(Applicable to the batch of students admitted in the academic year 2025-26 onwards)

B.Sc. (COMPUTER SCIENCE)
Syllabus (CBCS)
(w.e.f. 2025-2026)

FACULTY OF SCIENCE
SATAVAHANA UNIVERSITY
KARIMNAGAR – 505002

2025

TELANGANA COUNCIL OF HIGHER EDUCATION

(A Statutory body of the Government of Telangana)

REVISED COMMON CORE SYLLABUS OF B.Sc. (COMPUTER SCIENCE)

three year degree under Choice Based Credit System(CBCS) framework for all Universities in Telangana with effect from 2025-2026

(With Learning Outcomes, Unit-wise Syllabus, References, Co-curricular Activities)

(To be implemented for candidates admitted from the 2025-26 Academic Year onwards)

TELANGANA STATE COUNCIL OF HIGHER EDUCATION
FACULTY OF SCIENCE
B.Sc. (Computer Science)
CBCS Pattern with Effect from the Academic Year 2025-26
Structure of Curriculum

Course Title	Hours/Week		Credits		
	Theory	Practical	Theory	Practical	Total
Semester-I					
Programming in C	4	2	4	1	5
Semester-II					
Data Structures using C	4	2	4	1	5
Semester-III					
Database Management Systems	4	2	4	1	5
Semester-IV					
Programming in Java	4	2	4	1	5
Semester-V					
Python Programming	4	2	4	1	5
Semester-VI-ELECTIVE					
(A) Web Technologies	4	2	4	1	5
(B) Computer Networks	4	2	4	1	5
(C) Data Science with Python	4	2	4	1	5
SEC-I SEMESTER-III					
Digital Documentation and Analysis	2		2	-	2
Power BI	2		2	-	2
SEC-II SEMESTER-IV					
Artificial Intelligence	2		2	-	2
Software Engineering	2		2	-	2
SEC-III SEMESTER-V					
Cyber Security	2		2	-	2
Block Chain Technology	2		2	-	2
Generic Elective (other than computer science students) SEMESTER-V					
Emerging Trends in Computer Science	4		4	-	4

COURSE STRUCTURE

B.Sc COMPUTER SCIENCE

Syllabus and Credits Structure under Choice Based Credit System [CBCS] General Degree for the Three Years B.Sc. with Computer Science Undergraduate Programme with effect from 2025-26

SEMESTER	Type of Course	Paper Title	Teaching Hours/ week			Credits	Examination Scheme Max. Marks			
			THEORY	PRACTICAL	TOTAL	Theory/ Practical	IA	EE	Total	Duration of Exam (Hrs)
I	DSC-1	Programming in C	4		4	4	20	80	100	3
	DSC-1 Lab	Programming in C Lab		2	2	1		50	50	2
II	DSC-2	Data Structures using C	4		4	4	20	80	100	3
	DSC-2 Lab	Data Structures using C Lab		2	2	1		50	50	2
III	DSC-3	Database Management Systems	4		4	4	20	80	100	3
	DSC-3 Lab	Database Management Systems Lab		2	2	1		50	50	2
	SEC-1	(a)Digital Documentation and Analysis (b) POWER BI	2		2	2		50	50	2
IV	DSC-4	Programming in JAVA	4		4	4	20	80	100	3
	DSC-4 Lab	Programming in JAVA Lab		2	2	1		50	50	2
	SEC-2	(a)Artificial Intelligence (b) Softwar Engineering	2		2	2		50	50	2
V	DSC-5	Python Programming	4		4	4	20	80	100	3
	DSC-5 Lab	Python Programming Lab		2	2	1		50	50	2
	SEC-3	a) Cyber Security b) Block Chain Technology c) LibreOffice Lab	2		2	2		50	50	2
	Generic Elective	Emerging Trends in Computer Science	4		4	4	20	80	100	3
VI	DSE-1	a) Web Technologies b) Computer Networks c) Data Science with Python	4		4	4	20	80	100	3
	DSE-1 Lab			2	2	1		50	50	2
		Total	34	12	46	40				

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – I
Programming in C
(w.e.f. 2025-2026)

Theory: 4 Hours/Week **Credits: 4**

Internal Marks = 20 **External Marks = 80**

Course Objectives:

Cob1: To introduce the basics of programming languages, focusing on the C programming language, and explore different ways of writing and designing algorithms.

Cob2: To understand the key concepts such as variables, data types, operators, control structures, arrays and strings.

Cob3: To emphasize problem-solving techniques using functions and pointers.

Cob4: To learn the concept of user-defined data types and files.

Course Outcomes:

CO1: Understand program structure and design algorithm.

CO2: Develop basic programs by applying concepts such as control structures, arrays and strings.

CO3: Implement functions, pointers and Dynamic Memory Allocation (DMA).

CO4: Create user-defined data types and implement file operations.

Unit I

Programming Fundamentals: Algorithms and Flow charts, Generation and classification of programming languages, Processes involved in program execution: compilation, interpretation, loading and linking.

Basics of C Programming: Introduction to C programming language, Structure of a C program, C tokens, data types, variables, constants, operators, expression evaluation (precedence, associativity), type conversions in C.

Unit II

Input and Output: Non-formatted and formatted input/output functions, Escape sequences and their usage in I/O. Control Statements - Sequence statements, Selection statements: if, if-else, nested if, switch, conditional operators, Iterative statements: while, do-while and for. Special control statements: goto, break, continue, return, exit.

Arrays and Strings: One-dimensional arrays, Multidimensional arrays and character arrays.

Unit III

Functions: Function definition, declaration and calling mechanisms, types of functions, ctype functions and string functions, call-by-value, call-by-reference. Passing arrays to functions, recursion, inline functions. Scope and lifetime of variables, storage classes.

Pointers: Introduction, address-of operator (&). Uses of pointers, Pointer types: pointers and arrays, pointers and strings, pointer to pointer, array of pointers. Dynamic memory allocation, malloc, calloc and free.

Unit IV

User-Defined Data Types: Structures and unions: Definition, initialization, accessing members, arrays of structures, structures vs. unions, enumeration types (enum).

File handling: Introduction, file operations, file functions: open, close, read and write. Working with text and binary files.

Suggested Books

1 Reema Thareja, 'Programming in C', Oxford University Press, Second Edition, 2016.

2 Kernighan, B.W and Ritchie, D.M, 'The C Programming language', Second Edition, Pearson Education, 2015.

Reference Books

1 Ivor Horton, Beginning C

2 Ashok Kamthane, Programming in C

3 Herbert Schildt, The Complete Reference C

4 Paul Deitel, Harvey Deitel, C How to Program

5 R.S.Bichkar, "Programming with C" University Press, 2024.

6 Byron S. Gottfried, Theory and Problems of Programming with C

7 Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language

8 B. A. Forouzan, R. F. Gilberg, A Structured Programming Approach Using C

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – I
Programming in C Lab

(w.e.f. 2025-2026)

Practical: 2 Hours/Week **Credits: 1** **Marks: 50**

Recommended Software: GCC on Linux, DevC++ or Code Blocks on Windows 10.

External Examination Requirements: Students need to demonstrate the execution of two programs in the external lab exam.

Course Objectives:

Cob1: Develop fundamental programming skills in C by implementing conditional statements, loops, functions, and data structures for problem-solving.

Cob2: Apply key C programming concepts such as arrays, pointers, strings, file handling, and recursion to build efficient algorithms for real-world applications. **Course Outcomes:**

CO1: Demonstrate the ability to write, debug, and execute C programs for solving mathematical and logical problems using control structures, functions, and data structures.

CO2: Apply fundamental C programming concepts, including file handling, recursion, and memory management, to develop efficient solutions for computational tasks.

Lab Experiments:

1. Write a C program to input numbers and find the largest of two or three numbers using if statements and the conditional (ternary) operator (? :). Display the largest number.
2. Write a C program that takes an integer input and outputs the reversed number.
3. Write a C program to print all prime numbers between 2 and a given number n.
4. Write a C program to find the roots of a quadratic equation $ax^2+bx+c=0$.
5. Write a C program to print a triangle pattern of stars (*), where the number of lines is given by the user.
6. Write a C program to find the largest and smallest elements in an array of n numbers.
7. Write a C program to multiply two matrices of 3x3.
8. Write a C program to find the Greatest Common Divisor (GCD) of two numbers using both iteration and recursion.
9. Write a C program to demonstrate the use of different storage classes (auto, register, static, extern).
10. Write a C program to demonstrate the concepts of call-by-value and call-by-reference.
11. Write a C program that takes a string from the command-line arguments and counts the occurrence of each alphabet letter in the string.
12. Write a C program to demonstrate the usage of the enum data type.
13. Write a C program that demonstrates various string functions from the <string.h> library.
14. Write a C program that demonstrates structures and unions.
15. Write a C program that opens a file and counts the total number of characters in it.
16. Write a C program that copies content from an existing text file to a new file.

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – II
Data Structures Using C
(w.e.f. 2025-2026)

Theory: 4 Hours/Week **Credits: 4**
Internal marks = 20 **External Marks = 80**

Course Objectives:

Cob1: To discuss the linear data structures and their applications.

Cob2: To Understand Queues, Linked list and Hashing Concepts .

Cob3: To understand and implement trees and graphs with efficient traversal, searching, and optimization techniques.

Cob4: Analyze and implement advanced searching and sorting techniques, including hashing and overflow handling, to optimize data organization and retrieval.

Course Outcomes:

CO1: Understand and implement fundamental data structures, including arrays and stacks, for efficient data manipulation and expression evaluation.

CO2: Apply linked lists, queues, and hashing techniques to optimize data storage, retrieval, and processing.

CO3: Analyze and implement tree and graph structures, including traversal techniques and efficient searching strategies.

CO4: Develop and optimize searching and sorting algorithms to enhance data organization and retrieval efficiency.

UNIT I

Introduction to Data structures: Definition, Types of Data structures.

Arrays: Arrays – ADT, ordered lists, Sparse matrices, representation of arrays.

Stacks: Stack ADT, Stacks using Arrays, Stacks using dynamic arrays, Evaluation of Expressions – Evaluating Postfix Expression, Infix to Postfix expression, checking well-formed parenthesis, reversing a string.

UNIT II

Queues: Queues ADT, operations, Circular Queues, Applications.

Linked Lists: Singly Linked Lists and Chains, Linked Stacks and Queues, Polynomials, Operations for circularly linked lists, Equivalence Classes, Doubly Linked Lists. **Hashing**: Static Hashing, Hash Tables, Hash Functions, Overflow Handling, Theoretical Evaluation of Overflow Techniques.

UNIT III

Trees: Introduction, Binary Trees, Binary Tree Traversals, Heaps, Binary Search trees (BST): Definition, Searching an element, Insertion into a BST, and Deletion from a BST, Efficient Binary Search Trees. AVL Trees: Definition, Insert, search and delete operations.

Graphs: Graph Abstract Data Type, Elementary Graph operations, Graph Traversal Techniques - DFS and BFS, Minimum Cost Spanning Trees - Prim's and Kruskal's Algorithms.

UNIT IV

Searching and Sorting: Sequential search, Binary search, Hash Tables: Hashing Functions, Overflow Handling, Theoretical Evaluation of Overflow Techniques. Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort, Heap sort, List and Table Sorts.

Suggested Book

Horowitz E, Sahni S and Susan Anderson-Freed, Fundamentals of Data structures in C, 2nd Edition (2008), Universities Press.

Reference Books

1. Mark A Weiss, Data Structures and Algorithm Analysis In C, Second Edition (2002), Pearson
2. Kushwaha D. S and Misra A.K, Data structures A Programming Approach with C, Second Edition (2014), PHI.
3. ilberg R. F and Forouzan B. A, Data structures: A Pseudocode Approach with C, Second Edition (2007), Cengage Learning
4. Tanenbaum A. M , Langsam Y. Augenstein M. J, Data Structures using C, Second Edition (2008), Pearson.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, Introduction to Algorithms, Third Edition (2009), MIT Press
6. Chandan Banerjee and Atanu Das, "Data Structures and Algorithms in C and PYTHON", University Press, 2023.
7. YedidyahLangsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Data Structures Using C and C++, Second Edition (2009), PHI

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – II
Data Structures Using C Lab

(w.e.f. 2025-2026)

Practical: 2 Hours/Week **Credits: 1** **Marks: 50**

Course Objectives:

Cob1: Develop and implement various data structures such as arrays, linked lists, stacks, queues, trees, and graphs for efficient data manipulation.

Cob2: Apply sorting, searching, and hashing techniques to solve computational problems effectively.

Course Outcomes:

CO1: Implement fundamental and advanced data structures, including arrays, linked lists, stacks, queues, trees, and graphs.

CO2: Apply efficient searching, sorting, and hashing techniques to solve computational problems.

Lab Experiments (Using C programming Language):

1. Implementation of Stacks and Queues using Arrays.
2. Implementation of Circular Queue.
3. Implementation of Infix to Postfix Conversion, Postfix Expression Evaluation.
4. Implementation of Singly Linked List
5. Implementation of Doubly Linked List.
6. Implementation of Circular Linked List.
7. Implementation of Stacks using Linked Lists
8. Implementation of Queues using Linked Lists.
9. Implementation of Linear search and Binary Search.
10. Implementation of Operations on Binary Tree
11. Implementation of Binary Search Tree.
12. Implementation of Traversal on Graphs.
13. Implementation of Selection, Bubble and Insertion Sort.
14. Implementation of Merge Sort.
15. Implementation of Quick Sort.

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – III
Database Management Systems
(w.e.f. 2025-2026)

Theory: 4 Hours/Week **Credits: 4**

Internal marks = 20 **External Marks = 80**

Course Objectives:

Cob1: Understand the fundamental concepts, architecture, and advantages of database management systems over traditional file-based systems.

Cob2: Design efficient database schemas using the Entity-Relationship model and apply normalization techniques to minimize redundancy.

Cob3: Develop SQL queries and PL/SQL programs for efficient data manipulation, retrieval, and transaction control in relational databases.

Cob4: Implement transaction management, concurrency control, and security mechanisms to ensure data integrity and secure database operations.

Course Outcomes:

CO1: Explain the fundamental concepts, architecture, and advantages of Database Management Systems.

CO2: Design and normalize relational database schemas using the Entity-Relationship model and normalization techniques.

CO3: Develop SQL and PL/SQL queries to perform data manipulation, retrieval, and transaction management.

CO4: Implement concurrency control, recovery mechanisms, and security measures to ensure data integrity and protection.

Unit I

Introduction to Database Management System: Data, Information, Metadata, Database, Database Management System, File Based System, Drawbacks of File-Based System, The Database approach, Components of Database Environment, Advantage of DBMS, Three-Tier Architecture, Database Languages, Data Models, Database Users, Database Administrator, Types of Databases.

Unit II

Entity-Relationship Model: Introduction, Components of ER Diagrams, Entities, Attributes, Keys, Strong and Weak entity sets, Relationships, Types of Entities, Types of Attributes, Types of Relationships, Degree of a relationship, Cardinality. Generalization and Specialization, Aggregation and Composition, Transforming E-R model to Relational model.

Normalization: Data Redundancy, Functional Dependencies, Basic Normal Forms: First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), Boyce Codd Normal Form (BCNF), De-normalisation.

Unit III

Structured Query Language: Basic form of SQL query, Data Types, Integrity Constraints, Null Values, Commit, rollback, Logical operators, Special Operators, Order by, Group by and Having clause, Aggregate Functions, Set Operators, Alias, Joins, Nested Sub queries, Correlated Sub queries, Views: Create a view, Delete a View.

PL/SQL: Introduction, Structure of PL/SQL, Elements, Data Types, Control Structures, Steps to Create a PL/SQL Program, Iterative Control Cursors, Steps to Create a Cursor, Procedures, Functions, Packages, Exception Handling, Database Triggers, Types of triggers.

Unit IV

Transaction Management: What is transaction, Properties of Transactions, Database, Concurrency control, serializability, recoverability, Concurrency control with locking methods, concurrency control with time stamping methods, Concurrency control with optimistic methods, Deadlock. Database Recovery: The Need for Recovery, Transactions and Recovery, Recovery Facilities, Recovery Techniques. Database Security: Threats, Computer Based Controls: Authorization, Access Controls, Views, Backup and Recovery, Integrity, Encryption, RAID.

Suggested Books

1. Abraham Silberschatz, H. Korth and S. Sudarshan, Database System Concepts, 6th Ed., Tata McGraw Hill, 2011
2. Thomas M. Connolly, Carolyn E. Begg, Database Systems–A Practical Approach to Design, Implementation, and Management (6e)
3. Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, 4th Revised Edition with CD-ROM by Ivan Bayross, BPB Publications.

FACULTY OF SCIENCE

B.Sc. (Computer Science)

SEMESTER – III

Database Management Systems Lab

(w.e.f. 2025-2026)

Practical: 2 Hours/Week **Credits:1** **Marks: 50**

Course Objectives:

Cob1: Develop and manage relational database schemas by creating tables, defining primary and foreign keys, and performing data manipulation using SQL queries.

Cob2: Implement PL/SQL programming concepts such as procedures, functions, triggers, and exception handling for efficient database management.

Course Outcomes:

CO1: Construct and execute complex SQL queries to retrieve, modify, and analyze data based on real-world scenarios.

CO2: Apply PL/SQL programming techniques to automate database operations and enforce business rules.

Lab Experiments:

1. Create a database having three tables to store the details of students of Computer Department in your college:

   Student (Roll number, Name of student, Date of birth, Address, Marks, Phone number)

   Paper (Paper code, Name of the Paper)

   Attendance (College roll number, Paper Code, Attendance, Marks).

   a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.

   b) Design a query that will return the records from the table name **Paper** along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper2.

   c) List all students who live in "Warangal" and have marks greater than 60 inpaper1.

   d) Find the total attendance and total marks obtained by each student.

   e) List the names of all the students who secured highest marks in paper2.

2. Create the following tables, enter at least 5 records in each table and answer the queries given below:

   Employee (Person_Name, Street, City)

   Works (Person_Name, Company_Name, Salary)

   Company (Company_Name, City)

   Manages (Person_Name, Manager_Name)

   a) Identify primary and foreign keys.

   b) Alter table employee, add a column "email" of type varchar(20).

   c) Find the name of all managers who work for both Samba Bank and NCB Bank.

   d) Find the names, street address and cities of residence and salary of all employees who work for "Samba Bank" and earn more than $10,000.

   e) Find the names of all employees who live in the same city as the company for which they work.

f) Find the highest salary, lowest salary and average salary paid by each company.

g) Find the sum of salary and number of employees in each company.

h) Find the name of the company that pays highest salary.

3. Create the following tables, enter at least 5 records in each table and answer the queries given below.

Suppliers (SNo, Sname, Status, SCity)

Parts (PNo, Pname, Colour, Weight, City)

Project (JNo, Jname,Jcity)

Shipment (Sno, Pno, Jno, Qunatity)

a) Identify primary and foreign keys.

b) Get supplier numbers for suppliers in Paris with status>20.

c) Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.

d) Get suppliers names for suppliers who do not supply part P2.

e) For each shipment get full shipment details, including total shipment weights.

f) Get all the shipments where the quantity is in the range 300 to 750 inclusive.

g) Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.

h) Get the names of cities that store more than five red parts.

i) Get full details of parts supplied by a supplier in Hyderabad.

j) Get part numbers for part supplied by a supplier in Warangal to a project in Chennai.

k) Get the total number of project supplied by a supplier (say, S1).

l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).

4. Write a PL/SQL Program to demonstrate Procedure.

5. Write a PL/SQL Program to demonstrate Function.

6. Write a PL/SQL program to Handle Exceptions.

7. Write a PL/SQL Program to perform a set of DML Operations.

8 . Create a View using PL/SQL program.

9. Write a PL/SQL Program on Statement Level Trigger.

10. Write a PL/SQL Program on Row Level Trigger.

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – IV
Programming in Java

(w.e.f. 2025-2026)

Theory: 4 Hours/Week **Credits: 4**

Internal marks = 20 **External Marks = 80**

Course Objectives:

Cob1: To demonstrate an understanding of the fundamental principles of object-oriented programming in Java by defining classes, calling methods, and utilizing class libraries.

Cob2: To analyze and apply concepts of interfaces, packages, exception handling, and multithreading in Java.

Cob3: To apply the Collection Framework and event handling to solve real-world problems in Java.

Course Outcomes:

CO1: Design and Implement Java programming fundamentals, including syntax, object–oriented concepts, and string handling.

CO2: Apply inheritance, polymorphism, interfaces, and exception handling to develop modular and reusable Java applications.

CO3. Implement multithreading, collections framework, and file handling for efficient Java programming.

CO4: Develop Java applications that incorporate Swing, JavaFX for creating visually appealing user interfaces

UNIT I

Introduction to Java: Features of Java, bytecode, Structure of java program, language Fundamentals – tokens (comments, Identifiers, Keywords), data types, variables and types of variables (reference, local, static, final), command line arguments, operators, expressions, type conversion and casting, control statements.

Object-Oriented Programming (OOP) Basics: Classes and Objects - Concepts of classes, objects, constructors, methods (types of methods), access control (public, private, protected and default). this keyword, garbage collection, nested and inner classes, String, StringBuffer and StringBuilder.

UNIT II

Inheritance: Inheritance concept, types of inheritance, Member access rules, use of super and final. **Abstract** classes and methods, **Polymorphism** - method overriding (Runtime-time Polymorphism) and method overloading (Compile-time Polymorphism), overloading constructors

Interfaces: Defining an interface, implementing interfaces, extending interface. **Packages:** inbuilt packages, user defined packages - Defining, Creating and Accessing a Package, importing packages.

Exception handling: Benefits of exception handling, Exception hierarchy, exception Vs error, classification, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, re throwing exceptions, built in exceptions, Implementing custom exceptions.

UNIT III

Multithreading: Java Thread Model – Thread, Thread Life Cycle, The Main Thread, Ways to create a Thread – Implementing Runnable interface, Extending Thread class, creating multiple threads, thread priorities, synchronization.

Collections: Overview of Java Collection frame work, **java.util** - Collection classes – Array List, Linked List, Hash Set, Tree Set, Collection Interfaces – Collection, List, Set, Accessing Collection via iterator, working with Map. Other Utility classes: String Tokenizer, Date, Calendar, Gregorian calendar, Scanner.

Java Input/Output: exploring java.io, Java I/O classes and interfaces, File, Stream classes, byte stream, character stream, serialization.

UNIT IV

Swing: JFrame, JLabel, ImageIcon, JTextField, JPasswordField, the Swing buttons, JTabbedpane, JScrollPane, JList, JComboBox, Event Handling: Event Classes, Source of Events, Event Listener Interfaces. Handling mouse and keyboard events.

JavaFX: Introduction, Setup, Scene Graph (root, children, hierarchy). Layout containers –Vbox, HBox, GridPane, StackPane, Buttons, controls, Styling in JavaFX, Animation and Transitions in JavaFX.

Suggested Books

1. Herbert Scheldt, "The Complete Reference Java, 11th Edition, Tata McGraw Hill, 2021.
2. James M Slack, Programming and Problem Solving with JAVA, Thomson Learning, 2002.
3. C Thomas Wu, An Introduction to Object Oriented Programming with Java 6th Edition, McGraw Hill Publishing, 2019.
4. H. M. Dietel and P. J. Dietel, Java How to Program, Sixth Edition, Pearson Education/PHI
5. Carl Dea, Mark Heckler, Gerrit Grunwald, José Pereda, " JavaFX 8: Introduction by Example", 1st Edition, Apress - 2014

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – IV
Programming in Java Lab

(w.e.f. 2025-2026)

Practical: 2 Hours/Week **Credits:1** **Marks: 50**

Course Objectives:

Cob1: Develop a strong foundation in Java programming, covering core concepts such as object-oriented principles, exception handling, multithreading, and collections.

Cob2: Design and implement interactive Java applications using GUI frameworks like Swing and JavaFX, incorporating event handling and advanced Java functionalities.

Course Outcomes:

CO1: Apply object-oriented programming concepts, exception handling, and multithreading to develop efficient and scalable Java applications.

CO2: Design interactive user interfaces using Swing and JavaFX while implementing event handling and advanced Java utilities

Lab Experiments:

1. Write a Java program that demonstrates the concept of a class, along with method overloading.
2. Write a Java program that reads a line of integers, displays each integer, and calculates the sum of all the integers.
3. Write a Java program to demonstrate single-level and multi-level inheritance .
4. Write a Java program to showcase the use of interfaces and abstract classes.
5. Write a Java program to implement exception handling .
6. Write a Java program that illustrates the concept of threads using Thread class and runnable interface.
7. Write a Java program to demonstrate thread synchronization.
8. Write a java program to create a package and import the classes from the defined package .
9. Write a Java program to illustrate the use of collection classes, such as ArrayList and LinkedList.
10. Write a Java program that demonstrates the concept of I/O streams.
11. Write a Java program to implement the concept of serialization.
12. Write a Java program that functions as a simple calculator. Use a grid layout to arrange buttons for digits and operations (+, -, *, %), and include a text field to display the result.
13. Create a simple user profile form using Swing, with fields for name, age, and a profile picture.
14. Develop a JavaFX application that demonstrates the following core concepts:
    a.Use a VBox as the root node to establish a basic scene graph hierarchy.
    b.Utilize layout containers such as HBox, GridPane, and StackPane to organize UI elements.
    c.Incorporate Button, Label, and TextField controls for user interaction

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – V
Python Programming

(w.e.f. 2025-2026)

Theory: 4 Hours/Week **Credits: 4**

Internal marks = 20 **External Marks = 80**

Course Objectives:

Cob1: Understand Python syntax, control structures, and loops to develop basic programs.

Cob2: Learn function design, recursion, and file handling for modular programming.

Cob3: Implement data structures like lists, tuples, strings, dictionaries, and sets for efficient data management.

Cob4: Apply object-oriented principles and GUI programming to develop interactive applications.

Course Outcomes:

CO1: Implement basic Python programming constructs, including decision structures, loops, and input/output operations for problem-solving

CO2: Develop modular and recursive programs using functions and handle file operations and exceptions efficiently.

CO3: Manipulate and process data effectively using lists, tuples, dictionaries, sets, and string operations.

CO4: Apply object-oriented programming principles and design interactive GUI applications using the Tkinter module.

Unit I

Introduction to Python Programming: How a Program Works, Program Development Cycle, Input, Processing and Output, displaying output with the print function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations - Operators, Type conversions, Expressions, Data Output, Decision Structures and Boolean Logic: if, if-else, if-elif-else statements, nested decision structures, comparing strings, logical operators, Boolean variables.

Repetition Structures: Introduction, while loop, for loop, Calculating a Running Total, Input Validation Loops, Nested Loops.

Unit II

Functions: Introduction, defining and calling a void function, designing a Program to use functions, local variables, passing arguments to functions, global variables and global constants, value-returning functions-generating random numbers, writing our own value-returning functions, the math module, storing functions in modules.

Recursion: Introduction, Problem Solving with Recursion, Examples of Recursive Algorithms.

File and Exceptions: Introduction to file input and output, using loops to process files, processing records, exceptions.

Unit III

Lists and Tuples: Sequences, introduction to lists, list slicing, finding items in lists with the in operator, list methods and useful built-in functions, copying lists, processing lists, two- dimensional lists, Tuples.

Strings: Basic string operations, string slicing, testing, searching, and manipulating strings. Dictionaries and Sets: Dictionaries, Sets, Serializing Objects.

Unit IV

Object-Oriented Programming: Procedural and Object-Oriented Programming, Classes, Working with Instances, Techniques for Designing Classes, Inheritance, Polymorphism.

GUI Programming: Graphical User Interfaces, Using the tkinter Module, Display text with Label Widgets, Organizing Widgets with Frames, Button Widgets and Info Dialog Boxes, Getting Input with Entry Widget, Using Labels as Output Fields, Radio Buttons, Check Buttons.

Suggested Book

1. Tony Gaddis, 'Starting Out With PYTHON' 4th Edition, Pearson, 2019
2. Reema Thareja, 'PYTHON PROGRAMMING using problems solving approach', 2nd Edition, OXFORD PRESS, 2022.

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – V
Python Programming Lab
(w.e.f. 2025-2026)

Practical: 2 Hours/Week **Credits:1** **Marks: 50**

Course Objectives:

Cob1: Develop foundational programming skills in Python, covering data handling, control structures, functions, and file operations.

Cob2: Implement problem-solving techniques using recursion, object-oriented programming, and GUI development with Tkinter.

Course Outcomes:

CO1: Apply Python programming concepts to design, implement, and debug efficient programs for real-world applications.

CO2: Utilize functions, file handling, exception handling, and GUI programming to build interactive and user-friendly applications.

Lab Experiments:

1. Program to display the information: Your name, Full Address, Mobile Number, College Name, Course Subjects .
2. Program to find the sum of all prime numbers between 1 and 1000
3. Program to print the sum of first 'n' natural numbers
4. Program that reads set of integers and displays first and second largest numbers.
5. Program with a function that accepts two arguments: a list and a number 'n'. It should display all the numbers in the list that are greater than the given number 'n'.
6. Program with a function to find how many numbers are divisible by 2, 3,4,5,6 and 7 between 1 to 1000.
7. Write a Python program to read a file, display its contents, and then append a new line to the file.
8. Create a function that takes two numbers and returns their division result. Handle division by zero using a try-except block and print an error message.
9. Program to read the lists of numbers as L1, print the lists in reverse order without using reverse function
10. Write a Python program to check if a given string is a palindrome. The program should use string slicing and loops for checking.
11. Write a program that combine lists L1 and L2 into a dictionary.
12. Write a recursive function to calculate the factorial of a given number and print the result.
13. Program to implement the inheritance
14. Program to implement the polymorphism
15. Using tkinter, create a simple calculator GUI that can add, subtract, multiply, and divide two numbers. Display results in a text label or message box.
16. Design a tkinter form where the user can input their name, email, and phone number, and then display a confirmation message.

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – VI
Elective: (A)Web Technologies

(w.e.f. 2025-2026)

Theory: 4 Hours/Week **Credits: 4**

Internal marks = 20 **External Marks = 80**

Course Objectives:

COb1: Understand the fundamentals of web technologies, including HTML, CSS, JavaScript, and responsive web design.

COb2: Develop skills in designing and building web applications using HTML, CSS, and JavaScript.

COb3: Learn about responsive web design principles and techniques using Bootstrap.

COb4. Understand the role of JavaScript in web development and learn to use it for form validation, DOM manipulation, and event handling.

UNIT I

Introduction to World Wide Web: Web Browsers, Webservers, DOM, HTTP. HTML5: Introduction, Structure of HTML page, HTML5 Tags, Links, Input, Images, Favicon, Lists, Tables, Creating Forms, Html audio, Html Media, Styling Forms, Placeholder, Inline and Block elements, Id vs Class elements.

UNIT II

CSS3–Basics: Need and Benefit of CSS3, CSS3 Syntax, Comments, Including CSS3 in HTML Documents (Inline, Embedded and External Style Sheets).CSS3-Selectors: Universal Selector, Element Type Selector, Id Selectors, Class Selectors, Group Selectors. CSS3-Styles: CSS Color, CSS Background, CSS Fonts, CSS Text, CSS Links, CSS Lists, CSS Tables.CSS3-Box Model: Margin, Padding, Border, Outline, Visibility, Display, Multiple Columns.

UNIT III

Responsive Web Design (RWD) : Introduction, Viewport, Creating Responsive Websites, Responsive Images, Responsive Texts. RWD-Media Queries: Introduction, Media Types, Device Breakpoints. RWD-Grid View: Introduction, grid-row, grid-column.

Introduction to Bootstrap: Introduction, Bootstrap Grid System –Grid classes, rows, columns, container, Break Points, **Components:** buttons, Alert, Carousel, Dropdown, Card, Navbar.

UNIT IV

Javascript: Introduction to JavaScript, JavaScript and Forms Variables, Functions, Operators, Conditional Statements and Loops, Arrays, DOM Methods, Strings, Dialogue Boxes – Alert, Confirmation, prompt Box. Events Handling (Mouse Events, Keyboard Events, Window Events).

Introduction to XML: XML basics, Structuring data, Document type definition and its types, XML Namespaces, XML DOM.

Suggested Books:

1. Robert W.Sebesta, Programming the World Wide Web, 8thEdition, Pearson Education, 2021.
2. Internet & World Wide Web-HOW TO PROGRAM-5th Edition, Deitel. Published by Pearson (July 14th 2021) - Copyright © 2012.
3. Silvio Moreto, Bootstrap 4 By Example, 1st Edition, Packt Publishing, 2018.

FACULTY OF SCIENCE

B.Sc. (Computer Science)

SEMESTER – VI

Elective: (A) Web Technologies Lab

(w.e.f. 2025-2026)

Practical: 2 Hours/Week **Credits:1** **Marks: 50**

Course Objectives:

Cob1: Develop a strong foundation in web development by utilizing HTML5, CSS3, and JavaScript to create structured and visually appealing webpages.

Cob2: Implement responsive and interactive web designs using Bootstrap, DOM manipulation, and form validation techniques

Course Outcomes:

CO1: Build and style user-friendly, responsive web pages incorporating multimedia, navigation, and interactive elements.

CO2: Apply modern web technologies to enhance user experience through structured layouts, dynamic content, and validation techniques.

Lab Experiments:

1. Visual Studio Code Installation and installing the requires extensions.
2. Create a Webpage Layout using Semantic elements
3. Add Audio and Video element to a Webpage.
4. Styling text and fonts using CSS3 properties.
5. Create a navigation menu using lists and Style Lists and Links using CSS3 properties.
6. Styling tables using CSS3 properties.
7. Styling Webpage backgrounds using CSS3 properties.
8. Demonstrate Form validation.
9. Demonstrate DOM methods.
10. Demonstrate HTML events
11. Design the Responsive Layout for a Web Page for an Educational Institution.
12. Design the Responsive Webpage Using Bootstrap components.

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – VI
Elective: (B) Computer Networks
(w.e.f. 2025-2026)

Theory: 4 Hours/Week	**Credits: 4**
Internal Marks = 20	**External Marks = 80**

Course Objective:

Cob1: Understand the fundamental concepts of data communication, network components, topologies, and protocols, along with the OSI and TCP/IP reference models.

Cob2: Learn error detection and correction techniques, data link layer protocols, and Ethernet standards.

Cob3: Explore network layer functionalities, internetworking devices, IP addressing, subnetting, and routing algorithms.

Cob4: Gain knowledge of transport and application layer protocols, including TCP, UDP, DNS, HTTP, and socket programming.

Course Outcome:

CO1: Explain data communication principles, network models, and transmission media used in modern communication.

CO2: Analyze error control methods and Ethernet technologies for efficient data link layer communication.

CO3: Demonstrate the ability to configure IP addressing and implement routing techniques for network communication.

CO4: Apply transport and application layer protocols for secure and reliable data transfer over the internet.

Unit I

Data Communications: Components - Direction of Data flow - networks - Components and Categories - types of Connections - Topologies - Protocols and Standards - ISO/OSI model - Layered Architecture, Functions of Layers, TCP/IP Reference Model.

Transmission Media: Guided Media-Twisted Pair Cable Coaxial Cable, Optical Fiber, Unguided Media-Satellite Communication, and Cellular Telephony.

Unit II

Data link Layer: Error detection and correction, CRC, Hamming code, Flow Control and Error control - stop and wait - go back-N ARQ - selective repeat ARQ-sliding window - HDLC.

Introduction to Ethernet: IEEE 802.3 -IEEE 802.4 -IEEE 802.5, Bridges.

Unit III

Network Layer: Networking and Internetworking Devices: Repeaters, Bridges, Routers, Gateways, Brouters, Switches, virtual circuit and Datagram approach, Routers IP addressing, Subnetting, CIDR.

Routing - Distance Vector Routing, Link State Routing.

Unit IV

Transport Layer: Services of transport layer, Multiplexing. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), Socket Programming: What is a Socket, Two Types of internet Sockets, Primitive and Advanced System calls. Client-Server background: a simple stream server, a simple stream client, datagram sockets.

Application Layer: Domain Name Space (DNS) - SMTP - FTP - HTTP – WWW

Suggested Books

1. Andrew S. Tanenbaum, 'Computer Networks', Pearson Education; Fourth Edition, 2011.
2. Behrouz A. Forouzan, 'Data communication and Networking', Tata McGraw-Hill, 2009.
3. Brain Beej Hall, "Beej's Guide to Network Programming using Internet Sockets",2001
4. W. Richard Stevens , "Unix Network Programming", Prentice Hall , 2003

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – VI
Elective: (B) Computer Networks Lab

(w.e.f. 2025-2026)

Practical: 2 Hours/Week **Credits:1** **Marks: 50**

Course Objectives:

Cob1: To develop hands-on experience with network diagnostic tools and simulation techniques for analyzing network behavior and performance.

Cob2: To implement and simulate networking protocols, error correction methods, and congestion control mechanisms using programming and simulation tools.

Course Outcomes:

CO1: Gain practical skills in network analysis, protocol implementation, and troubleshooting using tools like Wireshark, tcpdump, and network simulators.

CO2: Demonstrate the ability to design, implement, and evaluate network communication programs using TCP/UDP sockets and routing algorithms.

Lab Experiments:

1. Learn to use commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and trace route PDUs using a network protocol analyzer and examine.
2. Write a HTTP web client program to download a web page using TCP sockets.
3. Applications using TCP sockets like: Echo client and echo server
4. Use a tool like Wireshark to capture packets and examine the packets.
5. Write a code simulating ARP/RARP protocols.
6. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS.
7. Study of TCP/UDP performance using Simulation tool.
8. Simulation of Distance Vector/Link State Routing algorithm.
9. Simulation of an error correction code like CRC.
10. Simulation of DNS using UDP sockets.

FACULTY OF SCIENCE

B.Sc. (Computer Science)

SEMESTER – VI

Elective: (C) Data Science with Python

(w.e.f. 2025-2026)

Theory: 4 Hours/Week **Credits: 4**

Internal Marks = 20 **External Marks = 80**

Course Objectives:

Cob1: Understand the fundamentals of Data Science, its applications, and types of data.

Cob2: Learn NumPy for numerical data processing, including array operations and mathematical computations

Cob3: Explore Pandas for data handling, processing, and transformation techniques.

Cob4: Master data visualization techniques using Matplotlib for effective data representation.

Course Outcomes:

CO1: Understand the fundamentals of Data Science, its applications across various domains, and the different types of data used in data-driven decision-making.

CO2: Demonstrate proficiency in handling numerical data using NumPy, including array manipulation, mathematical operations, indexing, and transformations.

CO3: Apply Pandas for data manipulation, including importing, cleaning, merging, and analyzing structured and unstructured datasets.

CO4: Effectively visualize and interpret data using Matplotlib, creating various types of plots and customizing visual representations for better data insights.

Unit I

Introduction to Data Science: Introduction- Definition - Data Science in various fields - Examples - Impact of Data Science - Data Analytics Life Cycle - Data Science Toolkit - Data Scientist - Data Science Team.

Introduction to Data: Data, Types of Data: Numeric, Categorical, Graphical, High Dimensional Classification of digital Data: Structured, Semi-Structured and Unstructured.

Unit II

Numpy Arrays: The Basics of NumPy Arrays, NumPy Arrays Creation Using array () Function, Array Attributes, NumPy Arrays Creation with Initial Placeholder Content, Integer Indexing, Array Indexing, Boolean Array Indexing, Slicing and Iterating in Arrays, Basic Arithmetic Operations on NumPy Arrays, Mathematical Functions in NumPy, Changing the Shape of an Array, Stacking and Splitting of Arrays, Broadcasting in Arrays. Numpy Matrices: The Basics of NumPy Matrices, Computation on NumPy, Universal Functions, Aggregations: Min, Max, and other functions, Comparisons, Masks, and Boolean Logic, Fancy Indexing, NumPy and Structured data, Sorting.

Unit III

Introduction to Pandas: Environment Setup, Introduction to Data, and Data Processing with Pandas: Introduction, Environment Setup, Working with Data Series and Frames: Pandas Data Structures: Series, Data Frame, Panel.

Importing Data from files: Processing CSV File, JSON File, HTML Files.

Data Pre-Processing: Append, Reshaping Data, Taming Pandas File I/O. Handling missing Values Data Indexing and Selection, Merge, Concatenation, Grouping, filtering, Transformation, Aggregation Merging Datasets - inner join, left join, right join, outer join.

Unit IV

Data Visualization with matplotlib: Simple Line Plots, Simple Scatter Plots, Pie chart, Box Plot, Matplotlib Grid, Display images, Tables, Date handling, Log and Polar plot. Visualizing Errors, Density and Contour Plots, Histograms, Binnings and Density, Customizing Plot Legends, Customizing Colorbars, Multiple Subplots, Text and Annotation. Customizing Ticks, Customizing Matplotlib: Configurations and Style sheets. Three-dimensional plotting with Matplotlib.

Suggested Books

1. Jake VanderPlas, 'Python Data Science Handbook - Essential Tools for Working with Data', 1st edition, O'Reilly Publishers, 2017.
2. Samir Madhavan, "Mastering Python for Data Science", Packt Publishing, 2015.
3. Laura Lgual; Santi Segui, 'Introduction to Data Science A Python Approach to Concepts, Techniques and Applications', 2nd Edition, Springer Publications, 2024.
4. Sandhya Arora; Latesh Malik, 'Data Science and Analytics with PYTHON' Univesity Press(India) Pvt.Ltd, 2023.

{FACULTY OF SCIENCE

B.Sc. (Computer Science)

SEMESTER – VI

Elective: (C) Python for Data Science Lab

(w.e.f. 2025-2026)

Practical: 2 Hours/Week **Credits: 1** **Marks: 50**

large{Course Objectives:

Cob1: Develop proficiency in handling, processing, and analyzing structured data using Python libraries such as NumPy and Pandas.

Cob2: Gain expertise in data visualization techniques using Matplotlib for effective representation and interpretation of data.

{Course Outcomes:

CO1: Demonstrate the ability to read, merge, filter, and aggregate data from various file formats like CSV and JSON using Python.

CO2: Apply data visualization concepts by creating plots, charts, and modifying graphical representations for meaningful insights.

Lab Experiments:

1. Write a Python Program to read data from .CSV Files and perform data aggregation & filtering.
2. Write a Python Program to read data from multiple JSON Files and perform data merging.
3. Write a Python Program to implement Data Merge using merge() operation.
4. Write a Python Program to demonstrate Data Aggregation Using Group by.
5. Write a Python program to implement Array Manipulation, Searching and Splitting using Numpy.
6. Write a Python Program to Implement Numpy Universal Functions.

Using NumPy module, perform the following:

a) Create a 2D numpy array with 3 columns and 10 rows filled with random numbers

b) Perform Aggregate functions

c) Perform Data Sorting

7. Write a Pandas program to perform following on DataFrame

a) Create and display

b) Insert new column and renaming existing column

c) Select first & last 10 rows

d) Filter based upon specific column

e) Reshape a dataframe using pivot_table function.

f) Count the number of missing values in each column. .

8. Given two lists Days and Speed of a car Days=[1,2,3,4,5,6,7,8]

Speed=[70,52,56,63,71,67,54,71]

a) Plot a Basic line plot between days and car speed

b) Change the line style to green dotted line, marker shape to +, change markersize, markerface color

c) Plot Axes Labels, Chart title, Legend

d) Plot Histogram and Pie chart for the above data

9. Write a Matplotlib program to perform programming

a) Insert and display image data

b) Converting color image into grayscale

c) Selecting a part of image.

FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEC-1:(A) LibreOffice Lab

(w.e.f..2025-2026)

SEMESTER – III

Theory: 2 Hours/Week **Credits: 2**

Course Objective:

Cob1: Students will learn to use LibreOffice Writer and Impress to create and format documents and presentations, including text styles, tables, and transitions.
Cob2: Students will learn to use LibreOffice Calc for data analysis and LibreOffice Base for managing student data with functions, tables, queries, and reports.

Course Outcomes:

CO1: Students will create and format documents and presentations in LibreOffice Writer and Impress, using text styles, tables, and transitions.

CO2: Students will analyze data in LibreOffice Calc and manage student data in LibreOffice Base, using functions, tables, and reports.

Note: LibreOffice should be taught in lab.

Lab Experiments:

LibreOffice Writer:

1. Open LibreOffice Writer, create a new document, type some text, and apply different text styles (bold, italics, font size, etc.). Save the document.
2. Create a table with 3 rows and 4 columns in a new document. Add data to the table, then modify it by adding or removing rows/columns. Save the document.
3. Create a letter using Mail Merge. Use a list of names and addresses insert the details into the letter. Save the document.

LibreOffice Impress:

4. Add a title slide, then format one slide by changing the background color and adding a transition effect. Save the changes.
5. Create a presentation with at least 5 slides, then practice presenting it in slideshow mode, using slide transitions. Save the presentation.
6. Open LibreOffice Impress, create a new presentation, and add at least three slides with text and images. Save the presentation

LibreOffice Calc:

7. In the student table, use functions like SUM or AVERAGE to calculate total marks or average marks, then create a chart to display the data. Save the workbook.
8. Apply conditional formatting to highlight students with marks above a certain value and use the filter feature to show students with marks greater than a specific number.
9. Open LibreOffice Calc, create a table with student names, roll numbers, and marks, then save the workbook.

LibreOffice Base:

10. Open LibreOffice Base and create a table for student information with the following fields: Roll No, Name, M1 (Marks in Subject 1), M2 (Marks in Subject 2), M3 (Marks in Subject 3), Total, and Average. Use Design View and Table Wizard to create the table. Save it.

11. Add some student records, change the marks for one student, and delete another student's record. Save the table.

12. Use the Query Wizard to create a simple query to filter students with marks above a certain value, then create a report to show the results.

FACULTY OF SCIENCE

B.Sc.(Computer Science)

SEC-1:(B) POWER BI

(w.e.f..2025-2026)

SEMESTER –III

Theory: 2 Hours/Week Credits: 2

|Course Objectives:

COb1: To familiarize students with the basic concepts of Power BI, data connectivity, and transformation techniques

COb2: To introduce students to data modeling, DAX functions, and visualization techniques in Power BI.

Course Outcomes:

CO1: Students will gain the skills to set up Power BI, connect to various data sources, and clean and transform data using Power Query Editor

CO2: Students will learn to build data models, use DAX for calculations, create visualizations, and share reports using Power BI.

UNIT I

Introduction to Power BI and Data Preparation : Introduction to Power BI and its components, installation of Power BI Desktop, and exploring tables, Connecting Power BI to relational databases, Excel, and CSV files, Using Power Query Editor for data shaping, transformation, and cleaning.

UNIT II

Data Modelling, DAX, and Visualizations: Creating and optimizing data models, building relationships, merges, and appends, Using DAX for calculated columns and measures, understanding visualizations, creating reports, and sharing dashboards via Power BI Service.

Suggested Book

Mastering Power BI - 2nd Edition, Chandraish Sinha, bpb.

Reference Books:

1. Microsoft Power BI Dashboards Step by Step| First Edition| By pearson
2. Shroff/O'Reilly Learning Microsoft Power BI: Transforming Data into Insights (Grayscale Indian Edition) (Paperback, Jeremey Arnold)

Lab Experiments:

1. Connecting Power BI to an Excel file and cleaning data using Query Editor.
2. Connecting Power BI to a relational database and performing data transformation.
3. Building a simple data model and establishing relationships between tables.
4. Create a data model with multiple related tables and apply DAX measures to calculate key metrics.
5. Build a dashboard with visualizations that highlight business insights.
6. Publish and share a report on Power BI Service.

FACULTY OF SCIENCE

B.Sc. (Computer Science)

SEC-2:(A) Introduction to Artificial Intelligence

(w.e.f.,2025-2026)

SEMESTER – IV

Theory: 2 Hours/Week **Credits: 2**

Course Objective:

Cob1: Understand AI problem-solving approaches and search techniques, including heuristic methods for optimization.

Cob2: Learn various knowledge representation techniques, reasoning methods, and the fundamentals of expert systems and game-playing strategies.

Course Outcome:

CO1: Explain AI problem-solving strategies and implement search techniques, including heuristic approaches.

CO2: Apply knowledge representation methods, reasoning techniques, and expert system architectures to solve AI-related problems.

UNIT I

Concepts in AI: Problem solving in AI, Defining an AI Problem as state space search, production systems. AI problem characteristics.

Search techniques: Breadth first search, depth first search.

Heuristic Search Techniques: Generate and test, hill climbing, best first search, Heuristic function applications, problem Reduction, simulated annealing.

UNIT II

Knowledge Representation in AI: Knowledge representations approaches & issues in knowledge representation, Knowledge Representation using predicate logic, forward and backward reasoning, semantic nets, frames, scripts & conceptual Dependency.

Game Playing: The Minimax Search procedure, Adding Alpha-Beta Cutoffs, Additional Refinements, Iterative Deepening.

Expert systems: Architecture, the knowledge base, inference engine, Knowledge acquisition Expert system development process MYCIN as an example. Types of Expert Systems – Rule Based, Case Based, Model Based.

Suggested Books

1. E. Rich and Knight, "Artificial Intelligence", 3nd Edition, 2009, TMH.

2. S. J. Russel and P. Norvig, "**Artificial Intelligence: A Modern Approaches**", Prentice Hall.

3. George F Luger, "**Artificial Intelligence**", Fourth Edition, Pearson Education Asia,

FACULTY OF SCIENCE

B.Sc.(Computer Science)

SEC-2:(B) Software Engineering

(w.e.f.2025-2026)

SEMESTER – IV

Theory: 2 Hours/Week **Credits: 2**

Course Objectives:

Cob1: Understand the fundamental principles of Software Engineering and explore different Software Development Life Cycle (SDLC) models

Cob2: Learn the basics of software design, architecture, and testing methodologies.

Course Outcomes:

CO1: Explain the importance of software engineering concepts and analyze various SDLC models and requirement specifications.

CO2: Apply software design principles and testing techniques to ensure software reliability and maintainability.

Unit I

Fundamentals of Software Engineering: Introduction to Software Engineering, Software Development Life Cycle (SDLC): Waterfall Model, Agile Model, Software Requirements: Functional vs. Non-functional Requirements, Software Requirement Specification (SRS).

Unit II

Basics of Software Development & Testing: Software Design, Introduction to Software Architecture, Object-Oriented Design (OOD) Basics, Introduction to Software Testing: Unit Testing, Integration Testing, Black Box & White Box Testing, Software Maintenance.

Suggested Books

1. 'Software Engineering' – Ian Sommerville
2. 'Software Engineering: A Practitioner's Approach' – Roger S. Pressman & Bruce R. Max
3. 'Fundamentals of Software Engineering' – Rajib Mall

Reference Books

1. 'Software Requirements' – Karl Wiegers & Joy Beatty
2. 'Clean Code: A Handbook of Agile Software Craftsmanship' – Robert C. Martin

FACULTY OF SCIENCE

B.Sc. (Computer Science)

SEC-3:(A) Cyber Security

(w.e.f..2025-2026)

SEMESTER – V

Theory: 2 Hours/Week **Credits: 2**

Course Objectives:

Cob1: Understand various types of cyber crimes, their impact, and techniques used by cybercriminals.

Cob2: Gain knowledge of cyber security challenges, legal frameworks, and regulatory measures to combat cyber threats

Course Outcomes

CO1: Understand Cyber Crimes and analyze the types of cyber crimes.

CO2: Understand the Issues and challenges of Cyber security and Apply appropriate cyber laws.

Unit I

Cyber Crime: Cyber Crime , Cyber crime targeting computers and mobiles, cyber crime against women and children. Email scams, Phishing, Vishing, Smishing, Financial frauds, social engineering attacks, malware and ransom ware attacks, zero day and zero click attacks, Data diddling attacks, Spyware, Logic bombs,

Unit II

Cyber security and Cyber Laws: Cyber security, Issues and challenges of Cyber security Cyber crime and legal landscape around the world, IT Act, 2000 and its amendments. Limitations of IT Act, 2000. RBI guidelines on Digital payments and Customer protection in unauthorized banking transactions. Relevant provisions of Payment Settlement Act, 2007. Personal Data Protection Bill and its compliance, Cyber Laws and Legal and ethical aspects related to new technologies- AI/ML, IoT, Block chain, Darknet and Social media. Reporting of cybercrimes, Remedial and mitigation measures.

Suggested Books:

1. Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Sumit Belapure and Nina Godbole, Wiley India Pvt. Ltd.
2. Cyber Laws: Intellectual Property & E-Commerce Security by Kumar K, Dominant Publishers.
3. Security in the Digital Age: Social Media Security Threats and Vulnerabilities by Henry A. Oliver, Create Space Independent Publishing Platform.
4. Data Privacy Principles and Practice by Natraj Venkataramanan and Ashwin Shriram, CRC Press.

FACULTY OF SCIENCE

B.Sc. (Computer Science)

SEC-3:(B) BLOCK CHAIN TECHNOLOGY

(w.e.f..2025-2026)

SEMESTER – V

Theory: 2 Hours/Week **Credits: 2**

Course Objective:

Cob1: Understand the fundamental concepts, evolution, and key components of blockchain technology, including cryptographic principles and consensus mechanisms.

Cob2: Explore the structure, mining process, and consensus mechanisms of Bitcoin and Ethereum, along with the role of smart contracts and blockchain-based cryptocurrency ecosystems.

Course Outcome:

CO1: Students will be able to explain the fundamental properties, evolution, and components of blockchain technology, including cryptographic functions and different blockchain platforms

CO2: Students will be able to analyze the structure and working of Bitcoin and Ethereum, including transaction mechanisms, mining processes, consensus protocols, and smart contract functionalities.

UNIT I

Introduction: Introduction to Blockchain; Properties of Blockchain; Evolution of Blockchain. **Cryptocurrency and Block chain:** Anonymity and Pseudonymity in Cryptocurrency; Programmable Money; Hash Functions and Merkle Trees; Components of Blockchain Ecosystem; Cryptography and Consensus Algorithms; Types of Blockchain; Blockchain Platforms.

UNIT II

Bitcoin Platform: Bitcoin and its uses; Bitcoin Ecosystem; Structure of a Bitcoin Transaction; Nodes in a Bitcoin Network; Bitcoin Mining, Bitcoin Economics; Types of bitcoin Mining; Consensus mechanism in bitcoin.

Introduction to Ethereum: What is Ethereum; Introducing Smart Contracts; Cryptocurrency in Ethereum; Mining in Ethereum; Consensus mechanism in Ethereum; Technologies that support Ethereum; Ethereum Programming Language; Ethereum Test Networks.

Suggested Books

1. Andreas M. Antonopoulos, "Mastering Bitcoin. Programming the Open Blockchain", O'Reilly, 2017.
2. Vikram Dhillon, David Metcalf, Max Hooper 'Blockchain Enabled Applications', Apress,2019
3. Chandramouli Subhramanian, Asha A George Abhilash.K.A and Meena Karthikeyan, BLOCKCHAIN TECHNOLOGY, University Press, 2023

FACULTY OF SCIENCE

B.Sc. (Computer Science)

SEC-1:(C) Digital Documentation and Analysis

(w.e.f..2025-2026)

SEMESTER – V

Theory: 2 Hours/Week **Credits: 2**

Course Objective:

Cob1: Learn to create, edit, format, and manage documents in LibreOffice Writer, and develop visually appealing presentations with effective slide designs in LibreOffice Impress.

Cob2: Understand LibreOffice Calc for data handling and advanced features, and learn LibreOffice Base for database creation and Management.

Course Outcomes:

CO1: *Apply* skills in LibreOffice writer for creating, editing, and formatting documents to improve communication.

CO2: *Analyze* data in LibreOffice Calc using functions, charts, and advanced tools for better decision-making.

UNIT I

LibreOffice writer: Introduction, LibreOffice Writer Basics, Opening, Saving, Printing and Closing documents, Formatting Text, Table Manipulation, Mail Merge, Headers and Footers.

LibreOffice Impress: Introduction – Basics, using LibreOffice Impress, Components of Presentation window, Creation of Presentation, Preparation of Slides, Presentation of Slides.

UNIT II

LibreOffice Calc: Introduction, Basic Elements of LibreOffice Calc –opening a calc, printing a calc, Saving a workbook, Functions and Charts, Conditional Formatting, Filtering Data –Advanced Filters.

LibreOffice base: Introduction, Tables, Creating tables in design View, Creating a table using Wizard, adding, Modifying and Deleting Records, Queries, Creating a simple Query using Query Wizard, Forms, Reports.

Suggested Books

1. Fundamentals of Computers by Reema Thareja, Second Edition, Oxford higher Education.
2. Introduction to Information Technology – Pearson Education –ITL Education Solutions Limited

Reference Books

1. Computer Fundaments by Anitha Goel –Pearson, 1st Edition. Pearson Education.
2. Mastering Advanced Excel, Ritu Arora, BPB Publications.
3. Introduction to Computers by Peter Norton –McGraw Hill Education (India) Private Limited –Seventh Edition.

FACULTY OF SCIENCE
B.Sc. (Computer Science)
Semester-V
Generic Elective (GE)
Emerging Trends in Computer Science

(w.e.f. 2025-2026)

Theory: 4 Hours/Week **Credits: 4**

Internal marks = 20 External Marks = 80

Course Objectives:

Cob1: To introduce emerging computing technologies and their societal impacts.

Cob2: To provide fundamental knowledge of computer networks and e-commerce.

Cob3: To explain the need for information security and various security measures.

Cob4: To explore advanced technologies like IoT, Blockchain, Quantum Computing, and AR/VR with applications.

Course Outcomes:

CO1: Understand emerging computing technologies and their ethical impacts.

CO2: Demonstrate knowledge of computer networks and internet-based applications.

CO3: Identify security threats and apply suitable security solutions.

CO4: Apply advanced technologies like IoT, Blockchain, Quantum Computing, and AR/VR in real-world scenarios.

Unit I

Computer Networks: Introduction, Connection Media, Network Topologies, Types of Networks, Networking Devices, Distributed Networking, Peer-to-peer Computing, Wireless Network, Bluetooth.
Internet and E-commerce: Internet Services, Types of Internet Connections, ECommerce, E-Cash and Electronic Payment System.

Unit II

Introduction to Emerging Computer Technologies: Grid Computing, Cloud Computing, Utility Computing, On-demand Computing, Artificial Intelligence, Machine Learning Concepts and Applications, Ethical and Societal Impacts of Emerging Technologies.

Unit III

Advanced Computing Paradigms and Applications: Internet of Things (IoT) Applications, Blockchain Technology- Fundamentals and applications, Quantum Computing -Basics and Applications – Augmented and Virtual Reality (AR/VR).

Unit IV

Introduction to Security: Introduction to Information security, Need for Information Security, Internet Security, Threats to Information Systems, Information Assurance, Cyber Security, Application Security, Data Security Considerations, Security Threats, Security Technologies.

Suggested Books

1. Dr. Surya Prakash T, Ritendra G, Praveen Kumar S, KLSI, Introduction to Information Security and Cyber Laws (Dreamtech Publication)

2. Emerging Technologies in Computing: Theory, Practice, and Applications, M. N. Hoda, Rajiv Pandey, Dharmendra Singh Rajput, Publisher: Springer, Year: 2021,ISBN: 9789813345763 (Print), 9789813345770 (Online)

Reference Books

1. **Handbook of Research on Cloud and Fog Computing Architectures for Data Science**, Rajiv Pandey, A. K. Tripathi, Dharmendra Singh Rajput, **Publisher:** IGI Global, **Year:** 2021, **ISBN:** 9781799850547
2. **Blockchain Technology: Platforms, Tools and Use Cases**, Sudhir Kumar Sharma, B. B. Gupta, Dharma P. Agrawal, **Publisher:** CRC Press (Taylor & Francis Group), **Year:** 2020, **ISBN:** 9780367343066
3. **Internet of Things: Principles and Paradigms**, RajkumarBuyya, Amir VahidDastjerdi (Editors), **Publisher:** Morgan Kaufmann (Elsevier), **Year:** 2016, **ISBN:** 9780128053959
4. **Cybersecurity Essentials**, Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short, **Publisher:** Wiley, **Year:** 2018, **ISBN:** 9781119362391