Assam University, Silchar



Four Year Undergraduate Programme

Implemented under NEP 2020

Effective from the Academic Year 2023-24

Syllabus of Computer Application (2nd Year)

Approved in the 96th meeting of the Academic Council on 12th April 2024 vide Resolution No. AC:96:04-24:5

Programme Specific Outcome

Bachelor in Computer Application with Honours/Honours and Research

Computer Application (CA) has been evolving as an important branch of Science and Technology in the last two decades and it has carved out a space for itself like Computer Science and Engineering. Computer Application spans theory and more application and it requires thinking both in abstract terms and in concrete terms. The ever-evolving discipline of Computer Application has strong connections to other disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers and its applications, but finding a solution requires both computer science expertise and knowledge of the particular application domain. BCA and BCA (Hons) are aimed at undergraduate level training facilitating multiple career paths. Students who have graduated, can take up postgraduate programmes in CS or MCA leading to research as well as R&D, can be employable at IT industries, or can pursue a teaching profession or can adopt a business management career. BCA and BCA (Hons) aims at laying a strong foundation of computer application at an early stage of the career. There are several employment opportunities and after successful completion of BCA, graduating students can fetch employment directly in companies as programmer, Web Developer, Software Engineer, Network Administrator, Data Scientist, or AI/ML personnel.

The present Curriculum Framework for BCA degrees is intended to facilitate the students to achieve the following.

- 1. Students will have a comprehensive understanding of computer science principles, programming languages, software development methodologies, and data structures.
- 2. Students will be proficient in one or more programming languages and possess the ability to develop software applications, write efficient code, and solve programming problems.
- 3. BCA graduates will have strong analytical and problem-solving skills, allowing them to identify and resolve complex computing problems through the application of logical reasoning and critical thinking.
- 4. Graduates will be capable of designing, developing, and testing software applications using appropriate software engineering principles and methodologies.
- 5. Students will acquire knowledge and skills in designing and managing databases, including the ability to create database schemas, query data, and ensure data integrity.
- 6. BCA graduates will have effective oral and written communication skills, enabling them to collaborate with clients, understand requirements, and document software projects accurately.
- 7. Students will be adept at working in multidisciplinary teams, demonstrating the ability to communicate, cooperate, and contribute to team projects effectively.
- 8. Graduates will understand the principles of information security and possess knowledge of techniques to secure computer systems, networks, and applications.
- 9. BCA graduates will be aware of ethical considerations related to computer science and information technology and demonstrate a commitment to professional and ethical practices.
- 10. Students will be equipped with a strong foundation that allows them to adapt to emerging technologies, learn new programming languages, and continuously update their skills to keep pace with the evolving field of computer science.

Semester	Course Code	Title of the Courses	Credits
Ι	CADSC101	Fundamentals of Information Technology	3
	CADSC102	Discrete Mathematics	3
	CADSC151	Data Structure	3
	CADSC152	Lab on Data Structure	3
III	CADSC201	Computer Organization and Architecture	4
	CADSC202	Operating System	4
	CADSC251	Programming with Java	4
IV	CADSC252	Database Management System	4
- '	CADSC253	Lab on Java Programming & DBMS	4
	CADSC301	Computer Graphics and C++	4
V	CADSC302	System Analysis and Design	4
	CADSC303	Lab on Computer Graphics and C++	4
	CADSC351	Computer Network and Internet Technology	4
N/I	CADSC352	E-Commerce	4
VI	CADSC353	Programming with PHP	4
	CADSC354	Lab on PHP & Network Programming	4
	CADSC401	Design and Analysis of Computer Algorithms	4
VII	CADSC402	Theory of Computation and Compiler Design	4
	CADSC403	Artificial Intelligence	4
	CADSC404	Lab on DACA & Compiler Design	4
VIII	CADSC451	(A) Research MethodologyOR(B) Software Engineering	4
	CADSC452	(A) Image ProcessingOR(B) Data Analytics	4
	CADSC453	Natural Language Processing	4
	CADSC454	(A) IoT OR (B) Cloud Computing	4
	CADSC455	Research Project/Dissertation	12

Table 1: Semester wise list of Computer Application DSC Courses

Semester	DSM1/D SM2	Course Code	Title of Courses	Credits
Ι	DSM 1	CADSM101	Programming with C	3
II	DSM 2	CADSM151	Programming with C	3
III	DSM 1	CADSM201	Database Management System	4
	DSM 1	CADSM251	Lab on C & DBMS	3
IV	DSM 2	CADSM252	Database Management System	3
V	DSM 1	CADSM301	Computer Graphics	3
V	DSM 2	CADSM302	Computer Graphics	3
VI	DSM 2	CADSM351	Lab on C & DBMS	4
VII	DSM 1	CADSM401	Internet Technology	4
VIII	DSM 2	CADSM451	Internet Technology	4

Table 2: Semester wise list of Computer Application DSM Papers

Table 3: Semester wise list of Computer Application SEC Courses

Semester	Course Code	Title of Courses	Credits
Ι	CASEC101	Programming with C	3
II	CASEC151	Python Programming	3
III	CASEC201	Web Programming	3

Table 4: Semester wise list of Computer Application IDC Courses

Semester	Course Code	Name of the Paper	Credits
Ι	CAIDC101	Fundamentals of Information Technology	3
II	CAIDC151	Programming Fundamentals with C	3
III	CAIDC201	Introduction to Web Designing & Cyber Security	3

Syllabi of <u>Computer Application</u> DSC Courses

Semester	: 111
Course Type	: DSC
Course Code	: CADSC201
Name of the Course	: Computer Organization and Architecture
Learning level	: Intermediate-level course
Credits	: 4
Contact Hours	: 60
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course Objectives: The course objective is to

- 1. Students should gain knowledge about the various components of a computer system.
- 2. Learn about different types of instruction sets (e.g., CISC, RISC) and understand how instructions are executed by the CPU.
- 3. Introducing assembly language programming and its relationship to computer architecture.
- 4. Exploring the memory hierarchy, including registers, cache memory, main memory (*RAM*), and secondary storage.
- 5. Studying the organization and design principles of processors, including pipelining, parallelism, and microarchitecture.
- 6. Understanding how input/output devices are interfaced with the CPU and memory, including concepts such as I/O controllers, interrupts, and DMA.

Unit-I

Register transfer and Micro operation: Register transfer language, bus & memory transfer, Arithmetic; Logic; Shift Micro operation, Arithmetic Logic unit. Computer organization design-Computer Instruction with details like timing; control and Instruction Cycle, memory Reference; Input and output and Interrupt Instructions, Design of a Basic Computer.

Unit-II

Programming the Basic Computer: Machine and Assembly Language with Programming Details. Microprogrammed Control – Control memory Address sequencing, Microprogram Examples, Design of control unit.

Unit-III

Central processing Unit: General Resister organization, Stack organization, Instruction format, addressing modes, Data transfer and Manipulation, Program control, Length and type of instruction, RISC and CISC.

Unit-IV

Computer Arithmetic: Addition; Subtraction; Multiplication; Division Algorithms with hardware implementation, Floating point and Decimal Arithmetic Operations.

Unit-V

Input output Organization: Input Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, DMA, Input Output Processor, Serial Communication. Advances in Memory system- Memory Hierarchy, Different Memory Organization (Main, Auxiliary, Associative, and Cache) and Virtual Memory.

Course Learning Outcomes: *After successful completion of the course, the students will be able to:*

- 1. Understand Fundamental Concepts of computer organization and architecture.
- 2. Analyze and Evaluate Architectural Designs.
- 3. Apply Assembly Language Programming and understanding how high-level language constructs map to machine instructions and memory organization.
- 4. Design and optimize computer systems for performance.
- 5. Analyze Real-World Architectures

Text Books:

- 1. M.M. Mano, Computer System Architecture, Pearson Education Asia, 3rd Edition, 2015.
- 2. V. Carl Hamacher, Zvonko G. Vranesic, Safwat G. Zaky, Computer organization, McGraw Hill, 5th Edition, 2010.

Reference Books:

1. PVS Rao, Perspectives in Computer Architecture, PHI, 2nd Edition, 2005.

Semester	: III
Course Type	: DSC
Course Code	: CADSC202
Name of the Course	: Operating System
Learning level	: Intermediate-level course
Credits	: 4
Contact Hours	: 60
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course Objectives: The course objective is to

- 1. Understand the fundamental concepts of operating systems, including processes, threads, scheduling, etc.
- 2. Learn about the structure and components of operating systems, such as the kernel, device drivers, system calls, and user interface layers.
- 3. Explore process management concepts, including process creation, scheduling algorithms, inter-process communication, synchronization mechanisms, and deadlock handling.
- 4. Understand memory management techniques, including virtual memory, paging, segmentation, memory allocation algorithms
- 5. Explore operating system security mechanisms, including authentication, authorization, access control, encryption, and security policies

Unit I

Introduction: Basic OS functions, resource abstraction, types of operating systems– multiprogramming systems, batch systems, time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

Operating System Organization: Processor and user modes, kernels, system calls and system programs.

Unit II

Process Management: System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and preemptive scheduling algorithms.

Unit III

Process Coordination: Synchronization, concurrent processes, critical section, semaphores, methods for inter-process communication; deadlocks.

Unit IV

Memory Management: Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory.

Unit V

File and I/O Management: Directory structures, file operations, file allocation methods, device management.

Protection and Security: Policy mechanism, Authentication, Internal access Authorization.

Course Learning Outcomes: *After successful completion of the course, the students will be able to:*

- 1. Understanding of Operating System Concepts including processes, threads, memory management, file systems, and I/O operations.
- 2. Skills to analyse and solve problems related to process scheduling, resource allocation, etc.
- 3. Develop the ability to implement basic operating system functionalities such as process management, memory management, and file system operations.
- 4. Proficiency in file system organization, file I/O operations, directory management, and file allocation methods.

Text Books:

- 1. A Silberschatz, P.B. Galvin, G. Gagne, **Operating Systems Concepts**, John Wiley Publications. 8th Edition, 2008.
- 2. A.S. Tanenbaum, Modern Operating Systems, Pearson Education, 3rd Edition, 2007.

- 1. G. Nutt, Operating Systems: A Modern Perspective, Pearson Education, 2nd Edition 1997.
- 2. W. Stallings, Operating Systems, Internals & Design Principles, PHI, 5th Edition, 2008.

Semester	: IV
Course Type	: DSC
Course Code	: CADSC251
Name of the Course	: Programming with Java
Learning level	: Intermediate-level course
Credits	: 4
Contact Hours	: 60
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course Objectives: The course objective is to

- 1. Understand the basics of Java programming language syntax, including variables, data types, operators, and expressions.
- 2. Learn the principles of object-oriented programming, including classes, objects, inheritance, polymorphism, and encapsulation
- 3. Learn how to handle exceptions in Java using try-catch blocks, and understand the concept of checked and unchecked exceptions
- 4. Learn how to perform input and output operations in Java, including file handling, reading from/writing to files, and using streams.
- 5. Gain an introduction to GUI development in Java using Swing or JavaFX, including creating windows, panels, buttons, and event handling
- 6. Learn how to connect Java applications to databases using JDBC (Java Database Connectivity), and perform database operations such as querying and updating data.

Unit I

Introduction to Java : Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods)

Unit II

Arrays, Strings and I/O: Creating & Using Arrays (One Dimensional and Multidimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

Object-Oriented Programming Overview: Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

Unit III

Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata: Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

Unit IV

Exception Handling, Threading, Networking and Database Connectivity: Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread, Accessing and manipulating databases using JDBC.

Unit V

Applets and Event Handling: Java Applets/frame: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, text fields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

Course Learning Outcomes: *After successful completion of the course, the students will be able to:*

- 1. Demonstrate a solid understanding of fundamental programming concepts.
- 2. *Apply object-oriented programming principles*
- 3. Understand and apply the syntax and features of the Java programming language, including packages, access modifiers, interfaces, etc.
- 4. Demonstrate proficiency in handling exceptions in Java programs using try-catch blocks, throwing and catching exceptions
- 5. Design and develop graphical user interfaces (GUIs) using Java Swing or JavaFX, including creating windows, panels, buttons, text fields, and event handling

Text Books:

- 1. Ken Arnold, James Gosling, David Homes, "**The Java Programming Language**", 4th Edition, 2005.
- James Gosling, Bill Joy, Guy L Steele Jr, Gilad Bracha, Alex Buckley, "The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
- 3. Joshua Bloch, "Effective Java", Addison-Wesley, 2nd Edition, 2008.

Reference Books:

- 1. Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
- 2. E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.
- 3. Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011.

Semester	: IV
Course Type	: DSC
Course Code	: CADSC252
Name of the Course	: Database Management System
Learning level	: Intermediate-level course
Credits	: 4
Contact Hours	: 60
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course Objectives: The course objective is to

- 1. Understand the fundamentals of databases, including definitions, types of databases and their applications in various domains
- 2. Learn data modeling techniques such as Entity-Relationship Diagrams (ERDs) and normalization to design efficient and scalable database schemas.
- 3. Understand the principles of relational databases, including tables, rows, columns, keys (primary, foreign), relationships, and constraints.
- 4. Gain proficiency in SQL for database querying, data manipulation (insertion, deletion, modification), data definition (creating tables, indexes) etc.
- 5. Learn about transaction properties (ACID), concurrency control mechanisms (locking, timestamps), and recovery techniques (undo, redo, logging) to ensure data consistency and reliability.
- 6. Understand database security principles, including authentication, authorization, encryption, and auditing, to protect sensitive data.

Unit I

Introduction: Characteristics of database approach, data models, database system architecture and data independence.

Entity Relationship(ER) Modeling: Entity types, relationships, constraints.

Unit II

Relation data model: Relational model concepts, relational constraints, relational algebra, SQL Queries.

Unit III

Database design: Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition, Normal forms (upto BCNF).

Unit IV

Transaction Processing: ACID properties, concurrency control, Locking protocols, Deadlock detection and prevention.

Unit V

File Structure and Indexing: Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files (Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

Course Learning Outcomes: *After successful completion of the course, the students will be able to*:

- 1. Demonstrate a solid understanding of fundamental database concepts, including data models, schemas, keys, relationships, and database management system architectures
- 2. Demonstrate proficiency in SQL (Structured Query Language) for database querying, data manipulation, data definition, and data control operations on relational databases.
- 3. Apply database management concepts and techniques to analyze, design, and implement solutions for real-world database problems and scenarios
- 4. Evaluate and critique database designs, implementations, and performance optimizations, and propose improvements

Text Books:

- 1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems, Pearson Education, 6th Edition, 2010.
- 2. A. Silberschatz, H.F. Korth, S. Sudarshan, **Database System Concepts**, McGraw Hill, 6th Edition, 2010.

- 1. C. J. Date, An Introduction to Database Systems, Pearson India, 8th edition, 2005.
- 2. R. Ramakrishanan, J. Gehrke, **Database Management Systems**, 3rd Edition, McGraw Hill, 2002.

Semester	: IV
Course Type	: DSC
Course Code	: CADSC253
Name of the Course	: Lab on Java Programming & DBMS
Learning level	: Intermediate-level course
Credits	: 4
Contact Hours	: 120
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course Objectives:

- 1. Java Programming Skills Development.
- 2. Object-Oriented Design and Implementation.
- 3. Data Structure and Algorithm Implementation.
- 4. Graphical User Interface (GUI) Development.
- 5. Database Application Development
- 6. SQL Querying and Data Manipulation
- 7. Database Design and Implementation
- 8. Database Connectivity and Interaction

This paper provides practical knowledge of Java Programming and SQL queries. List of laboratory programming assignments (not limited to these):

Lab on Java Programming

- 1. To find the sum of any number of integers entered as command line arguments
- 2. To find the factorial of a given number
- 3. To learn use of a single dimensional array by defining the array dynamically.
- 4. To learn use of length in case of a two dimensional array
- 5. To convert a decimal to binary number
- 6. To check if a number is prime or not, by taking the number as input from the keyboard.
- 7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument.
- 8. Write a program that shows the working of different functions of String and StringBuffer Classes like setCharAt(), setLength(), append(), insert(), concat() and equals().
- 9. Write a program to create a distance class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer.
- 10. Modify the distance class by creating a constructor for assigning values (feet and inches) to the distance object. Create another object and assign a second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.

- 11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions(from lower to higher data type).
- 12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of the final keyword.
- 13. Write a program to show the use of static functions and to pass variable length arguments in a function.
- 14. Write a program to demonstrate the concept of boxing and unboxing.
- 15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
- 16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci series is given in a different file belonging to the same package.
- 17. Write a program that illustrates different levels of protection in classes/subclasses belonging to the same package or different packages.
- 18. Write a program DivideByZero that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
- 19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
- 20. Write a program to create your own exception types to handle situations specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
- 21. Write a program to demonstrate priorities among multiple threads.
- 22. Write a program to demonstrate multi thread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
- 23. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseErsted(), mousePressed, mouseReleased() and mouseDragged().
- 24. Write a program to demonstrate different keyboard handling events.
- 25. Write a program using JDBC to perform: a) insert b) delete c) update and d) search operations.

Lab on DBMS

- 1. Implementation of DDL commands of SQL with suitable examples a) Create table b) Alter table c) Drop Table.
- 2. Implementation of DML commands of SQL with suitable examples a) Select b) Insert c) Update d) Delete.
- 3. Implementation of different types of function with suitable examples a) Number function b) Aggregate Function c) Character Function d) Conversion Function e) Date Function.
- 4. Implementation of different types of operators in SQL a) Arithmetic Operators b) Logical Operators c) Comparison Operator d) Special Operator e) Set Operation.
- 5. Implementation of different types of Joins a) Inner Join b) Outer Join c) Natural Join etc.
- 6. Study and Implementation of a) Group By & having clause b) Order by clause c) Indexing.

- 7. Study & Implementation of a) Sub queries b) Views.
- 8. Study & Implementation of different types of constraints.
- 9. Study & Implementation of Database Backup & Recovery commands. Study & Implementation of Rollback, Commit, Savepoint.
- 10. Creating Database /Table Space a) Managing Users: Create User, Delete User b) Managing roles:-Grant, Revoke.

Course Learning Outcomes: After successful completion of the course, the students will be able to:

- 1. Demonstrate proficiency in Java programming by implementing various programming tasks, exercises, and projects using Java language features and libraries.
- 2. Design and develop graphical user interfaces (GUIs) for Java applications using Swing including creating interactive components and event handling.
- 3. Integrate Java applications with relational databases using JDBC (Java Database Connectivity) to perform database operations such as querying, insertion, deletion, and modification.
- 4. Write SQL queries to retrieve, update, delete, and manipulate data stored in relational databases.
- 5. Establish database connections from Java applications, handle database transactions, manage database resources, and implement error handling and exception management strategies.

Syllabi of <u>Computer Application</u> DSM Courses

Semester	: 111
Course Type	: DSM
Course Code	: CADSM201
Name of the Course	: Database Management System
Learning level	: Intermediate-level course
Credits	: 4
Contact Hours	: 60
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course Objectives: The course objective is to

- 1. Understand the fundamentals of databases, including definitions, types of databases and their applications in various domains
- 2. Learn data modeling techniques such as Entity-Relationship Diagrams (ERDs) and normalization to design efficient and scalable database schemas.
- 3. Understand the principles of relational databases, including tables, rows, columns, keys (primary, foreign), relationships, and constraints.
- 4. Gain proficiency in SQL for database querying, data manipulation (insertion, deletion, modification), data definition (creating tables, indexes) etc.
- 5. Learn about transaction properties (ACID), concurrency control mechanisms (locking, timestamps), and recovery techniques (undo, redo, logging) to ensure data consistency and reliability.
- 6. Understand database security principles, including authentication, authorization, encryption, and auditing, to protect sensitive data.

Unit I

Introduction: Characteristics of database approach, data models, database system architecture and data independence.

Entity Relationship(ER) Modeling: Entity types, relationships, constraints.

Unit II

Relation data model: Relational model concepts, relational constraints, relational algebra, SQL Queries.

Unit III

Database design: Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition, Normal forms (up to BCNF).

Unit IV

Transaction Processing: ACID properties, concurrency control, Locking protocols, Deadlock detection and prevention.

Unit V

File Structure and Indexing: Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files (Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

Course Learning Outcomes: *After successful completion of the course, the students will be able to*:

- 1. Demonstrate a solid understanding of fundamental database concepts, including data models, schemas, keys, relationships, and database management system architectures
- 2. Demonstrate proficiency in SQL (Structured Query Language) for database querying, data manipulation, data definition, and data control operations on relational databases.
- 3. Apply database management concepts and techniques to analyze, design, and implement solutions for real-world database problems and scenarios
- 4. Evaluate and critique database designs, implementations, and performance optimizations, and propose improvements

Text Books:

- 1. R. Elmasri, S.B. Navathe, **Fundamentals of Database Systems**, Pearson Education, 6th Edition, 2010.
- 2. A. Silberschatz, H.F. Korth, S. Sudarshan, **Database System Concepts**, McGraw Hill, 6th Edition, 2010.

- 1. C. J. Date, An Introduction to Database Systems, Pearson India, 8th edition, 2005.
- 2. R. Ramakrishanan, J. Gehrke, **Database Management Systems**, 3rd Edition, McGraw Hill, 2002.

Semester	: IV
Course Type	: DSM
Course Code	: CADSM251
Name of the Course	: Lab on C & DBMS
Learning level	: Intermediate-level course
Credits	: 3
Contact Hours	: 90
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course objective: To provide students with practical skills and experiences in C programming and DBMS application

- 1. Develop proficiency in the C programming language by implementing various programming tasks.
- 2. Understand memory management concepts in C, including dynamic memory allocation.
- 3. Gain a basic understanding of Database Management System (DBMS) concepts.
- 4. Understand the basics of Structured Query Language (SQL) for database querying and manipulation, including SELECT, INSERT, UPDATE, DELETE statements, and simple SQL queries.

Lab on C

Problem solving of various natures by implementing programs in C Programming languages based on unit wise contents of the theory paper Programming with C. Following are some programming tasks for laboratory programming assignments but the assignments should not be limited to these only.

List of laboratory programming assignments (not limited to these):

1. Write a program to

- a) Produce ASCII equivalent of given number
- b) Find divisor or factorial of a given number
- c) Evaluate the following algebraic expressions after reading necessary values from

the user $(ax+b)/(ax-b) - 2.5 \log x - \cos 30 + |x2-y2| + sqrt (2xy) - (x5+10x4+8x3+4x+2)$

- d) Find sum of a geometric series
- e) Cipher a string
- f) Check whether a given string follows English capitalization rules
- g) Find sum of the following series $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{20}$
- h) Search whether a given substring exist in an input string or not and then delete this string from input string
- 2. Write a recursive program for the Tower of Hanoi problem.

3. The Fibonacci sequence of numbers is 1, 1, 2, 3, 5, 8..... Based on the recurrence relation F(n)=F(n-1)+F(n-2) for n>2 Write a recursive program to print the first *n* Fibonacci number.

4. Write a menu driven program for matrices to do the following operation depending on whether the operation requires one or two matrices

- a) Addition of two matrices
- b) Subtraction of two matrices
- c) Finding upper and lower triangular matrices
- d) Trace of a matrix
- e) Transpose of a matrix
- f) Check of matrix symmetry
- g) Product of two matrices.

5. Write a program that takes two operands and one operator from the user perform the operation and then print the answer

7. Write functions to add, subtract, multiply and divide two complex numbers (x+iy) and (a+ib) Also write the main program.

- 8. Write a menu driven program for searching and sorting with following options:
 - a) Searching (1) Linear searching (2) Binary searching
 - b) Sorting (1) Insertion sort (2) Selection sort
- 9. Write a program to copy one file to another, use command line arguments.
- 10. Write a program to mask some bit of a number (using bit operations)
- 11. An array of records contains information of managers and workers of a company.

Print all the data of managers and workers in separate files.

Lab on DBMS

- 1. Implementation of DDL commands of SQL with suitable examples a) Create table b) Alter table c) Drop Table.
- 2. Implementation of DML commands of SQL with suitable examples a) Select b) Insert c) Update d) Delete.
- Implementation of different types of function with suitable examples a) Number function
 b) Aggregate Function c) Character Function d) Conversion Function e) Date Function.

- 4. Implementation of different types of operators in SQL a) Arithmetic Operators b) Logical Operators c) Comparison Operator d) Special Operator e) Set Operation.
- 5. Implementation of different types of Joins a) Inner Join b) Outer Join c) Natural Join etc.
- 6. Study and Implementation of a) Group By & having clause b) Order by clause c) Indexing.
- 7. Study & Implementation of a) Sub queries b) Views.
- 8. Study & Implementation of different types of constraints.
- 9. Study & Implementation of Database Backup & Recovery commands. Study & Implementation of Rollback, Commit, Savepoint.
- 10. Creating Database /Table Space a) Managing Users: Create User, Delete User b) Managing roles:-Grant, Revoke.

Course Learning Outcomes: After successful completion of the course, the students will be able to:

- 1. Demonstrate proficiency in programming using the C language, including understanding of syntax, data types, control structures, functions, and pointers.
- 2. Utilize file handling techniques in C for reading from and writing to files, including text and binary files, and perform input/output operations effectively.
- 3. Understand memory management concepts in C, including dynamic memory allocation
- 4. Understand the basics of Structured Query Language (SQL) for database querying and manipulation, including SELECT, INSERT, UPDATE, DELETE statements, and simple SQL queries.

Semester	: IV
Course Type	: DSM
Course Code	: CADSM252
Name of the Course	: Database Management System
Learning level	: Intermediate-level course
Credits	: 3
Contact Hours	: 45
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course Objectives: The course objective is to

- 1. Understand the fundamentals of databases, including definitions, types of databases and their applications in various domains
- 2. Learn data modeling techniques such as Entity-Relationship Diagrams (ERDs) and normalization to design efficient and scalable database schemas.
- 3. Understand the principles of relational databases, including tables, rows, columns, keys (primary, foreign), relationships, and constraints.
- 4. Gain proficiency in SQL for database querying, data manipulation (insertion, deletion, modification), data definition (creating tables, indexes) etc.

- 5. Learn about transaction properties (ACID), concurrency control mechanisms (locking, timestamps), and recovery techniques (undo, redo, logging) to ensure data consistency and reliability.
- 6. Understand database security principles, including authentication, authorization, encryption, and auditing, to protect sensitive data.

Unit I

Introduction: Characteristics of database approach, data models, database system architecture and data independence.

Entity Relationship(ER) Modeling: Entity types, relationships, constraints.

Unit II

Relation data model: Relational model concepts, relational constraints, relational algebra, SQL Queries.

Unit III

Database design: Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition, Normal forms (up to BCNF).

Unit IV

Transaction Processing: ACID properties, concurrency control, Locking protocols, Deadlock detection and prevention.

Unit V

File Structure and Indexing: Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files (Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

Course Learning Outcomes: *After successful completion of the course, the students will be able to*:

- 1. Demonstrate a solid understanding of fundamental database concepts, including data models, schemas, keys, relationships, and database management system architectures
- 2. Demonstrate proficiency in SQL (Structured Query Language) for database querying, data manipulation, data definition, and data control operations on relational databases.
- 3. Apply database management concepts and techniques to analyze, design, and implement solutions for real-world database problems and scenarios
- 4. Evaluate and critique database designs, implementations, and performance optimizations, and propose improvements

Text Books:

1. R. Elmasri, S.B. Navathe, **Fundamentals of Database Systems**, Pearson Education, 6th Edition, 2010.

2. A. Silberschatz, H.F. Korth, S. Sudarshan, **Database System Concepts**, McGraw Hill, 6th Edition, 2010.

- 1. C. J. Date, An Introduction to Database Systems, Pearson India, 8th edition, 2005.
- 2. R. Ramakrishanan, J. Gehrke, Database Management Systems, 3rd Edition, McGraw Hill, 2002.

Syllabi of Computer Application SEC Courses

Semester	: III
Course Type	: SEC
Course Code	: CASEC201
Name of the Course	: Web Programming
Learning level	: Intermediate-level course
Credits	: 3
Contact Hours	: 30
Total Marks	: 100
End Semester Marks	: 80 [50+30 (Lab)]
Internal Marks	: 20

Course Objectives: The course objective is to

- 1. Develop a solid understanding of web technologies, including HTML, CSS, and JavaScript, and their roles in building dynamic and interactive websites.
- 2. Gain proficiency in client-side scripting languages like JavaScript for enhancing user interfaces, validating forms, and implementing interactive features.
- 3. Learn CSS (Cascading Style Sheets) for styling web pages, including layout design, typography, color schemes, and responsive design principles for various devices.

Unit I

Internet and the World Wide Web: Introduction to internet and its applications, Intranet, E-mail, telnet, FTP, e-commerce, video conferencing, e-business. Internet service providers, domain name server, internet address, World Wide Web (WWW): World Wide Web and its evolution, uniform resource locator (URL), browsers, search engine, TCP/IP, web server – apache, IIS, proxy server, HTTP protocol.

Unit II

Introduction to HTML: HTML Basics, Essential Tags, Tags and Attributes, Open tags & Closed tags, Text Styles and Text Arrangements, Exposure to Various Tags, Color and Background of Web Pages, Lists and their Types, Order and their Types, Hypertext, Hyperlink, Links, Anchors and URLs, Links to External Documents, Creating Table, Frame.

Unit III

Style sheets: Introduction to CSS, syntax, selectors, comments, colors, backgrounds, borders, margins, paddings, Text, fonts, icons, links, lists, tables; CSS advanced: shadows, text effects, web fonts, 2D transform, 3D transform, transitions, animations.

Creating Form, different form controls and user interface.

Unit IV

Scripting language: Java Script: Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Operators: Assignment Operators, Comparison Operators, Arithmetic Operators, Statements: Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while, with; Events and Event

Handlers : General Information about Events, Defining Event Handlers, event: onChange, onClick, on DblClick. JavaScript functions.

Unit V

Server side scripting: ASP, ASP variables, procedures, conditionals, looping, forms, cookies, session, ASP AJAX, ASP VB functions, Accessing a database from an ASP page, ADO connect, ADO Recordset.

Course Learning Outcomes: *After successful completion of the course, the students will be able to:*

- 1. Gain proficiency in HTML, CSS, and JavaScript for building the user interface of web applications and websites.
- 2. Develop skills in client-side scripting using JavaScript for implementing dynamic and interactive features, form validation
- 3. Acquire knowledge of server-side scripting language and frameworks for server-side development.

Text Books: -

- 1. N.P. Gopalan and J. Akilandeswari, **Web Technology: A Developer's Perspective,** PHI Publication, 7th Edition, 2016.
- 2. Nilakshi Jain and Ramesh Menon, **Cyber security and Cyber Laws**, Wiley Publication, 2nd Edition, 2020.

Reference Books: -

- 1. Ivan Bayross, Web Enabled Commercial Application Development Using HTML, DHTML, Java Script, Perl CGI, BPB Publications, 2009.
- 2. M. Merkow, J. Breithaupt, Information Security Principles and Practices, Pearson Education.2005

Lab on Web Programming: 30 Hours. (Practical /Project/Field work): Total marks: 30 Pass marks: 12

The web programming lab can provide students with a well-rounded understanding of web development principles.

This part provides practical knowledge of Web Programming.

Following are some programming tasks for laboratory programming assignments but the assignments are not limited to these only.

- 1. Create a HTML document consisting of HTML heading, paragraphs and images.
- 2. Create a HTML document and insert comments in the HTML source code and insert horizontal lines.
- 3. Construct HTML document to set the font of a text, size of the font, style of the font.
- 4. Create a HTML document to show how to create hyperlinks.

- 5. Create a HTML document to use an image as a link.
- 6. Create a HTML document to open a link in a new browser window.
- 7. Create a HTML document to jump to another part of a document (on the same page).
- 8. Create a HTML document to insert images from another folder or another server.
- 9. Create an image-map, with clickable regions.
- 10. Create a HTML document with all table elements (Table, Caption, Table Row, Table Data element, Table Heading Element, THEAD, TFOOT, TBODY)
- 11. Create an HTML document to make an unordered list, an ordered list, different types of ordered lists, different types of unordered lists, Nested list, Definition list.
- 12. Create HTML form with the all FORM elements (text fields, password field, Checkboxes, Radio buttons, Select elements, Drop-down list with a pre-selected value, Textarea (a multi-line text input field) and buttons.
- 13. Create HTML document with all Frame elements (FRAMESET, FRAME, NOFRAMES, and INLINE FRAME).
- 14. Create a HTML document to add AUDIO and VIDEO.
- 15. Create a HTML document to aligning images (Let the image float to the left/ right of a paragraph).
- 16. Create a HTML document to jump to a specific section within a frame
- 17. Construct a HTML document with CSS to Set the background colour of a page.
- 18. Construct a HTML document with CSS to set an image as the background of a page
- 19. Construct HTML document with CSS to Set the text color of different elements and align the text.
- 20. Construct HTML document to set different colours to visited/unvisited links, Specify a background colour for links.
- 21. Write a program in javascript to accept a name from the user and display the same name in an alert box.
- 22. Write a program in javascript to display a message in a confirm box.
- 23. Write a program in javascript to display the message _time is running out' in the status bar.
- 24. Write a program in JavaScript to enter marks of a student and find his/her grade according to the following:
 - if marks>=90 then grade A
 - if marks>=80 then grade B
 - if marks>=70 then grade C
 - if marks>=60 then grade D
 - else fail.
- 25. Write a program in JavaScript to create a button and when the button is clicked the message _Hello World' is displayed on an alert box.
- 26. Write a program in JavaScript to accept 2 nos from the user and show the working of all arithmetic operators.
- 27. Write a program in JavaScript to accept 2 strings and concatenate them.
- 28. Write a program in JavaScript to display the current date and time.
- 29. Write a program in JavaScript to find the length of an array.
- 30. Write a program in JavaScript to check whether a string is palindrome or not.
- 31. Write a program in JavaScript that responds to a mouse click anywhere on the page (using mouse click).
- 32. Write a program in JavaScript to display the contents of a check box in an alert box.
- 33. Write a program to validate a form in the user id and password forms.
- 34. Write a program in JavaScript to create a welcome cookie, Button animation, Image map with added JavaScript Simple timing, Timing event in an infinite loop

Syllabi of Computer Application IDC Courses

Semester	: 111
Course Type	: IDC
Course Code	: CAIDC201
Name of the Course	: Introduction to Web Designing & Cyber Security
Learning level	: Intermediate-level course
Credits	:3
Contact Hours	: 45
Total Marks	: 100
End Semester Marks	: 70
Internal Marks	: 30

Course Objectives: The course objective is to

- 1. To comprehend the basics of the internet and web terminologies.
- 2. To introduce the client-side scripting language concepts for developing client-side applications.
- 3. To prepare students with technical knowledge and skill needed to protect and defend computer systems, networks and data from unauthorized access, attacks and damage.

Unit I

Basics of Internet and Web: The basics of Internet, World Wide Web, IP Address, Web page, Home page, Web site, Static, Dynamic and Active web page, Web Server, Web Browser, Web Hosting, DNS, Domain Registration, URL, **Overview of Protocols**: SMTP, FTP, HTTP etc, HTTP request and response.

Unit II

Introduction to HTML: HTML Basics, Essential Tags, Tags and Attributes, Open tags & Closed tags, Text Styles and Text Arrangements, Exposure to Various Tags, Color and Background of Web Pages, Lists and their Types, Order and their Types, Hypertext, Hyperlink, Links, Anchors and URLs, Links to External Documents, Creating Table, Frame, Form and Style Sheet.

Unit III

Java Script: Scripting language, Client-Side scripting language, Java Script, Simple Java Script, variables, functions, conditions, loops, Operators, Web forms and validations.

DHTML: Features of DHTML, Combining HTML, CSS, Java Scripts, events and buttons, controlling browser.

Unit IV

Introduction to Cyber Security & Cyber Law: Cyber Law- Importance of Cyber Law, Cybercrime, categories of Cybercrime, Cyber criminology, **Cyber security:** Importance of Cyber security, Different domain of Cyber security, hardware vulnerability, software vulnerability, **Threat**: Definition, Types of Threat; Cyber-attack.

Terms associated with Cyber Crime & Cyber Security: Hacking, Cracking, Phishing, Spoofing, Data masking, Cryptanalysis, Cyber warfare, Scanning, Session hijacking, Malicious software, Strong Password, Weak Password.

Information Gathering Techniques: Tools or techniques of the attacker to gather information. **IT Act 2008:** Importance, different sections of IT Act 2008.

Course Outcomes: *After successful completion of the course, the students will be able to*

- 1. Learn the basics of the internet and web.
- 2. Design and develop the web applications using HTML and Java Scripts.
- 3. Understand the importance of Cyber Laws and Cyber Security.
- 4. Know the techniques to prevent cyber-attack and different sections of IT Act 2008.

Text Books:

- 1. N.P. Gopalan and J. Akilandeswari, **Web Technology: A Developer's Perspective**, PHI Publication, 7th Edition, 2016.
- 2. Nilakshi Jain and Ramesh Menon, **Cyber security and Cyber Laws**, Wiley Publication, 2nd Edition, 2020.

- 3. Ivan Bayross, Web Enabled Commercial Application Development Using HTML, DHTML, Java Script, Perl CGI, BPB Publications, 2009.
- 4. M. Merkow, J. Breithaupt, Information Security Principles and Practices, Pearson Education.2005