problem Solving
- **Critical Thinking and Learning**
  - **Making Predictions and Reasoning**
    - The Role of Predictions in Problem Solving
    - Types of Reasoning: Deductive, Inductive, Abductive
  - **Memory and Critical Thinking**
    - The Role of Memory in Problem Solving
    - Enhancing Memory for Critical Thinking
  - **Emotions and Critical Thinking**
    - Impact of Emotions on Thinking and Decision Making
    - Emotional Regulation in Critical Thinking
- **Thinking Skills**
  - Types of Thinking: Analytical, Creative, Practical
  - Developing Effective Thinking Skills for Problem Solving

Module II: Hindrances to the Problem-Solving Process

- **Perception**
  - How Perception Shapes Problem Solving
  - Overcoming Perceptual Biases
- **Expression**
  - Barriers in Expressing Ideas and Solutions
  - Improving Communication for Effective Problem Solving
- **Emotion**
  - Emotional Barriers in Problem Solving
  - Strategies for Emotional Intelligence in Decision Making
- **Intellect**
  - Intellectual Barriers: Fixed Mindsets and Cognitive Biases
  - Enhancing Intellectual Flexibility
- **Work Environment**
  - Impact of Work Environment on Problem Solving
  - Creating a Conducive Environment for Creative and Critical Thinking

Module III: Problem Solving

- **Recognizing and Defining a Problem**
  - Identifying the Core Problem
  - Techniques for Problem Definition (5 Whys, Root Cause Analysis)
- **Analyzing the Problem (Potential Causes)**
  - Approaches for Analyzing Underlying Causes
  - Tools for Problem Analysis (Fishbone Diagram, Pareto Analysis)
- **Developing Possible Alternatives**
  - Techniques for Idea Generation

- Lateral Thinking and Creative Solutions
   - **Evaluating Solutions**
      - Criteria for Evaluating Solutions (Feasibility, Effectiveness, Impact)
      - Decision-Making Tools (Decision Matrix, Cost-Benefit Analysis)
   - **Resolution of Problem**
      - Selecting the Best Solution
      - Strategic Implementation and Problem Resolution
   - **Implementation**
      - Steps in Implementing the Chosen Solution
      - Monitoring and Adjusting During Implementation
   - **Barriers to Problem Solving**
      - **Perception**: Recognizing and Overcoming Perceptual Barriers
      - **Expression**: Improving Communication and Idea Sharing
      - **Emotion**: Managing Emotional Responses
      - **Intellect**: Overcoming Cognitive Biases and Mental Blocks
      - **Work Environment**: Addressing Environmental Factors Affecting Problem Solving

Module IV: Plan of Action

   - **Construction of POA (Plan of Action)**
      - Defining Clear Goals and Objectives
      - Outlining Steps and Resources for Action
      - Time Management and Prioritization in POA
   - **Monitoring**
      - Tracking Progress Against Objectives
      - Tools and Techniques for Monitoring (Gantt Charts, Dashboards)
   - **Reviewing and Analyzing the Outcome**
      - Post-Implementation Review and Analysis
      - Identifying Lessons Learned and Opportunities for Improvement

Module V: Creative Thinking

   - **Definition and Meaning of Creativity**
      - Understanding Creativity in Problem Solving
      - The Role of Innovation in Creative Thinking
   - **The Nature of Creative Thinking**
      - **Convergent and Divergent Thinking**
         - Convergent Thinking: Focusing on One Solution
         - Divergent Thinking: Generating Multiple Solutions
      - **Idea Generation and Evaluation**
         - Techniques for Idea Generation (Brainstorming, Mind Mapping)
         - Evaluating Ideas for Practicality and Innovation
      - **Image Generation and Evaluation**
         - Using Mental Imagery to Enhance Creativity
         - Evaluating Visual Ideas and Concepts
      - **Debating**
         - Using Debate to Challenge and Refine Ideas
         - Encouraging Constructive Criticism in Creative Discussions
   - **The Six-Phase Model of Creative Thinking: ICEDIP Model**
      - **I**: Idea Generation

- **C**: Conceptualization
- **E**: Evaluation
- **D**: Development
- **I**: Implementation
- **P**: Promotion and Refinement of Ideas

BCU341(COMMUNICATION)

Module I: Principles of Effective Writing

- **Spellings**
    - 100 Most Misspelled Words in English
    - Common Spelling Mistakes and Tips for Correcting Them
- **Web-Based Writing**
    - Writing for Online Audiences
    - Best Practices for Writing Blogs, Articles, and Web Content
    - Optimizing Content for Search Engines (SEO Basics)
- **Note Taking: Process & Techniques**
    - Effective Note-Taking Strategies
    - Methods: Cornell Method, Mind Mapping, Outlining
    - How to Organize and Review Notes for Better Retention

Module II: Formal Letter Writing

- **Block Format**
    - Structure of a Block-Format Letter
    - Key Components: Date, Address, Salutation, Body, Closing
    - Examples and Practical Applications
- **Types of Letters**
    - Formal Letters: Business, Complaint, Request, Application
    - Informal Letters: Personal, Friendly
    - Letters of Recommendation, Resignation, Invitation
- **E-mail**
    - Structure of Professional Emails (Subject, Salutation, Body, Closing)
    - Tips for Clear and Concise Email Communication
    - Common Email Etiquette
- **Netiquette**
    - Guidelines for Proper Online Behavior
    - Respectful and Professional Communication in Emails, Forums, and Social Media
    - Avoiding Miscommunication in Digital Correspondence

Module III: Business Memos

- **Format & Characteristics**
    - Structure of a Business Memo (Heading, Subject Line, Body, Conclusion)
    - Tone and Language in Memos
    - Use of Memos in Organizational Communication (Internal Communication)

Module IV: Short Stories

- **Stench of Kerosene** by Amrita Pritam
    - Summary and Analysis of the Story
    - Themes: Struggle, Social Issues, Women's Rights
- **A Flowering Tree** by A.K. Ramanujan

- ○ Summary and Analysis of the Story
- ○ Themes: Folklore, Transformation, Cultural Symbolism
- **The Gift of the Magi** by O. Henry
  - ○ Summary and Analysis of the Story
  - ○ Themes: Sacrifice, Love, Irony
- **A Fly in Buttermilk** by James Baldwin
  - ○ Summary and Analysis of the Story
  - ○ Themes: Racism, Identity, Personal Struggles

FLU344

	SPEECH TOPICS	EQUIVALENT ENGLISH MEANING
1	Où faire ses courses ? — Découvrez et dégustez !	Where to shop? — Discover and taste!
2	Au restaurant : Commander et commenter	At the restaurant: Order and comment
3	Inviter et répondre à une invitation	Invite and respond to an invitation
4	Les aliments et Les quantités	Food and Quantities
5	Les commerces et les commerçants	Shops and merchants
6	Demander et dire le prix	Ask and say the price
7	Les services	Services
8	Les moyens de paiement	Means of payment
9	Le pays des gourmands	The country of gourmands
10	Tout le monde s'amuse — Les ados au quotidien	Everyone has fun — Teens in everyday life
11	Décrire une tenue	Describe an outfit
12	Ecrire un message amical	Write a friendly message
13	Les sorties	Going out
14	Situer dans le temps	Situate in time
15	La famille	Family
16	Les vêtements et les accessoires	Clothes and accessories

	GRAMMAR TOPICS	EQUIVALENT ENGLISH MEANING
1	Les articles partitifs	Partitive articles

2	Le pronom « en » (la quantité)	The pronoun "en" (quantity)
3	Très ou beaucoup	Very much
4	La phrase négative	The negative sentence
5	C'est/ Il est	It is/It is
6	L'impératif	The imperative
7	Il faut- les verbes : devoir, pouvoir, savoir, vou	It is necessary- the verbs: devoir, pouvoir, savoir, vou
8	Les adjectifs démonstratifs	Demonstrative adjectives
9	La formation du féminin	Formation of the feminine
10	Le pronom indéfini « On »	The indefinite pronoun "On"
11	Le futur proche	The near future
12	Le passé composé	The past tense
13	Les verbes du premier groupe en « - yer »	The first group verbs in "- yer"
14	Les verbes voir et sortir	The verbs voir and sortir

CSC301(CYBER SECURITY)

Module I: Basic Concepts and Uses of Cryptography

- **Overview of Cryptography**
  - Introduction to Cryptography
  - Importance of Cryptography in Cybersecurity
- **Identifying Resources for Hunting Cyber Threats**
  - Tools and Techniques for Threat Hunting
  - Cyber Threat Intelligence (CTI)
- **Security Flaws in TCP/IP Protocol**
  - Vulnerabilities in TCP/IP Protocol Suite
  - Exploits and Mitigations
- **Cryptography and Its Types**
  - Symmetric vs. Asymmetric Cryptography
  - Key Management and Cryptographic Primitives
- **Cryptographic Algorithms Used in Cybersecurity**
  - RSA (Rivest-Shamir-Adleman) Algorithm
  - AES (Advanced Encryption Standard) Algorithm
  - MD5 (Message Digest Algorithm 5)
- **Web Service Security Concepts**
  - Securing Web Services with Cryptography
  - SSL/TLS for Web Service Security
- **Understanding Attacks on Networks and Hosts**
  - Common Attacks on Networks (e.g., Man-in-the-Middle)
  - Host-Based Attacks (e.g., Malware, Ransomware)
- **Common Endpoint Security Technologies**
  - Antivirus and Anti-malware Solutions
  - Endpoint Detection and Response (EDR)
  - Device Control and Encryption

Module II: Communication Characteristics and Deployment Mechanisms

- **Incident Investigation Models**
  - **Kill Chain Model**: Phases of a Cyber Attack (Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, Action)
  - **Diamond Model**: Framework for Threat Analysis (Adversary, Infrastructure, Capability, Target)
- **Exploit Kits and Threat Actors**
  - Role of Exploit Kits in Cyber Attacks
  - Common Exploit Kits Used by Threat Actors
- **Event Data Normalization and Correlation**
  - Importance of Normalizing Event Data

- Event Correlation to Identify Patterns and Suspicious Activities
- **Common Attack Vectors**
  - Phishing, Malware, Social Engineering, DDoS Attacks, Insider Threats
- **Identifying Malicious Activities and Suspicious Behaviors**
  - Behavioral Analysis to Detect Anomalies
  - Indicators of Compromise (IOCs)
- **Security Incident Investigations**
  - Procedures for Investigating Security Incidents
  - Using Logs and Network Data for Forensic Analysis
- **Playbooks in Security Operations Centers (SOC)**
  - Definition and Importance of Playbooks
  - Automating Incident Response with Playbooks
- **SOC Metrics and Effectiveness**
  - Key Performance Indicators (KPIs) for SOC
  - Measuring SOC Efficiency and Incident Response Times
- **Workflow Management and Automation in SOC**
  - Tools and Techniques to Automate SOC Workflows
  - Integration of Automation for Incident Handling

Module III: Incident Reporting

- **Incident Response Plan (IRP)**
  - Structure and Components of an Incident Response Plan
  - Roles and Responsibilities in Incident Response
- **Computer Security Incident Response Team (CSIRT)**
  - Functionality and Structure of CSIRT
  - Coordination with External Agencies
- **Vocabulary for Event Recording and Incident Sharing (VERIS)**
  - Overview of VERIS Framework
  - Use of VERIS in Incident Reporting
- **Security Incident Format**
  - Common Incident Reporting Formats
  - Best Practices for Incident Documentation
- **Defining the Security Operations Center (SOC)**
  - Role and Function of SOC in Cybersecurity
  - Organizational Structure of a SOC
- **Network Infrastructure and Security Monitoring Tools**
  - Tools for Network Traffic Analysis (e.g., Wireshark, Zeek, Suricata)
  - Network Intrusion Detection Systems (NIDS)
- **Common TCP/IP Attacks**
  - Types of Attacks: IP Spoofing, SYN Flood, DNS Spoofing
  - Mitigation and Detection Strategies
- **Endpoint Security Technologies**
  - Endpoint Protection Platforms (EPP)
  - Advanced Threat Protection (ATP)
- **Incident Analysis in a Threat-Centric SOC**

- ○ Threat-Centric Approach to Security Operations
  - ○ Analysis and Prioritization of Threats
- **Event Correlation and Normalization**
  - ○ Techniques to Normalize and Correlate Security Events for Incident Response

Module IV: Cybersecurity Issues

- **Identifying Common Attack Vectors**
  - ○ Social Engineering, Phishing, Malware Delivery
  - ○ Insiders and Advanced Persistent Threats (APTs)
- **Malicious Activity and Suspicious Behavior Patterns**
  - ○ Identifying Anomalous Traffic Patterns
  - ○ Behavioral Analytics and Anomaly Detection
- **Security Incident Investigations**
  - ○ Investigating Suspicious Activities and Indicators of Compromise (IOCs)
  - ○ Gathering Evidence and Preserving Forensics
- **Using Playbook Models for Security Monitoring**
  - ○ Standardizing Incident Response with Playbook Models
  - ○ Using Playbooks to Improve SOC Efficiency
- **Network Security Monitoring (NSM) Tools**
  - ○ Use of NSM Tools for Analyzing Data Categories (Suricata, Zeek, etc.)
  - ○ Detecting and Analyzing Network Traffic
- **Endpoint Security Analysis**
  - ○ Investigating Endpoints for Malware and Threats
  - ○ Detecting Suspicious Processes and Files on Endpoints
- **Hacker Methodology**
  - ○ Steps Involved in a Cyber Attack (Reconnaissance, Exploitation, etc.)
  - ○ Understanding Attack Methods to Improve Detection
- **Malicious Traffic Hunting**
  - ○ Techniques for Hunting Malicious Traffic in Networks
  - ○ Tools for Traffic Analysis and Threat Detection
- **Event Log Correlation and Packet Captures (PCAPs)**
  - ○ Correlating Event Logs from Different Sources
  - ○ Analyzing Packet Captures for Security Analysis
- **Security Onion Tools Overview**
  - ○ **Elasticsearch, Logstash, Kibana (ELK)**: Log Management and Visualization
  - ○ **Suricata and Zeek**: Network Traffic Analysis
  - ○ **Wazuh**: Host-based IDS and SIEM Integration
  - ○ **TheHive, Cortex**: Incident Response and Analysis
  - ○ **NetworkMiner**: Network Forensics
  - ○ **CyberChef, Stenographer**: Data Analysis and Steganography

Module V: Understanding Cybersecurity Operations Center (SOC)

- **SOC Metrics and Performance Measurement**
  - Key Metrics for SOC Effectiveness (Mean Time to Detect (MTTD), Mean Time to Respond (MTTR))
  - Continuous Improvement of SOC Performance
- **SOC Workflow and Automation**
  - Defining SOC Workflow Processes
  - Using Automation to Improve SOC Response Time
- **Incident Response in the SOC**
  - Response Procedures in the SOC
  - Integrating Automation for Faster Incident Handling
- **Using VERIS for Incident Reporting**
  - Applying VERIS to Document and Share Security Events
- **Alerts and Investigation of Browser-Based Attacks**
  - Identifying and Investigating Browser Exploits
  - Tools for Detecting and Mitigating Browser-Based Attacks
- **Suspicious DNS Activity**
  - Investigating Anomalous DNS Queries and Responses
  - Tools and Techniques for DNS Traffic Analysis
- **Exploring Security Data for Analysis**
  - Gathering and Analyzing Security Data from Various Sources (SIEM, IDS, Firewall Logs)
- **Investigating Advanced Persistent Threats (APTs)**
  - Characteristics of APTs
  - Techniques for Detecting and Mitigating APTs

CSC321(CYBER SECURITY LAB)

P1: To create a simple LAN using a switch and end machines to analyse the performance of configurations and protocols in LAN.
P2: Use Cisco packet tracer to Create two LAN and connect them using a Router. Explain the difference in connecting two LAN using a bridge and a router.
P3: Connect a Server to a Switch and configure it to work as a DHCP server.
P4: Use CISCO packet tracer to connect a wireless LAN.
P5: Create a wireless LAN using Static IP.
P6. Create a wireless LAN and assign DHCP IP using router and wireless access point.
P7: Implement a Subnet from a LAN with Network IP of 192.160.10.0 with 30 machines for each subnet. Create two subnets from within the above-mentioned IP Network ID.