Syllabi and Scheme of Examinations
for
4-year Bachelor of Computer Applications – Single Major Program

(Based on Curriculum and Credit Framework for UG Programs under NEP)

With Effect from the session 2023-24

MAHARSHI DAYANAND UNIVERSITY ROHTAK (HARYANA)

Credit Structure for Undergraduate Programmes (Single Major)

Semester	Discipline-Specific Courses (DSC) / Major Course	Minor(MIC)/ Vocational (VOC)/ Skill Enhancement Courses (SEC)/ Internship	Multidisciplinary courses (MDC)	Ability Enhancement courses (AEC)	Dissertation	Value-Added Courses (VAC)	Total Credits
I	DSC - A1 @ 4 credits DSC – A2 @ 4 credits	MIC1 @ 4 credits SEC1@ 3 credits	MDC1 @ 3 credits	AEC1 @ 2 credits	----------	VAC1 @ 2 credits	22
II	DSC – A3 @ 4 credits DSC – A4 @ 4 credits	MIC2 @ 4 credits SEC2@ 3 credits	MDC2 @ 3 credits	AEC2 @ 2 credits	----------	VAC2 @ 2 credits	22
Students exiting the programme after second semester and securing 48 credits including 4 credits of summer internship will be awarded UG Certificate in the relevant Discipline/Subject							
III	DSC – A5 @ 4 credits DSC – A6 @ 4 credits DSC – A7 @ 4 credits	MIC3 @ 4 credits SEC3@ 3 credits	MDC3 @ 3 credits	AEC3 @ 2 credits			24
IV	DSC – A8 @ 4 credits DSC – A9@ 4 credits DSC – A10 @ 4 credits DSC – A11 @ 4 credits	MIC4(VOC)@ 4 credits	----------	AEC4 @ 2 credits	----------	VAC3 @ 2 credits	24
Students exiting the programme after fourth semester and securing 94 credits including 4 credits of summer internship will be awarded UG Diploma in the relevant Discipline/Subject							
V	DSC – A12 @ 4 credits DSC – A13 @ 4 credits DSC – A14 @ 4 credits DSC – A15 @ 4 credits	MIC5(VOC)@ 4 credits Internship @ 4 credits#	----------	----------	----------	----------	24
VI	DSC – A16 @ 4 credits DSC – A17 @ 4 credits DSC – A18 @ 4 credits DSC – A19 @ 4 credits	MIC6(VOC)@ 4 credits	----------	----------	----------	----------	20
Students will be awarded 3-year UG Degree in relevant major Discipline/Subject upon securing 136 credits.							
VII	DSC – H1 @ 4 credits DSC – H2 @ 4 credits DSC – H3 @ 4 credits DSC – H4 @ 4 credits DSC – H5 @ 4 credits	SEC3 @ 4 credits OR MIC7 (VOC) @ 4 credits OR Internship @ 4 credits	----------	----------	----------	----------	24
VIII (4yr UG Hon.)	DSC – H6 @ 4 credits DSC – H7 @ 4 credits DSC – H8 @ 4 credits DSC – H9 @ 4 credits DSC – H10 @ 4 credits	SEC4 @ 4 credits OR MIC8 (VOC) @ 4 credits OR Internship @ 4 credits	----------	----------	----------	----------	24
VIII (4yr UG Hon. with Research)	DSC – H6@ 4 credits DSC – H7@ 4 credits	SEC5 @ 4 credits OR MIC8 (VOC) @ 4 credits OR Internship @ 4 credits	----------	----------	Research project/ Dissertation @ 12 credits	---------- TOTAL CREDITS	24 184

Note: Four credits of internship earned by a student during summer internship after 2nd semester or 4th semester will be counted in 5th semester of a student who pursue 3 year UG Programmes without taking exit option.

Discipline Specific Courses/ Major Course			**Credits Distribution**			**Total Credits**	**Workload**			**Total Workload**	**Marks**			
	Nomenclature of Course	**Course Code**	**L**	**T**	**P**		**L**	**T**	**P**		**Theory**		**Practical**	**Total Marks**
											Internal	**External**		
Semester I														
DSC - A1 @ 4 credits	Mathematical Foundations of Computer Science	23BCA401DS01	4	0	0	4	4	0	0	4	30	70	-	100
DSC – A2 @ 4 credits	Computer Fundamentals & Problem Solving using C	23BCA401DS02	3	0	1	4	3	0	2	5	25	50	25	100
Semester II														
DSC – A3 @ 4 credits	Digital Logic Design	23BCA402DS01	4	0	0	4	4	0	0	4	30	70	-	100
DSC – A4 @ 4 credits	Data and File Structures	23BCA402DS02	3	0	1	4	3	0	2	5	25	50	25	100
Semester III														
DSC – A5 @ 4 credits	Operating Systems	24BCA403DS01	4	0	0	4	4	0	0	4	30	70	0	100
DSC – A6 @ 4 credits	Object Oriented Programming using C++	24BCA403DS02	3	0	1	4	3	0	2	5	25	50	25	100
DSC – A7 @ 4 credits	Database Management System	24BCA403DS03	3	0	1	4	3	0	2	5	25	50	25	100
Semester IV														
DSC – A8 @ 4 credits	Computer System Architecture	24BCA404DS01	4	0	0	4	4	0	0	4	30	70	-	100
DSC – A9 @ 4 credits	Data Communication & Computer Networks	24BCA404DS02	4	0	0	4	4	0	0	4	30	70	-	100
DSC – A10 @ 4 credits	Java Programming	24BCA404DS03	2	0	2	4	2	0	4	6	15	35	50	100
DSC – A11 @ 4 credits	Computer Graphics	24BCA404DS04	3	0	1	4	3	0	2	5	25	50	25	100
Semester V														
DSC – A12 @ 4 credits	Software Engineering	25BCA405DS01	4	0	0	4	4	0	0	4	30	70	-	100
DSC – A13 @ 4 credits	Artificial Intelligence& Expert Systems	25BCA405DS02	4	0	0	4	4	0	0	4	30	70	-	100
DSC – A14 @ 4 credits	Cloud Computing	25BCA405DS03	3	0	1	4	3	0	2	5	25	50	25	100
DSC – A15 @ 4 credits	Web Application Development	25BCA405DS04	2	0	2	4	2	0	4	6	15	35	50	100

Semester VI														
DSC – A16 @ 4 credits	Internet of Things (IoT)	25BCA406DS01	3	0	1	4	3	0	2	5	25	50	25	100
DSC – A17 @ 4 credits	Computer Security	25BCA406DS02	3	0	1	4	3	0	2	5	25	50	25	100
DSC – A18 @ 4 credits	Data Warehousing & Mining	25BCA406DS03	3	0	1	4	3	0	2	5	25	50	25	100
DSC – A19 @ 4 credits	Advanced Database Systems	25BCA406DS04	2	0	2	4	2	0	4	6	15	35	50	100

L: Lecture; T: Tutorial; P: Practical

Note:

The Syllabi and Scheme of Examinations (SOE) for Discipline Specific Courses/Major Courses for UG Semester 7 and Semester 8 will be same as applicable for Syllabi and S.O.E. for Post Graduate semester 1 and semester 2 respectively.

Syllabus for
4-Year Bachelor of Computer Applications Program
[Leading to BCA and BCA(Honours)/BCA(Honours with Research) Degrees]

First Semester

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Mathematical Foundations of Computer Science**	**Course Code**	**23BCA401DS01**
Hours/Week	**4 (4+0+0)**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

The objective of this course is to inculcate in students the fundamental mathematical background in Computer Science and to provide reasoning, learning and understanding skills. The students get exposure of various concepts of mathematics such as sets, relations, and functions, trigonometry, limit, continuity, derivatives, matrix and determinants.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand and solve the problems on set, relation and functions.
CO2 Understand the concepts of trigonometry.
CO3 Solve the problems on limit and continuity.
CO4 Understand the concepts of derivative and solve the problems on derivative.
CO5 Understand the concept of Matrix and Determinants.

Unit – I

Sets: Sets, Subsets, Equal Sets Universal Sets, Finite and Infinite Sets, Operation on Sets, Union, Intersection and Complements of Sets, Cartesian Product, Cardinality of Set, Practical applications of set theory.

Relations And Functions: Properties of Relations, Equivalence Relation, Partial Order Relation. Function: Domain and Range, Onto, Into and One to One Functions, Composite and Inverse Functions.

Unit – II

Trigonometry: Introduction, Measurement of angles, trigonometric functions, relation between trigonometric functions, signs of trigonometric functions, trigonometric functions of standard angles. Basic of inverse trigonometry.

Limits & Continuity: Limit at a Point, properties of limit, computation of limits of various types of functions, Continuity of a function at a point, Continuity over an interval.

Unit – III

Differentiation: Derivative of a function, Derivatives of sum, differences, product & quotient of functions, Derivatives of polynomial, trigonometric, exponential, logarithmic, inverse trigonometric and implicit functions, Logarithmic Differentiation, Chain rule and differentiation by substitution.

Unit – IV

Matrices: Definition, Types of Matrices, Addition, Subtraction, Scalar Multiplication and Multiplication of Matrices.

Determinants: Definition, Minors, Cofactors, Properties of Determinants, Applications of determinants in finding area of triangle, Adjoint of matrix, Inverse of matrix, solving a system of linear equations using matrix method.

Suggested Readings:

1. C.L.Liu: Elements of Discrete Mathematics, McGraw Hill.
2. Lipschutz, Seymour: Discrete Mathematics, Schaum's Series

3. Babu Ram: Discrete Mathematics, Vinayek Publishers, New Delhi.
4. Trembley, J.P & R. Manohar: Discrete Mathematical Structure with Application to Computer Science, TMH.
5. Kenneth H. Rosen: Discrete Mathematics and its applications, TMH.
6. Doerr Alan &Levasseur Kenneth: Applied Discrete Structures for Computer Science, Galgotia Pub. Pvt. Ltd.
7. Any other book(s) covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Computer Fundamentals & Problem Solving using C**	**Course Code**	**23BCA401DS02**
Hours/Week	**5 (3+0+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

This is first course in programming. The objective of this course is to inculcate knowledge about fundamental concepts of computer and logical thinking and to teach the Programming Language C. However, the process of learning a computer language will also be emphasized. Emphasis is also on semantics and problem solving. Students will be able to develop logics which will help them to create programs, applications in C.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the concepts of computer and its applications in various fields.
CO2: Understand the fundamental concepts of programming in C language.
CO3: Demonstrate an understanding of data types, control structures, functions, arrays, and pointers
CO4: Develop basic programming solutions using C language.
CO5: Apply basic programming concepts to solve practical problems.

Unit – I

Computer Fundamentals: Generations of Computers, Block Diagram along with its components, classification of computers, Applications of computers in various fields. Input/Output Devices, Memory: Concept of primary & secondary memory, Cache Memory, Secondary storage devices.

Overview of Networking & Operating System: Introduction to computer networking, Network types, Network topologies, Internet and its applications; Operating system and its functions.

Unit – II

Planning the Computer Program: Problem definition, Program design, Debugging, Types of errors in programming, Techniques of Problem Solving- Flowcharting, Algorithms

Overview of C: History of C, Importance of C, Elements of C: C character set, identifiers and keywords, Data types, Constants and Variables, Assignment statement, Symbolic constant, Structure of a C Program,printf(), scanf() Functions, Operators & Expression, type casting and conversion, operator hierarchy & associativity.
.

Unit – III

Decision making & Branching: Decision making with IF statement, IF-ELSE statement, Nested IF statement, ELSE-IF ladder, switch statement, goto statement.

Decision making & Looping: while, do-while and for loop, jumps in loops, break, continue statement, Nested loops

Unit – IV

Functions: Standard Mathematical functions, Input/output: Unformatted & formatted I/O function in C, Input functions output functions, string manipulation functions.User defined functions**:** Introduction/Definition, function prototype, Local and global variables, passing parameters, recursion.

Arrays & Pointers: Definition, types, initialization, processing an array, passing arrays to functions, Declaration and initialization of string, Input/output of string data, Concepts of pointers.

Suggested Readings:

1. Gottfried, Byron S.: Programming with C, Tata McGraw Hill
2. Gill Nasib Singh: Computing Fundamentals and Programming in C, Khanna Book Publishing Company(Private) Limited, NewDelhi.
3. Balagurusamy, E.: Programming in ANSI C, Tata McGraw-Hill
4. Jeri R. Hanly& Elliot P. Koffman: Problem Solving and Program Design in C, Addison Wesley.
5. YashwantKanetker: Let us C, BPB.
6. Rajaraman, V.: Computer Programming in C, PHI.
7. YashwantKanetker: Working with C, BPB.
8. Any other book(s) covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Programs/Lab Assignments

1. Write a C program to demonstrate the use of `printf()` and `scanf()` functions. Prompt the user to enter their name and age, then display the entered information.
2. Write a C program to define and use symbolic constants. Define symbolic constants for PI and the radius of a circle, then calculate and display the area of the circle.
3. Write a C program to demonstrate the use of various operators and expressions. Perform addition, subtraction, multiplication, and division on two numbers entered by the user and display the results.
4. Write a C program to implement decision making using if-else statements. Check if a number entered by the user is positive, negative, or zero and display the result.
5. Write a C program to implement a nested if statement.
6. Write a C program to demonstrate the use of switch statement. Simulate a simple calculator to perform addition, subtraction, multiplication, and division based on user input.
7. Write a C program to implement looping using a while loop.
8. Write a C program to implement looping using a do-while loop.
9. Write a C program to implement looping using for loop
10. Write a C program to demonstrate the use of break and continue statements. Print numbers from 1 to 10, but skip the number 5 using continue and stop the loop when it reaches 8 using break.
11. Write a C program to define a user defined function to calculate the factorial of a number, entered by the user and display the result.
12. Write a C program to demonstrate the use of arrays. Read 10 integers from the user, store them in an array to find the largest number.
13. Write a C program toread a string from the user and then display its length, and convert it to uppercase and lowercase.
14. Write a C program to demonstrate the use of pointers. Declare an integer variable, assign a value to it, and then use a pointer to display its value and address.
15. Write a C program to pass arrays to a function. Read 5 integers into an array, pass the array to a function, and return sum of the elements of the array.

Any other Practicals/Lab Assignments/Programs assigned by the teachers.

Second Semester

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Digital Logic Design**	**Course Code**	**23BCA402DS01**
Hours/Week	**4 (4+0+0)**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

To acquire the basic knowledge of digital logic levels and application to understand digital electronics circuits. This course also prepares students to perform the analysis and design of various digital electronic circuits, design and analyze sequential and combinational logic circuits.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the concept of logic gates.
CO2: Understand and use of number system and their conversion.
CO3: Learn the concept of combinational circuit and sequential circuits.
CO4: Understand the concept of Computer Organization and instruction sets.
CO5: Explore concepts related to Memory Organization and Input Output Organization.

Unit – I

Digital Systems and Binary Numbers: Digital Systems: Digital Signals, Digital Waveforms, Digital Computers and Digital Integrated Circuits. **Number Systems:** Binary Number Systems, Octal and Hexadecimal Number System. Number Base Conversions. Complements, Signed Binary Numbers and Binary Codes, Error Detection and Correction codes.

Boolean Algebra and Logic Gates: Boolean Algebra: Axiomatic Definition, Theorems and Properties. Boolean Functions, Canonical Standard forms: SOP and POS forms. **Digital Logic Gates:** NOT, OR, AND, NOR, NAND, XOR and XNOR. Universal Gates and their implementation

Unit – II

Gate Level Minimization: Karnaugh Map (K-map) Method: Simplification: Algebra postulates and Canonical forms. Prime Implicants: Types, Determination and Selection of Prime implicants.

Don't Care Conditions, NAND and NOR implementation.

Unit – III

Combinational Circuits: Introduction, Characteristics and Designing principles of Combinational circuits. Binary Adder: Half-Adder & Full-Adder, Subtractor: Half-Subtractor& Full-Subtractor, Parallel binary Adder/Subtractor, Binary Multiplier, Comparators, Multiplexers, De-multiplexers, Encoders and Decoders.

Unit – IV

Sequential Circuits: Characteristics of Sequential Circuits, Latches, **Flip-Flops:** Introduction, S-R Flip flop, J-K Flip Flop, D Flip flop, T Flip flop and Master Slave Flip flop.

Registers: Shift Registers, Applications of Registers. **Counters**: Asynchronous & Synchronous Counters. Modulo-N Counters and Up-Down Counters.

Suggested Readings:

1. Mano, M.M. : Digital Logic and Computer Design, Prentice- Hall of India.
2. Gill, Nasib Singh and Dixit J.B.: Digital Design and Computer Organisation, University Science Press (Laxmi Publications), New Delhi.
3. Stallings, William: Computer Organisation & Architecture.
4. Mano, M.M. : Digital Design, Prentice-Hall of India.
5. Anand Kumar : Fundamentals of Digital Circuits, PHI.
6. Tokheim : Digital Electronics, TMH.
7. S. Rangnekar: Digital Electronics, ISTE/ EXCEL
8. Any other book(s) covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Data and File Structures**	**Course Code**	**23BCA402DS02**
Hours/Week	**5 (3+0+2)**	**Credits(L:T:P)**	**3:0:1**
Max. Marks.	**Theory:75(50+25)** **Practical: 25(20+5)**	**Time of end term examination**	**3Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

The course aims to empower students with a comprehensive skill set in data and file structures, fostering both theoretical understanding and practical application, preparing them for analyzing and applying algorithms, design and apply efficient algorithms using data structures, understand the significance of efficient data and file organization, develop coding proficiency in data structure applications.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the fundamental concepts of data structures.
CO2: Design and implement various data structures to solve computational problems.
CO3: Apply data structures for efficient storage and retrieval of information.
CO4: Develop algorithms for searching and sorting data.
CO5: Implement file handling operations in a programming environment.

Unit–I

Introduction: Elementary data organization, Data Structure definition, Data type vs. data structure, Categories of data structures, Data structure operations, Applications of data structures.

Arrays: Introduction, Linear arrays, Representation of linear array in memory, address calculations, Traversal, Insertions, Deletion in an array, Multidimensional arrays, Parallel arrays, Sparse arrays.

Searching: Introduction, Sequential search, Binary search, Prerequisite for binary search, Comparison in terms of efficiency.

Unit–II

Sorting: Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort, Comparison in terms of their efficiency.

Stack: Introduction, Array and linked representation of stacks, Operations on stacks, Applications of stacks: Polish notation, Recursion.

Queues: Introduction, Array and linked representation of queues, Operations on queues, Deques, Priority Queues, Applications of queues.

Unit–III

Linked List: Introduction, Representation of linked lists in memory, Traversal, Insertion, Deletion, Searching in a linked list, Header linked list, Circular linked list, Two-way linked list, Threaded lists, Garbage collection, Applications of linked lists.

Tree: Introduction, Definition, Representing Binary tree in memory, Traversing binary trees, Traversal algorithms using stacks.

Graph: Introduction, Graph Theory terminology, Sequential and Linked representation of Graphs.

Unit–IV

Introduction to file structures: Concept of a file, types of files, File operations - open, read, write, close. External storage devices, Concepts of record, file, database and database system.

File Organization: Sequential file organisation – structures and processing, Record structures and access methods. Indexed sequential file organisation – structures and processing, Indexing techniques, B-trees and hashing for indexed files. Direct file organisation. Hashed File Organization - Hash function implementation.

Suggested Readings:

1. Seymour Lipschutz, "Data Structure", Tata-McGraw-Hill
2. Horowitz, Sahni& Anderson-Freed, "Fundamentals of Data Structures in C", Orient Longman.
3. Trembley, J.P. And Sorenson P.G., "An Introduction to Data Structures With Applications", McGraw-Hill International Student Edition, New York.
4. Mark Allen Weiss Data Structures and Algorithm Analysis In C, Addison- Wesley, (An Imprint Of Pearson Education), Mexico City.Prentice- Hall Of India Pvt. Ltd., New Delhi.
5. YedidyanLangsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, "Data Structures Using C", Prentice- Hall of India Pvt. Ltd., New Delhi.
6. Any other book(s) covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Lab Assignments/Programs

1. Write a C program to demonstrate basic data structure operations such as creating, inserting, deleting, and displaying elements in an array.
2. Write a C program to insert and delete an element at a specified position in a linear array.
3. Write a C program to implement and demonstrate a sequential search on an array.
4. Write a C program to implement and demonstrate a binary search on a sorted array.
5. Write a C program to implement and demonstrate the Bubble sort algorithm.
6. Write a C program to implement and demonstrate the Selection sort algorithm.
7. Write a C program to implement and demonstrate the Insertion sort algorithm.
8. Write a C program to implement and demonstrate the Quick sort algorithm.
9. Write a C program to implement and demonstrate the Merge sort algorithm.
10. Write a C program to implement a stack using an array and perform basic stack operations: push, pop, and display.
11. Write a C program to implement a queue using an array and perform basic queue operations: enqueue, dequeue, and display.
12. Write a C program to implement a singly linked list and perform insertion, deletion, and traversal operations.
13. Write a C program to implement a binary tree and perform in-order, pre-order, and post-order traversal using recursion.
14. Write a C program to implement a binary search tree (BST) and perform insertion, deletion, and search operations.
15. Write a C program to represent a graph using an adjacency matrix and perform a depth-first search (DFS).
16. Write a C program to represent a graph using an adjacency list and perform a breadth-first search (BFS).
17. Write a C program to demonstrate basic file operations: open, read, write, and close.

Any other Practicals/Lab Assignments/Programsassigned by the teachers.

Third Semester

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Operating Systems**	**Course Code**	**24BCA403DS01**
Hours/Week	**4 (4+0+0)**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

This objective of this course is to enable students to learn about important concepts related to Operating Systems. It will help the students to enrich their knowledge and understanding of major functions performed by Operating System. It will give indepth knowledge to students covering process management, memory management, secondary storage structure, file management and Input/Output management.

Course Outcomes:

By the end of the course, the students will be able to:
CO1: Understand the basic concepts of Operating Systems and its classification.
CO2: Learn about the functions and structures of Operating system and relevance of System Calls and Services.
CO3: Understand the major: functions of Operating System.
CO4: Learn about different Scheduling algorithms.
CO5: Learn about the concept of controlled access to system resources through authentication and disk scheduling.

Unit – I

Introduction to Operating Systems: Objectives and Characteristics. Classification: Batch, Multi-programming, Multi-processing, Multi-tasking, Time-sharing, Distributed, Network and Real time Operating systems. System Calls and Services.

Functions and Structures: Operating System Functions- Process management, Memory management, Secondary storage management, I/O management, File management, Protection and Security. Structures- Simple Structure, Monolithic structure, Layered approach, Microkernel, Exokernel and Virtual Machines.

Unit – II

Process Management and Scheduling: Process concept- Process State Model, Process Control Block and Threads. Process Scheduling- Scheduling Queues, Schedulers and Context Switch. Operations on Processes, Cooperating processes and Inter-Process Communication.

Process Scheduling: Scheduling Criteria, Scheduling Algorithms: Single Processor Scheduling: FCFS, SJF, Round Robin, Multi Feedback Queue. Multiple Processor Scheduling and Real Time scheduling. Scheduling Algorithm Evaluation.

Unit – III

Memory Management: Concepts of Memory Management, Logical and Physical address space, Swapping, Memory allocation: Contiguous and Non-Contiguous. Paging: Hardware Support. Page Map Table and Protection. Segmentation: Hardware Support and Protection and Sharing.

Virtual Memory: Need of Virtual Memory, Demand paging, Pure Demand Paging. Handling page faults, Performance of Demand Paging. Page replacement Algorithms and Allocation of Frames: Allocation algorithms and Global vs Local Allocation. Thrashing.

Unit – IV

I/O Management: Basic I/O Devices, Types of I/O Devices: Block and Character Devices. I/O Software:

Device Independent I/O, User Space I/O and Kernel I/O Software. Device Controllers, Device Drivers and Interrupt Handlers. Communication Approaches to I/O Devices: Special Instruction I/O, Memory Mapped I/O and Direct Memory Access (DMA). Secondary Storage Structure: Disk Structure and Disk Scheduling Algorithms.

File System Interface: File Concept: Attributes, Operations and Types. File Access Methods: Sequential Access, Direct Access and Indexed Sequential. Free Space Management. Directory Structures: Single Level, Two level and Tree Structured. File Protection and Sharing.

Suggested Readings:

1. Silberschatz& Galvin: Operating System Concept, Wiley.
2. Milan Milenkovic: Operating Systems, Tata McGraw Hill.
3. William Stallings: Operating Systems, PHI.
4. YashawantKanetkar: Operating System Concepts, BPB.
5. H.M. Deitel, P.J Deitel: Operating Systems, Pearson .
6. A.S. Tanenbaum: Modern Operating Systems, Pearson/PHI.
7. Dhamdhere: Operating Systems, Tata McGraw Hill.
8. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	
Name of the Course	**Object-Oriented Programming using C++**	**Course Code**	**24BCA403DS02**
Hours/Week	**5 (3+0+2)**	**Credits (L: T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

The objective of this course is to introduce students to the principles and concepts of object-oriented programming (OOP) using the C++ programming language. The course aims to develop students' skills in designing and implementing object-oriented solutions to real-world problems.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the basic concepts of object-oriented programming.
CO2: Design and implement C++ programs using classes, objects, and inheritance.
CO3: Apply polymorphism and templates to develop reusable code.
CO4: Learn about the concept of exception handling and working with files.
CO5: Utilize advanced features of C++ to develop efficient and modular programs.

Unit – I

Introduction to OOP concepts: Procedural Vs. Object- Oriented Programming, Principles of OOP and their benefits. Object, classes, Inheritance, Abstraction, Encapsulation Polymorphism, Dynamic Binding, Message passing.
C++ Programming Basics: Syntax and structure of C++ programs, Data types, variables, and constants in C++, Control structures: decision making and looping constructs

Unit – II

Classes and objects: Defining and using classes and objects, Member functions and data members, Access specifiers: public, private, protected, Functions and parameter passing in C++, Arrays and strings in C++, Pointer, Constructors and destructors.
Inheritance: Derived class and Base class, Types of inheritance: single, multiple, multilevel, hierarchical, Access control in inheritance.

Unit – III

Polymorphism: function overloading, Operator overloading, Virtual functions and dynamic polymorphism, Abstract classes and pure virtual functions, Encapsulation and data hiding, Friend functions, static function.

Memory Management: Dynamic Memory Allocation: new, delete, Object Creation at run time.

Unit – IV

Exception handling: Throwing, Catching, Re-throwing an exception, specifying exception: processing unexpected exceptions; try-catch blocks, Exception propagation, Templates: class and function templates, Standard Template Library: benefits of STL and generic programming

Working with Files: Stream Classes, File input and output Operations in C++, Error handling during file operations.

Suggested Readings:

1. BalaGuruswamy : Object Oriented Programming and C++, THM
2. Rajaram : Object Oriented Programming and C++, New Age International.
3. Herbert Schildt : The complete Reference, 4th Edition, McGraw Hill Publications.

4. Shah &Thakker: Programming in C++, ISTE/EXCEL.
5. Subburaj : Object Oriented Programming with C++, Vikas.
6. Samanta : Object Oriented Programming with C++ & JAVA,PHI.
7. Stanley B. Lippman, Jose Lajoie, and Barbara E.Moo : C++ Primer.
8. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Programs/Lab Assignments

1. Create a C++ program to take two numbers as input from the user and display their sum.
2. Write a C++ program to demonstrate decision-making constructs like if-else and looping constructs like for and while loops.
3. Write a C++ program to define a class called `Rectangle` with attributes `length` and `width`, and display the area of the rectangle.
4. Implement a C++ program to demonstrate the concept of inheritance by creating a base class `Shape` and derived class `Rectangle`. Display the area of the rectangle using inheritance.
5. Create a C++ program to define a class called `Student` with attributes `name` and `roll number`. Use member functions to input and display student details.
6. Write a C++ program to demonstrate the use of constructor and destructor in a class.
7. Implement a C++ program to showcase the use of access specifiers (`public`, `private`, `protected`) in a class.
8. Create a C++ program to demonstrate function overloading by defining multiple functions with the same name but different parameters.
9. Write a C++ program to demonstrate dynamic polymorphism using virtual functions.
10. Write a C++ program to demonstrate the working of friend function.
11. Implement a C++ program to demonstrate the use of pointers to objects. Define a class `Book` with attributes `title` and `author`, and use pointers to access and display book details.
12. Write a C++ program to handle exceptions using `try-catch` blocks.
13. Create a C++ program to implement a simple template function to find the largest amomg two numbers. Test the function with different data types.
14. Write a C++ program to perform file input and output operations, including opening, reading, writing, and closing files.
15. Implement error handling during file operations in a C++ program, by handling exceptions and error codes.

Any other Practicals /Lab Assignments/Programs assigned by the teachers.

NameoftheProgram	**4-Year Bachelor of Computer Applications**	**ProgramCode**	**-----**
Name of the Course	**Database Management System**	**Course Code**	**24BCA403DS03**
Hours/Week	**5 (3+0+2)**	**Credits(L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

The objective of this course is to teach the student concepts related to database, database design techniques, transaction management, crash recovery and backup.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Learn about basics of database and database management system.
CO2: Understand about data models and file organization.
CO3:Indepth understanding of relational data models, databases and normalization process.
CO4: Learn about transaction processing and concurrency control techniques.
CO5: Understanding about relevance of recovery and backups.

Unit – I

Database Management System: Introduction, Database System Applications, History of Database Systems, Database System Vs. File Processing System, View of Data, Data Abstraction, Instances and Schemas. DBMS Environment, Database languages, Database Models.

Database design and ER Model: Physical, Conceptual and Logical Database design, Entity- Relationship Model: Entities, Relationships, Representation of entities, attributes, Representation of relationship set, Generalization, Aggregation, Conceptual design with ER Model

Unit– II

Relational Model: Introduction to the Relational Model, Attributes, Domains, Tuples, Relations and their schemes, relation representation, Keys, relationship, relational operations, , Integrity Constraint Over relations,EnforcingIntegrityconstraints,Queryingrelationaldata,View:IntroductiontoViews,Destroying / altering Views.
Relational Algebra and Calculus: Relational Algebra & its operations, Relational calculus &its types, Power of Algebra and calculus.

Lab Problem(s):*Creation and Querying relational data with SQL*

Unit–III

Normalization: Schema Refinement, Problems caused by redundancy, Decomposition & its properties; Normalization: First, Second, Third Normal forms, BCNF, Multivalued Dependencies,Join Dependencies.

Transaction Management & Concurrency Control: ACID properties, Transactions and Schedules, Concurrent execution of transaction, Serializability and Recoverability, Lock based Concurrency control, Lock Management, Lock Conversion, Dealing with deadlocks, Concurrency without Locking.

Unit– IV

Crash Recovery and Backup: Failure classifications, storage structure, Recovery & Atomicity, Log base recovery, Recovery with concurrent transactions, Failure with loss of nonvolatile storage, Database backup & recovery from catastrophic failure, Remote Backup System.

Storage and File Organization : Overview of physical storage media, Storage access; File organization, Operations on Files, Serial Files, Sequential Files , Index-Sequential Files, Direct Files.

Suggested Readings:

1. Ramez Elmasri, Shamkant B. Navathe: Fundamentals of Database Systems, Pearson.
2. Silberschatz Abraham : Database System Concept, Tata McGraw Hill, Latest edition.
3. C. J Date : Introduction to Database Systems, Pearson Education, Latest edition.
4. T.M. Connolly, Database Systems, Pearson Education.
5. P. Rob and C. Coronel, Database Systems: Design, Implementation, and Management, Thomson.
6. Krishnan Ram and Gehrke : Database Management System, Tata McGraw Hill.
7. ByrossIvan : Oracle 10 G The Database with HTML Database, BPB publication.
8. Any other book(s) covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Programs/Lab Assignments

1. Write SQL queries for DDL commands: Create, Desc, Alter, Rename, Drop.
2. Write SQL queries by using DML commands: Insert, Select, Update, Delete.
3. Write SQL queries using Logical operations: AND, OR, NOT, IN, BETWEEN, LIKE.
4. Write SQL queries using Arithmetic operators: +, -, *, /, %.
5. Write SQL demonstrating the use of Comparison operators: =, <>, >, >=, <, <=.
6. Write SQL queries using Aggregate functions: AVG, MIN, MAX, SUM, COUNT.
7. Write SQL queries to get current date and time: NOW, CURDATE, CURTIME, LAST_DAY, DATE_FORMAT.
8. Write SQL queries using Character functions: ASCII, CHAR, CONCAT, INSTR, INSERT, LENGTH, LOWER, UPPER.
9. Write SQL queries using Number functions: POWER, ROUND, SQRT, EXP, GREATEST, LEAST, MOD, FLOOR, CEIL.
10. Write SQL queries for Relational algebra operations: SELECT, PROJECT, UNION, INTERSECTION, DIFFERENCE.
11. Write SQL queries to perform table joins: INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN.
12. Write SQL queries to create, update, and delete views: CREATE VIEW, UPDATE VIEW, DROP VIEW.
13. Write SQL queries for subqueries and nested queries.
14. Write SQL queries to create and execute procedures.
15. Write SQL queries to create and manage triggers: CREATE TRIGGER, DROP TRIGGER.

Any other Practicals /Lab Assignments/Programs assigned by the teachers.

Fourth Semester

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Computer System Architecture**	**Course Code**	**24BCA404DS01**
Hours/Week	**4 (4+0+0)**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

Upon completion of this course, students will have better understanding of basic operations, principles, and components of Computer Architecture. Students will learn about different types of registers, instructions and interrupts used in Computer System. This course will empower the students to understand complex concepts related to computer architecture with ease such as addressing modes, types of memory etc.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the basic Computer concepts such as its different functional units and bus structure.
CO2: Interpret the functional architecture of Computer system in terms of its different components.
CO3: Analyze different types of Instructions and Interrupts used in Computer system..
CO4: Identify and compare different methods for computing Input/Output.
CO5: Explore Memory organization and different operations related to it.

Unit – I

Basic Computer Concepts: Computer Types: Analog and Digital, Digital Computer Organization and Architecture: Functional Units, Basic Organizational Concepts, Von-Neumann Architecture, Bus Structure. Data representation: Number Systems, Fixed and Floating point representation.

Register Transfer and Microoperations: Basic concepts and types of Registers, Register Transfer Language, Data Transfer between Registers, Bus and Memory Transfer, Microoperations: Arithmetic, Logic and Shift Microoperations.

Unit – II

Basic Computer Design: Instruction codes, Common Bus System Architecture, Computer Instructions: Instruction Set and Instruction Cycle. Types of Instructions: Register Reference, Memory Reference and Input-Output Reference.

Interrupts: Classifications of Interrupts: Maskable and Non maskable, Hardware and Software. Interrupt Service Routine, Context Switching, Interrupt Identification: Daisy Chaining, Polling and Vectored Interrupt. Interrupt Cycle.

Unit – III

Central Processing Unit: Introduction, General Register Organization, Stack Organization: Register/memory Stack, Reverse Polish Notation. Instruction Formats: Three Address, Two Address, One Address and Zero Address.

Data Transfer and Manipulation: Addressing Modes: Implied, Immediate, Register Direct and Indirect Mode and Direct Address Mode. Data Transfer and Data Manipulation Instructions, Program Control, Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC).

Unit – IV

Memory Organization: Memory Hierarchy, Main Memory Technologies, Auxiliary Memory, Associative Memory: Hardware requisites, Working Principle and Operations. Cache Memory: Characteristics, Types of Mapping, Writing into Cache, Cache Coherence.

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer: Programmed I/O and Interrupt Initiated I/O. Concepts related to Priority Interrupt: Daisy Chaining Priority and Parallel Priority Interrupt. Direct Memory Access (DMA): DMA Controller and DMA Transfer.

Suggested Readings:

1. Mano, M.M.: Computer System Architecture, Prentice-Hall of India.
2. Stallings, William: Computer Organisation & Architecture, Pearson Education.
3. Gill Nasib Singh and Dixit J.B: Digital Design and Computer Organisation, University Science Press (Laxmi Publications), New Delhi.
4. Mano, M.M.: Digital Logic and Computer Design, Prentice-Hall of India.
5. Hwang Kai: Advanced Computer Architecture, McGraw Hill International.
6. Hayes J. P: Computer System Architecture & Organization, McGraw-Hill Education.
7. Hamacher V. Carl: Computer Organization, McGraw Hill.
8. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Data Communication & Computer Networks**	**Course Code**	**24BCA404DS02**
Hours/Week	**4 (4+0+0)**	**Credits(L:T:P)**	**4:0:0**
Max. Marks.	**Theory:100(70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

The objective of this course is to inculcate knowledge in students about computer networks and data communication paradigms, about network models and standards, network protocols and their use, wireless technologies.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand basics of data communication and computer networks.
CO2: Learn about various network models used in networking.
CO3: Learn basics and categories of various transmission media.
CO4: Learn various routing techniques and congestion control techniques.
CO5: Understand the working of different applications.

Unit – I

Introduction: Data Transmission concepts, Transmission Media:Magnetic, twisted pair, coaxial cable, fiber optics, wireless transmission (radio, microwave, infrared), transmission impairments, switching, modulation, multiplexing.

Network: Basic concepts of network, Network topology & Protocols, types of network: LAN, MAN, WAN, Wireless networks, Inter-networks, network layer, interfaces and services.

Reference Models: OSI, TCP/IP and their comparison

Unit– II

Data Link Layer : Framing, Error control, Sliding window protocols (one bit, Go back n, selective repeat). DLL Protocols–HDLC, PPP.

Medium Access Sub layer : Channel Allocation, MAC protocols – ALOHA, CSMA protocols, Collision free protocols, Limited Contention Protocols, Wireless LAN protocols, IEEE 802.3, 802.4, 802.5 standards and their comparison.

Unit–III

Network Layer: Design issues, Routing algorithms (shortest path, flooding, flow based, distance vector, hierarchical, broadcast, multicast, for mobile hosts), Congestion control algorithms (Leaky bucket, Token bucket, Choke Packet, Load shedding), Internetworking, IP Protocol, ARP, RARP.

Unit– IV

Transport Layer: Addressing, establishing and releasing connection, flow control, buffering, Internet Transport Protocol (TCP and UDP).

Application Layer: Domain name system, E-mail, File transfer protocol, HTTP, HTTPS,TELNET.

SuggestedReadings:

1. Tanenbaum, Andrew S.: Computer Networks (4th Edition), PHI.
2. Gill Nasib Singh: Handbook of Computer Fundamentals, Khanna Book Publishing

Company(Pvt.)Limited, New Delhi.
3. Forouzan, B. A.: Data Communications and Networking, Fourth Edition, Tata McGraw Hill.
4. Douglas E. Comer: Internet Working with TCP/IP (Vol.1, 4th Edition), CPE.
5. Stallings, William : Data and Computer Communications (8th Edition), PHI.
6. Nance, Bary: Introduction to Networking, PHI, 4th Edition.
7. DoerrAlan&LevasseurKenneth:AppliedDiscreteStructuresforComputerScience,GalgotiaPub.Pvt.Ltd.
8. Anyother book(s)coveringthe contentsofthesubject.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**ProgramCode**	**-----**
Name of the Course	**Java Programming**	**Course Code**	**24BCA404DS03**
Hours/Week	**6 (2+0+4)**	**Credits(L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

The objective of this course is to enable the students to equip with the skills and knowledge to design, develop, and deploy Java applications, leveraging the language's powerful features and extensive libraries to solve complex programming challenges.

Course Outcomes:

By the end of the course the students will be able to:

CO1: Gaining an understanding about Java Programming and its constructs.
CO2: Learning to develop proficient Java Programs.
CO3: Understanding how to apply Object-Oriented principles.
CO4: Implementing and Utilizing Java Standard Libraries in developing Java applications.
CO5: Developing GUI and Multithreaded Applications.

Unit – I

Introduction to Java: Java Features, Java Virtual Machine (JVM), Byte code, Java API, Java Development Kit (JDK), Garbage Collection. Language Basics: Keywords, Constants, Variables and Data Types, Operators and Expressions, Decision Making, Branching and Looping.

Introducing Classes, Objects and Methods: Defining a Class, Methods Declaration, Creating Objects and accessing Class members, Constructors, Methods Overloading, Wrapper Classes, Inheritance, Methods Overriding, Final Class, variables and methods, Abstract Class and Methods, Interfaces.

Unit– II

Arrays, Strings andVectors: Creating and using Arrays, String operations, String Buffer, String builder, and String Tokenizer class, Vector class.

Packages and Exceptions: Java API packages, Creating and using packages, static import, Exceptions handling, Types of Exceptions, multiple catch statements, 'throw' and 'throws', using 'finally' statement, Creating your own exceptions.

Unit–III

Multithreaded Programming: Single threaded and multi-threaded program, Creating threads using Thread class, Life cycle of a Thread, Stopping and blocking a Thread, getting and setting the Thread Priority, Synchronization, implementing the Runnable interface.

Managing Input/Output Streams: Concept to fstreams, Byte and Character streams, Reading and Writing from Console and Files. Input output exceptions.

Unit– IV

Applet Programming: How Applets differs from Java Application, Applet Life Cycle, APPLET Tag, Running an Applet, Passing Parameters to Applet.

Event Handling: Mechanism, The Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and inner classes.

GUI Programming: Working with Frame Window, Graphics and Text, AWT Controls and classes. Layout Managers, working with Menus.

Suggested Readings:

1. E.Balagurusamy: Programming with Java: A Primer, McGraw Hill.
2. Herbert Schildt: Java:The Complete Reference, McGraw Hill.
3. Bruce Eckel: Thinking in Java, Prentice Hall.
4. Cay S. Horstmann, Gary Cornell: Core Java Volume I—Fundamentals, Prentice Hall.
5. Anyother book(s)coveringthe contentsofthesubject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Programs/Lab Assignments

1. Write a Java program to define a class `Student` with data members for name and age. Include member functions to input and display the details of a student.
2. Create a Java program to demonstrate method overloading by creating multiple methods with the same name but different parameters in a class
3. Create a Java program to demonstrate method overriding by creating a base class `Animal` and a derived class `Dog` that overrides a method from the base class.
4. Implement a Java program to define an abstract class `Shape` with an abstract method `draw()`. Create derived classes `Circle` and `Square` that implement the `draw()` method.
5. Implement a Java program to demonstrate the use of the `Vector` class by performing operations like adding, removing, and displaying elements of a vector.
6. Write a Java program to create a user-defined package and use it in another class.
7. Create a Java program to handle multiple exceptions using try-catch blocks.
8. Implement a Java program to demonstrate the use of the `finally` statement in exception handling.
9. Write a Java program to create and throw a custom exception.
10. Create a Java program to implement a multithreaded application by extending the `Thread` class.
11. Implement a Java program to demonstrate thread synchronization.
12. Create a Java program to read and write data to a file using `FileInputStream` and `FileOutputStream`.
13. Write a Java applet to display "Hello, World!".
14. Create a Java applet to demonstrate the use of the `APPLET` tag and pass parameters to the applet.
15. Create a Java program to demonstrate the use of AWT controls like buttons, labels, and text fields.

Any other Practicals/Lab Assignments/Programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Computer Graphics**	**Course Code**	**24BCA404DS04**
Hours/Week	**5 (3+0+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

The objective of the Computer Graphics course is to provide students with a comprehensive understanding of the fundamental principles and techniques of computer graphics. Students will learn to create and manipulate visual content, explore 2D and 3D modelling, rendering, and animation, and gain proficiency in using graphics software and programming languages to develop visually compelling applications and simulations. The course aims to equip students with the skills necessary to apply computer graphics in various fields such as gaming, simulations, and data visualization.

Course Outcomes:
By the end of the course the students will be able to:

CO1: Understand the fundamental concepts and principles of computer graphics.
CO2: Develop proficiency in 2D and 3D graphics programming using appropriate tools and libraries.
CO3: Apply graphical techniques to solve real-world problems in areas such as gaming, simulation, and visualization.
CO4: Analyze and critique graphical algorithms and techniques for efficiency and effectiveness.
CO5: Demonstrate creativity and innovation in designing and implementing graphical applications.

Unit – I

Introduction to Computer Graphics: Overview of computer graphics and its applications, Historical development, Application areas e.g. entertainment, scientific visualization, user interfaces, Basics of raster and vector graphics, differences between raster and vector graphics, Representation and storage formats, Graphics pipeline and rendering process, stages of the graphics pipeline (modeling, transformation, rendering), concepts of rendering primitives, vertices, and fragments.

Graphics primitives: Coordinate systems, types of primitives (points, lines, polygons), Cartesian and homogeneous coordinate systems, transformation matrices and their applications

Unit – II

2D Graphics Programming: Pixel operations and drawing algorithms, color models (RGB, CMYK, HSL), pixel operations (blending, interpolation), Line drawing algorithms (Digital differential analyzer algorithm, Bresenham's line drawing algorithm), Circle drawing algorithms (Midpoint circle algorithm, Bresenham's circle drawing algorithm).

2D Transformations: translation, rotation, scaling in 2D space, mirror reflection basic and advance problems involving composite operations.

Windowing techniques: Basic terminologies-window, viewport, clipping window, region codes.

Unit – III

Clipping techniques: Mathematics of point clipping, line clipping (midpoint subdivision method, Cohen-Sutherland line clipping algorithm), polygon clipping (Sutherland-Hodgman polygon clipping algorithm)

3D Graphics Programming: 3D transformations, Translation, Rotation, Scaling in 3D space, Homogeneous coordinates and transformations.

Projection techniques: Orthographic projection, Perspective projection and vanishing points

Unit – IV

Lighting and shading models: Phong reflection model, Gouraud and Phong shading techniques

Hidden surface removal algorithms: Z-buffer algorithm, Scan line algorithm for hidden surface removal

Texture mapping and image rendering: Texture coordinates and mapping techniques, Texture filtering and mipmapping.

Suggested Readings:

1. J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, and J. D. Foley: Computer Graphics: Principles and Practice, Addison-Wesley.
2. D. Shreiner, G. Sellers, J. M. Kessenich, and B. M. Licea-Kane: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Addison-Wesley.
3. D. Hearn and M. P. Baker: Computer Graphics with OpenGL, Pearson.
4. E. Angel and D. Shreiner: Interactive Computer Graphics: A Top-Down Approach with WebGL, Addison-Wesley.
5. F. Klawonn: Introduction to Computer Graphics: Using Java 2D and 3D, Springer.
6. Any other book(s) covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Programs/Lab Assignments

1. Implement a program to define and use basic graphics primitives such as points, lines, and polygons.
2. Implement a program to draw a line using the Digital Differential Analyzer (DDA) algorithm.
3. Create a program to draw a line using Bresenham's line drawing algorithm.
4. Write a program to draw a circle using the Midpoint Circle algorithm.
5. Implement a program to draw a circle using Bresenham's circle drawing algorithm.
6. Create a program to perform 2D transformations: translation, rotation, and scaling on a given set of points.
7. Write a program to perform mirror reflection of a 2D shape across the x-axis and y-axis.
8. Create a program to draw basic two-dimensional objects like rectangles, triangles, and polygons using inbuilt functions..
9. Write a program to implement the Cohen-Sutherland line clipping algorithm to clip a line within a rectangular clipping window.
10. Write a program to apply a shearing transformation to a 2D object (like a square or triangle) and display the original and transformed objects.
11. Write a program to apply various coloring techniques to 2D pictures.
12. Write a program to implement line clipping using the Cohen-Sutherland line clipping algorithm.
13. Write a program to perform 3D transformations: translation, rotation, and scaling on a set of 3D points.
14. Implement a program to demonstrate simple animations using transformations such as translation, rotation, and scaling.
15. Write a program to apply various coloring techniques to 3D object.

Any other Practicals/Lab Assignments/Programs assigned by the teachers.

Fifth Semester

Name of the Program	**4-year Bachelor of Computer Applications- Single Major**	**Program Code**	**----------**
Name of the Course	**Software Engineering**	**Course Code**	**25BCA405DS01**
Hours/Week	**4 (4+0+0)**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

This Course is intended to provide the students with an overall view over software engineering discipline and with insight into the processes of software development. The students know about the requirements, design, coding and testing concept of software development. The students also understand the concepts of software maintenance and software quality attributes and standards.

Course Outcomes:

By the end of the course the students will be able to:

CO1: Analyse and resolve software crisis issues in the development of software system.
CO2 Develop the software system with low cost, high quality and within the given time frame.
CO3: Know about software design process and design methodologies.
CO4: Learn various software testing level and software project management activities.
CO5: Learn software maintenance, software quality attributes, software quality assurance and standards

Unit – I

Introduction to Software Engineering: Software and its Characteristics, Evolving Role of Software, Software Product, Software Processes, Software crisis, Software engineering Approach and Challenges, Principles of software engineering, Software development process models with comparison: Waterfall, Prototype, Iterative enhancement model, Time boxing, Spiral Model, RAD Model and Object-Oriented Models.

Software Project Management: Management activities, Project planning, Project scheduling, and Risk management activities.

Unit – II

Software Requirements Engineering: Requirements Engineering Processes, Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

Software Requirements Analysis & Specifications: Software requirements, Structured analysis: DataFlow diagram, data dictionary. Object-oriented analysis, Software Requirement Specification (SRS):NeedofSRS,Characteristics ofSRS,ComponentsofSRS, StructureofSRS.

Software Metrics and Measures: Need and benefits of Software Metrics, Size Metrics: Line of code,Tokenmetrics,Functionpointmetrics,SoftwareProjectEstimationModels-COCOMO models.
.

Unit – III

Software Design: Basic Concept of Software Design, Architectural Design, Low-Level Design: Modularization, Design Structure Charts, Flow Charts, Coupling and Cohesion Measures; Design Strategies**:** Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design, User Interface Design

Coding: Goals of coding phase, Programming style, Structured programming: objectives of structured

programming, Principles of structured programming, advantages and disadvantages of structured programming.

Software Testing: Impracticality of Testing, Levels of Testing, Structural Testing (White Box Testing), Functional Testing (Black Box Testing).

Unit – IV

Software Maintenance: Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance; Software Re- Engineering, Reverse Engineering, Software Documentation.

Software Quality: Software quality attributes, Software Quality Assurance – plans & activities
SoftwareQuality Standards: ISO 9001,SEI-CMM.

Suggested Readings:

1. Gill, Nasib Singh: Software Engineering, Khanna Book Publishing Co.(P) Ltd, N. Delhi
2. Hoffer, George, Valacich, Modern System Analysis and Design. Pearson Education
3. Pressman S. Roger, Software Engineering, Tata McGraw-Hill.
4. JalotePankaj, An integrated Approach to Software, Engineering, Narosa Publishing House.
5. Sommerville Ian, Software Engineering, Addison Wesley
6. Jorgensen P. C., Software Testing-A Craftman's Approach, CRC Press.
7. Tom Gilb, Principles of Software Engineering Management, Addison-Wesley
8. Any other book(s) covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-year Bachelor of Computer Applications- Single Major**	**Program Code**	**----------**
Name of the Course	**Artificial Intelligence & Expert System**	**Course Code**	**24BCA405DS02**
Hours/Week	**4 (4 +0+0)**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

Introduce fundamental concepts, history, and applications of Artificial Intelligence (AI). Develop problem-solving skills using search algorithms, knowledge representation, and reasoning techniques. Provide foundational knowledge of machine learning and expert systems. Enable students to design and implement basic AI solutions for real-world problems. Highlight ethical considerations and societal impacts of AI technologies.

Course Outcomes:

By the end of the course the students will be able to:

CO1: Grasp core AI concepts and problem-solving techniques.
CO2: Utilize knowledge representations methods in AI systems.
CO3: Comprehend the role and development of expert systems.
CO4: explore perception, and learning concepts in AI.
CO5: Grasp the fundamental of Neural Network and Fuzzy Logic.

Unit – I

Introduction: Definition, Concepts and applications of Artificial Intelligence, Problem solving: Defining problem as State space search, Production systems, Problem characteristics, Search techniques: Brute force and Heuristic search and their different searching techniques.

Intelligent Agents: Agent types (simple reflex, model-based, goal-based, utility-based, learning agents). Agent environments (static/dynamic, observable/partially observable, deterministic/stochastic).

Ethical Considerations: Bias, privacy, job displacement, and societal impact.

Unit – II

Knowledge representation: Types of knowledge, Inferencing, Inference rule, Knowledge Representation: Logic based Knowledge representation, Rule based knowledge representation;

Knowledge representation: Non-Monotonic reasoning, Knowledge representation based on probability and uncertainty; Knowledge representation schemes: Formal logic, Semantic net, Frame, Scripts.

Unit – III

Expert System: Definition, Role of Knowledge in expert system, Architecture of Expert system.

Expert system development life cycle: Problem selection, Prototype construction, Formalization, Implementation, Evaluation, Knowledge acquisition and different acquisition techniques, Knowledge engineering, Cognitive behaviour.

Perception: Sensing, Speech recognition, Vision, Action.

Unit – IV

Learning, Planning and Understanding: Learning and its different types, Planning, understanding.

Neural Networks: Introduction, Comparison of artificial neural networks with biological neural networks, Learning in neural networks, Perceptions, Back propagation networks, application of neural networks.

Fuzzy logic: Definition, Difference between Boolean and Fuzzy logic, fuzzy subset, fuzzy membership function, fuzzy expert system, Fuzzy Inference system, fuzzy controller.

Suggested Readings:

1. Russell S. and Norvig P.: Artificial Intelligence: A Modern Approach, Prentice-Hall.
2. Elaine Rich, Kevin Knight and Nair: Artificial Intelligence, TMH.
3. Luger G. F. and Stubblefield W. A.: Artificial Intelligence: Structures and strategies for Complex Problem Solving, Addison Wesley.
4. Nilsson Nils J.: Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publishers Inc.
5. Patrick Henry Winston: Artificial Intelligence, Addison-Wesley Publishing Company.
6. M. Mitchell: An Introduction to Genetic Algorithms, Prentice-Hall.
7. J.S.R. Jang, C.T. Sun and E. Mizutani: Neuro-Fuzzy and Soft Computing, PHI, Pearson Education.
8. Davis E. Goldberg: Genetic Algorithms: Search, Optimization and Machine Learning, Addison Wesley.
9. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of Program	**4-year Bachelor of Computer Applications- Single Major**	**Program Code**	**----------**
Name of the Course	**Cloud Computing**	**Course Code**	**25BCA405DS03**
Hours/Week	**5 (3+0+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:

The objective of this course is to provide the learners the ability to understand the evolution, models, and architectures of cloud computing; analyze virtualization and distributed systems as they apply to cloud infrastructures; evaluate cloud security, privacy, and governance challenges in today's digital landscape; design and implement scalable, cost-efficient cloud solutions using modern tools and platforms; and critically assess emerging trends such as serverless computing, multi-cloud management, and edge computing.

Course Outcomes:

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

CO1: Understand cloud computing concepts and distinguish between IaaS, PaaS, and SaaS and select appropriate models for varied applications.
CO2: Develop scalable and resilient cloud architectures using virtualization and containerization techniques.
CO3: Apply best practices and frameworks to secure cloud environments, addressing data privacy and compliance issues.
CO4: Use orchestration tools (e.g., Docker, Kubernetes, OpenStack, etc) to automate deployment and resource management.
CO5: Analyze cutting-edge cloud technologies (e.g., serverless, edge computing) and integrate them into comprehensive cloud solutions.

Unit-I

Fundamentals of Cloud Computing
Definitions, history, and evolution of cloud computing. Overview of cloud service models: IaaS, PaaS, SaaS.

Deployment models: public, private, hybrid, and multi-cloud.

Key architectural principles: virtualization, elasticity, resource pooling, and scalability.

Review of leading cloud providers and market trends.

Unit-II

Cloud Infrastructure and Virtualization Technologies: Virtualization fundamentals: hypervisors (Type-1 and Type-2), virtual machines, and containers (Docker, Kubernetes).

Cloud networking: Software-defined networking (SDN), network virtualization, and load balancing.

Cloud storage solutions: distributed storage, object storage (e.g., Amazon S3, OpenStack Swift). Cloud orchestration and resource provisioning tools.

Emerging concepts: Serverless computing and Function as a Service (FaaS).

Unit-III

Cloud Security, Privacy, and Governance: Threats and vulnerabilities in cloud environments.

Cloud security fundamentals: Encryption, identity and access management (IAM), and secure communication protocols. Privacy challenges and data protection techniques in the cloud. Legal and regulatory frameworks (e.g., GDPR, HIPAA, CCPA, IT Act 2000, IT Rules 2011, etc) and their implications for cloud deployments. Governance, compliance, and cost management in cloud environments.

Unit-IV

Advanced Topics and Emerging Trends in Cloud Computing
Performance optimization and cost-efficiency in cloud systems. Big Data and Analytics in the cloud. Edge computing and IoT integration with cloud platforms. Cloud-native architectures and serverless computing.

Case studies and practical labs on deploying cloud-based applications.

Future trends: multi-cloud management, hybrid cloud strategies, and emerging research directions (e.g., quantum cloud computing).

Suggested Readings:

1. Anthony T. Velte Toby J. Velte, Robert Elsenpeter: Cloud Computing - A Practical Approach, McGraw-Hill.
2. ZaighamMahmood, Ricardo Puttini, Thomas Erl: Cloud Computing: Concepts, Technology & Architecture, Pearson
3. Michael J. Kavis: Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS), Wiley.
4. Kris Jamsa: Cloud Computing: SaaS, PaaS, IaaS, Virtualization and more, Jones & Bartlett Publishers.
5. Tim Mather, SubraKumaraswamy, ShahedLatif: Cloud Security and Privacy: An Enterprise Perspective on

Risks and Compliance, O'Reilly Media Inc.
6. Barrie Sosinsky: Cloud Computing Bible, Wiley-India.
7. Jason Venner, Pro: Hadoop, Apress.
8. RajkumarBuyya, James Broberg, Andrzej M. Goscinski: Cloud Computing- Principles and Paradigms, Wiley.
9. Nikos Antonopoulos, Lee Gillam: Cloud Computing: Principles, Systems and Applications, Springer.
10. Ronald L. Krutz, Russell Dean Vines: Cloud Security- A Comprehensive Guide to Secure Cloud Computing, Wiley-India.
11. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals /Lab Assignments

Each lab assignment is designed to span foundational topics, hands-on deployment and configuration, security, cost management, and emerging trends. These assignments are designed to provide practical experience aligned with the course units.

1. Exploring Cloud Service Models: Review and compare IaaS, PaaS, and SaaS using cloud provider demos and simulated environments.
2. Virtual Machine Deployment in the Cloud: Use a platform (e.g., OpenStack or AWS Free Tier or any other) to deploy, configure, and manage a virtual machine.
3. Containerization with Docker: Containerize a sample application, build Docker images, and run containers locally.
4. Orchestration with Kubernetes: Deploy a multi-container application on a Kubernetes cluster, configure pods, services, and scaling policies.
5. Cloud Networking Fundamentals: Set up virtual networks and subnets in a cloud environment (using AWS VPC, Azure Virtual Network, or similar).
6. Implementing Object Storage: Deploy and interact with an object storage solution (e.g., OpenStack Swift or Ceph or any other) to store and retrieve data.
7. Configuring Identity and Access Management (IAM): Set up IAM policies and roles in a chosen cloud platform, and test role-based access.
8. Cloud Security Best Practices: Configure encryption (in transit and at rest), establish security groups/firewall rules, and review audit logs.
9. Deploying a Private Cloud with OpenStack: Install and configure a basic OpenStack cloud in a lab (using virtualized or containerized components).
10. Cloud Auto-Scaling and Cost Optimization: Create auto-scaling groups, simulate load changes, and analyze cost implications using cloud cost calculators.
11. Cloud Migration Simulation: Use migration tools (or scripts) to move an on-premises application to a cloud environment.
12. Deploying Serverless Applications: Develop and deploy a serverless function (using AWS Lambda, Azure Functions, or Google Cloud Functions) to process a simple task.
13. Integrating Big Data in the Cloud: Deploy a cloud-based Hadoop or Spark cluster and run sample data processing jobs.
14. Multi-Cloud Management and Orchestration: Explore tools that enable management of resources across two cloud providers, and simulate integrated operations.
15. Exploring Edge Computing in a Cloud Context: Set up a simulated edge computing environment (using IoT devices or virtual edge nodes) to process data locally and sync with the central cloud.

Lab Tools/Software:

Students may use AWS, Azure, GCP, OpenStack or any other for cloud deployment; VirtualBox/VMware, Docker, and Kubernetes for virtualization/containerization; Terraform and Ansible for orchestration; along with security (OpenSSL), monitoring (Prometheus, Grafana), serverless platforms (AWS Lambda), and IoT/edge tools (Raspberry Pi, MQTT brokers) with IDEs like VS Code.

Note:*These lab assignments can be adapted or expanded or decided based on available resources.*

Any other Practicals/Lab Assignments given by the teachers.

Name of Program	**4-year Bachelor of Computer Applications- Single Major**	**Program Code**	**----------**
Name of the Course	**Web Application Development**	**Course Code**	**25BCA405DS04**
Hours Per Week	**6 (2+0+4)**	**Credits**	**2:0:2**
Maximum Marks	**Theory: 50 (35+15)** **Practical: 50 (35+15)**	**Time of Examinations**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Students will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

To provide students with the knowledge and skills to develop dynamic, secure, and scalable web applications. This course emphasizes client-server architecture, interactive interfaces, server-side scripting, database connectivity, and security practices while fostering innovative thinking and practical expertise in creating real-world web solutions.

Course Outcomes:

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

CO1: Understand web technologies and client-server architecture.
CO2: Build dynamic web applications using server-side scripting.
CO3: Implement authentication, authorization, and data validation.
CO4: Optimize web application performance and scalability.
CO5: Apply version control and testing tools in development workflows.

Unit – I

Introduction: Concept of WWW, Internet and WWW, HTTP Protocol: Request and Response, Web browser and Web servers, Features of the latest version of Web.

Web Design: Concepts of effective web design, Web design issues including Browser, Bandwidth and Cache, Display resolution, Look and Feel of the Website, Page Layout and linking, User-centric design, Sitemap, Planning and publishing website, Designing effective navigation.

Unit – II

HTML: Basics of HTML, formatting and fonts, commenting code, color, hyperlink, lists, tables, images, forms, XHTML, Meta tags, Character entities, frames and frame sets, Browser architecture and Web site structure. Overview and features of the latest version of HTML.

Stylesheets: Need for CSS, introduction to CSS, basic syntax and structure, using CSS, background images, colours and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS, CSS2, Overview and features of the latest version of CSS.

Unit – III

JavaScript: Client-side scripting with JavaScript, variables, functions, conditions, loops and repetition, Pop up boxes, Advance JavaScript: JavaScript and objects, JavaScript own objects, the DOM and web browser environments, Manipulation using DOM, forms and validations, DHTML: Combining HTML, CSS and JavaScript, Events and buttons.

XML: Introduction to XML, uses of XML, simple XML, XML key components, DTD and Schemas, Using XML with application. Transforming XML using XSL and XSLT.

Unit – IV

PHP: Introduction and basic syntax of PHP, decision and looping with examples, PHP and HTML, Arrays,

Functions, Browser control and detection, string, Form processing, Files, Advance Features: Cookies and Sessions.

PHP and MySQL: Basic commands with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names, creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, PHP my admin and database bugs.

Suggested Readings:

1. Nixon, R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. O'Reilly.
2. Flanagan, D. JavaScript: The Definitive Guide. O'Reilly.
3. Bangia, R. Web Development and Design Foundations with HTML5. Pearson.
4. Thomas A. Powell, Web Design: The Complete Reference, Fourth Edition, Tata McGraw-Hill.
5. Deitel and Goldberg, Internet and World Wide Web: How to Program, PHI.
6. Matthew A. Russell, Mining the Social Web, Second Edition, O'Reilly Media.
7. Jon Duckett, HTML and CSS: Design and Build Websites.
8. A. Weiss, "Complete Reference Guide to JavaScript," QUE.
9. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Programs/Lab Assignments

1. Build a responsive webpage with a form that includes input validation for user data.
2. Create a simple REST API using Node.js and Express that handles GET and POST requests.
3. Develop a basic blogging website with pages for adding, viewing, and editing blog posts.
4. Write a JavaScript program to dynamically load content from an external API into a webpage.
5. Create a simple contact form using HTML, CSS, and JavaScript that sends data to a backend server.
6. Write a program to Build a weather application that fetches weather data from an external API and displays it.
7. Write a JavaScript program to implement infinite scrolling on a webpage.
8. Create a CRUD web application where users can create, read, update, and delete records from a database.
9. Build a form with validation that stores user input in a database (using Node.js and MongoDB).
10. Write a program to implement user authentication using OAuth 2.0 for logging into a web application.
11. Create a to-do list web application that allows users to add, remove, and mark tasks as completed.
12. Write a program to implement a search functionality that filters results based on user input.
13. Build a responsive navigation menu with dropdowns using HTML, CSS, and JavaScript.
14. Write a program to store user preferences using cookies or local storage in the browser.
15. Create a dynamic quiz web application that stores questions and answers and tracks user scores.

Any other Practicals/Lab Assignments/Programs assigned by the teachers.

Sixth Semester

Name of Program	**4-year Bachelor of Computer Applications- Single Major**	**Program Code**	**----------**
Name of the Course	**Internet of Things**	**Course Code**	**25BCA406DS01**
Hours/Week	**5 (3+0+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objective:

The objective of this course is to provide the learners the ability to understand the fundamental principles and practical applications of IoT. This course aims to sensitize the students about IoT architecture, embedded systems, communication protocols, data management, cloud and edge integration, as well as security and privacy challenges. Students will gain hands-on experience through labs and projects while examining current trends and emerging technologies.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand IoT architectures, protocols, and enabling technologies.
CO2: Design IoT systems integrating sensors, actuators, and communication modules.
CO3: Implement IoT protocols (MQTT, CoAP) and networking standards (Zigbee, LoRaWAN).
CO4: Develop IoT applications using cloud platforms and edge computing.
CO5: Evaluate security threats and privacy-preserving techniques in IoT deployments.

Unit-I

Fundamentals of IoT and Embedded Systems
Introduction to IoT: Definition, history, evolution, characteristics of IoT and key concepts. IoT architecture layers (perception, network, middleware, application). Overview of IoT connectivity (wired and wireless).

Enabling technologies: RFID, WSN, cloud/edge computing. Embedded systems basics: Sensors, actuators, and microcontrollers. IoT vs. M2M, societal and economic impacts.

Unit-II

Communication Protocols and Networking for IoT Protocol stack: MQTT, CoAP, HTTP/HTTPS, AMQP. Wireless standards: Zigbee, BLE, LoRaWAN, 5G. Wireless communication techniques and challenges.
Network topologies and data transmission in IoT. IP-based IoT (6LoWPAN, Thread). Integration with cloud networks.

Unit-III

IoT Devices, Data Management, Cloud Integration, and Analytics: Sensors (temperature, motion), actuators, microcontrollers (Arduino, Raspberry Pi). Data acquisition, storage, and preprocessing in IoT.

Cloud integration: IoT data ingestion and processing using AWS IoT/Azure IoT Hub/Google Cloud IoT/Things Board, etc. Edge computing fundamentals for real-time data processing. Data analytics and visualization for IoT data.

Unit-IV

Security, Privacy, and Emerging Trends in IoT: IoT security challenges: Threats, vulnerabilities, data breaches and attack vectors.

Privacy issues in IoT: Encryption, authentication, data protection and anonymization techniques. Regulatory and compliance frameworks (e.g., GDPR, CCPA, etc.) in an IoT context. **Emerging trends:** Blockchain for IoT, AI-based analytics, and next-generation IoT platforms.

Case studies: Smart homes, healthcare, precision agriculture, etc.

Suggested Readings:

1. ArshdeepBahga and Vijay Madisetti: Internet of Things: A Hands-On-Approach, Elsevier.
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete: IoT Fundamentals: Networking Technologies, Protocols, and Use Cases, Cisco Press.
3. Alice James, Ajith Abraham: IoT System Design: Project Based Approach, Springer.
4. Claire Rowland, Elizabeth Goodman, Martin Charlier, Ann Light, and Alfred Lui: Designing Connected Products: UX for the Consumer Internet of Things, O'Reilly Media.
5. RajkumarBuyya and Amir VahidDastjerdi: Internet of Things: Principles and Paradigms, Elsevier.
6. Shancang Li and Li Da Xu: Cybersecurity for the Internet of Things, Springer.
7. Raj Kamal: Internet of Things-Architecture and design principles, McGraw Hill Education.
8. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, StamatisKarnouskos, David Boyle: From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, Academic Press.
9. Peter Waher, Learning Internet of Things, PACKT publishing, BIRMINGHAM – MUMBAI
10. Bernd Scholz-Reiter, Florian Michahelles: Architecting the Internet of Things, Springer.
11. Daniel Minoli: Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications, Wiley Publications.
12. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practical/Lab Assignments

This lab will provide students with hands-on exposure to IoT device setup, data communication, cloud integration, security, and end-to-end system design.

1. Install IDE of Arduino and write a program using Arduino IDE to blink LED.
2. Interface LED and buzzer with Arduino to buzz for a period of time.
3. Interface RGB LED with Aurdino to obtain different colours and brightness using PWM.
4. a) Control a servo motor using Arduino with an input given through a push button (e.g: When the push button is pressed the servo motor has to rotate by 15 degrees). b) Rotate Stepper motor either clockwise or anti clockwise at 'n' number of steps using Arduino.
5. Write a program to read the data from the RFID tag and display the information on the display board using Arduino and control LED (e.g: if it is a valid card then the LED should be ON otherwise OFF).
6. Control any two actuators connected to the Arduino using Bluetooth/Wifi.
7. Interface analog/digital sensors with Arduino and analyse the corresponding readings. (Sensors like temperature, alcohol, humidity, pressure, gas, sound pollution, level, weight, flow, proximity, LDR, PIR, pulse, vibration, sound etc.)
8. IoT Device Setup: Assemble and program a Raspberry Pi/Arduino with basic sensors (temperature, humidity).
9. Sensor
10. Data Acquisition: Interface various sensors with a microcontroller; capture and display sensor readings.
11. Local Communication Protocols: Establish serial, I2C, or SPI communication between IoT devices.
12. MQTT Messaging: Set up an MQTT broker (e.g., Mosquitto) and implement publish/subscribe for sensor data.
13. CoAP and RESTful APIs: Develop a simple REST/CoAP API to expose sensor data from a microcontroller.
14. Data Logging: Log IoT data to a local or cloud-based database (e.g., InfluxDB).

15. Edge Processing: Configure an edge device (e.g., Raspberry Pi) to preprocess/filter sensor data before transmission.
16. Cloud Integration: Connect IoT devices to a cloud platform (AWS IoT Core or Azure IoT Hub) for remote monitoring.
17. Data Visualization: Build a real-time dashboard using Node-RED or Grafana to visualize collected data.
18. IoT Application Development: Write custom applications in Python to control and interact with IoT devices.
19. IoT Security Implementation: Secure device communication using TLS/SSL and configure secure MQTT messaging.
20. Over-the-Air (OTA) Updates: Implement a basic OTA update mechanism on an ESP8266/ESP32 platform.
21. Mobile Integration: Create a simple mobile interface (using MIT App Inventor or similar) to control IoT devices.
22. Smart Home Prototype: Develop a prototype that integrates sensors, actuators, and cloud services to automate home functions.

Lab Tools/Software:

Hardware: Arduino Uno, ESP32, Raspberry Pi 4, sensors (DHT11, PIR), LoRa modules, relays or others
Software: Arduino IDE, Python, Visual Studio Code, AWS IoT Core, Mosquitto, Node-RED, InfluxDB, Grafana, Wireshark, or any other.
Cloud Platforms: AWS IoT, Azure IoT Hub, Google Cloud IoT, OpenStack, Things Board (open-source) or any other.
Simulation: TinkerCAD Circuits for prototyping without hardware or any other.
Data Storage/Visualization: InfluxDB, Grafana, cloud monitoring tools
Security: OpenSSL for TLS/SSL certificate management

Note:*These lab assignments can be adapted or expanded or decided based on available resources.*

Any other Practicals/Programs/Lab Assignments given by the teachers.

Name of the Program	**4-year Bachelor of Computer Applications- Single Major**	**Program Code**	**-------**
Name of the Course	**Computer Security**	**Course Code**	**25BCA406DS02**
Hours/Week	**5 (3+0+2)**	**Credits (L: T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25**	**Time of end-term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

This course provides a broad introduction to a variety of topics computer, network, and system security. These include system/software vulnerabilities, applied cryptography, host-based and network-based security, privacy, anonymity, usability, security economics, risk and vulnerabilities, policy formation, controls and protection methods and issues of law and privacy.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the core principles of data security, including the CIA triad (confidentiality, integrity, availability).

CO2: Identify and understand common network threats like viruses, worms, and phishing attacks.
CO3: Understand security controls like firewalls and access control mechanisms, and their role in network protection.
CO4: Demonstrate various network security application, Firewall, IDS and malicious software etc.

Unit – I

Overview of computer security concepts: Historical perspectives on computer security, Common threats and vulnerabilities in computer systems, Principles of security: Confidentiality, Integrity, Availability (CIA), Risk management and threat modeling. Security Functional requirement, Security Architecture and scope of Computer Security, Computer Security Trends and Strategies.

Cryptography: Basic concepts and techniques Symmetric encryption algorithms (e.g., DES, AES), Asymmetric encryption algorithms (e.g., RSA, Diffie-Hellman), Hash functions and digital signatures, Applications of cryptography in securing data and communications. Cryptanalysis.

Unit – II

Program Security: Secure Program, Non- malicious Program Error, Viruses and other Malicious Code, Targeted Malicious Code, Control against Program Threats.

Database Security: Database Management System, Relational Databases, Database Access Control, Inference, Security Requirement, Reliability and Integrity, Sensitive data, Database Encryption.

Unit – III

Network Security: Firewalls, IDS/IPS, VPNs, Secure communication protocols (e.g., SSL/TLS), Secure shell (SSH) and Secure File Transfer Protocol (SFTP) Intrusion detection and prevention systems, Network attacks and countermeasures (e.g., DDoS, man-in-the-middle), Wireless network security: WPA, WEP, WPA2, WPA3

Internet Security: Secure socket Layer (SSL) and Transport Layer Security (TLS), IPv4 and IPv6 Security, Public key Infrastructure.

Unit – IV

Linux Security: Linux Security Model, File System Security, Linux Vulnerability, Linux System Hardening, Application Security. Windows Security Architecture, Windows Security Défense, Browser Defences.

Physical Security: Threats, Physical Security Prevention and Mitigation Measures, Recovery form Physical Security Breaches, Security Risk assessment, Security Controls or safeguard, IT Security Plan, Cybercrime and Computer Crime, Intellectual Property, Privacy, Ethical Issues.

Suggested Readings:

1. William Stallings: Advanced Network Security Pearson Education Limited, 2017
2. Niels Ferguson, Bruce Schneier, Tadayoshi Kohno, John Wiley & Sons: Cryptography Engineering: Design Principles and Practical Applications, 2010
3. Eric Goldman: Web Security: A Concise Guide, O'Reilly Media, 2016
4. Jon Erickson, Patrick Engebretson: Hacking: The Art of Exploitation, Pearson Education, 2008
5. Matt Bishop: Computer Security: Art and Science, 2nd edition Addison- Wesley, 2019
6. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Lab Assignments/Programs

1. Write a program to encrypt a string by shifting each character by a fixed number and decrypt it back.
2. Write a program to simulate the key exchange to generate a shared secret between two users.
3. Write a program to simulate a brute-force attack to crack a password by trying all possible combinations.
4. Write a program to encrypt and decrypt files using a symmetric key encryption algorithm like AES.
5. Write a program to implement a basic Digital Signature system using hashing.
6. Write a program to implement public key cryptography using RSA.
7. Write a program to simulate a password strength checker based on entropy.

8. Write a program to simulate SSL/TLS handshake.
9. Write a program to implement the XOR-based encryption algorithm.
10. Write a program to implement an Intrusion Detection System using pattern matching.

Any other Practicals/Lab Assignments/Programs assigned by the teachers.

Name of Program	**4-year Bachelor of Computer Applications- Single Major**	**Program Code**	**- - - - - -**
Name of the Course	**Data Warehouse & Mining**	**Course Code**	**25BCA406DS03**
Hours Per Week	**5 (3+0+2)**	**Credits**	**3:0:1**
Maximum Marks	**Theory: 75 (50+25)** **Practical: 25**	**Time of Examinations**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

To impart knowledge on designing and managing data warehouses and utilizing data mining techniques for actionable insights. Students will learn about ETL processes, data analysis methods, and tools for clustering, classification, and association to solve decision-making problems in domains like business, healthcare, and research.

Course Outcomes:

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

CO1: Understand the architecture and components of data warehouses.
CO2: Design ETL processes for data integration and transformation.
CO3: Apply data mining techniques for pattern discovery and predictive analytics.
CO4: Evaluate and deploy algorithms for clustering, classification, and association.
CO5: Solve real-world problems using data warehousing and mining tools.

Unit – I

Introduction to Data Warehouse: Design Guidelines for Data Warehouse Implementation, dimensional modeling (star schema, snowflake schema)

OLAP: Introduction, Characteristics, Architecture, Multidimensional view, Efficient Processing of OLAP Queries, OLAP Server Architecture, ROLAP versus MOLAP Versus HOLAP and Data Cube, Data Cube Operations, Data Cube Computation. Motivation for data mining, Introduction to data mining system, Data mining functionalities, KDD, Data object and attribute types, Statistical description of data, Issues and Applications, OLAP vs OLTP.

Unit – II

Data Mining Fundamentals: Data Mining Concepts: Definitions, Tasks, and Applications, Data Preprocessing: Cleaning, Integration, Transformation, and Reduction, Association Rule Mining: Apriori and FP-Growth Algorithms, Classification: Decision Trees, Naive Bayes, Clustering Techniques: K-Means, Hierarchical Clustering.

Advanced Topics in Data Mining: Text Mining: Concepts and Techniques, Web Mining: Web Content, Structure, and Usage Mining, Big Data Analytics: Hadoop, Spark, and NoSQL Databases, Anomaly Detection: Methods and Use Cases, Case Study: Implementing Data Mining Techniques Using Tools like Weka.

Unit – III

Classification and Prediction: Issues regarding Classification & Prediction, Classification by Decision Tree Induction, Bayesian Classification, Classification by Back Propagation, k-Nearest Neighbour Classifiers, Genetic Algorithms, Rough Set & Fuzzy Set Approaches.

Unit – IV

Cluster Analysis: Types of Data in Clustering Analysis, Categorization of Major Clustering Methods, Hierarchical Methods, Density-based methods, Grid-based methods, Grid-based methods, Model-based Clustering Method.

Performance Optimization and Applications: Data Warehouse Optimization: Indexing, Partitioning, and Query Optimization, Data Visualization Tools: Tableau, Power BI, Ethical and Privacy Issues in Data Mining, Applications of Data Mining: Healthcare, E-commerce, Finance, Case Study: Analyzing Real-world Data for Predictive Insights.

Suggested Readings:

1. Han, J., Kamber, M., & Pei, J. Data Mining: Concepts and Techniques. Morgan Kaufmann.
2. Kimball, R., & Ross, M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. Wiley.
3. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.
4. J. Han, M Kamber: Data Mining Concepts and Techniques, Morgan Kaufmann.
5. M. Dunham: Data Mining - Introductory and Advance Topics, Pearson Education.
6. DT Editorial Services: Big Data, Black Book: Covers Hadoop 2, MapReduce, Hive, YARN, Pig, R and Data Visualization.
7. Mohammed Guller: Big Data Analytics with Spark: A Practitioner's Guide to Using Spark for Large Scale Data Analysis.
8. Glenn J. Myatt: Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining.
9. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals/Programs/Lab Assignments

1. Write a program to handle missing data (replace missing values with mean, median, etc.) in a dataset.
2. Write a program to Implement a program to aggregate data in a CSV file (sum, average, count, etc.) based on certain conditions.
3. Write a program to Implement the Apriori algorithm to find frequent item sets and generate association rules.
4. Write a program to Implement the K-means clustering algorithm to group a set of data points into clusters.
5. Write a program to visualize data using a histogram (bar chart).
6. Write a program to Build a decision tree using the ID3 or CART algorithm to classify data points.
7. Write a program to Implement linear regression to predict a dependent variable based on one independent variable.
8. Write a program to Implement the Naive Bayes classification algorithm to predict class labels from a dataset.
9. Write a program to Implement a program to compute the Pearson correlation coefficient between two datasets.
10. Write a program to Implement PCA to reduce the dimensionality of a dataset while preserving most of its variance.

Any other Practicals/Lab Assignments/Programs assigned by the teachers.

Name of the Program	**4-year Bachelor of Computer Applications- Single Major**	**Program Code**	**-----**
Name of the Course	**Advance Database Systems**	**Course Code**	**25BCA406DS04**
Hours/Week	**6 (2+0+4)**	**Credits (L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

Course Objectives:

This course aims to provide students with an in-depth understanding of advanced database models, architectures, query optimization techniques, and emerging database technologies. It covers object-oriented, object-relational, parallel, and distributed databases, as well as modern applications like spatial, multimedia, and XML databases.

Course Outcomes:

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

CO1: Understand the concepts of E-R Model and EER models.
CO2: Explore the concepts related to object-oriented and object-relational database systems.
CO3: Understand the working of parallel, and distributed database systems.
CO4: Explore Client/Server & Server architectures and how to retrieve information.
CO5: Analyze modern database applications, including Active, Temporal, spatial, multimedia, and XML-based databases etc.

Unit – I

EER Model: Prerequisite of Entity-Relationship Model: Entity Types, Entity Sets, Attributes & Keys, Relationship Types & Instances, ER Diagrams, Enhanced Entity Relationship (EER) Model: Subclasses, Superclasses, Inheritance, Specialization and Generalization, Constraints & Characteristics of Specialization and Generalization, Union/Category.

Object-Oriented & Object-Relational Databases: Object-Oriented Concepts, Object Identity Object Structure, Type Constructors, Encapsulation of operations, Methods and Persistence, Type hierarchies and Inheritance, Complex objects, Schema design for OODBMS, OQL, OODBMS architectures and Storage issues, Transaction Management & Concurrency Control.

Unit – II

Object Relational Database : Database design for an ORDBMS–Nested relations and collections; storage and access methods, Query processing and Optimization, Advance Querying: User define data types, manipulating objects table, object views;

Information Retrieval Information Retrieval & ways to retrieve information.

Client/Server Architecture: Client/Server Architectures, Server Architectures

Unit – III

Parallel Databases: Architectures for Parallel Databases, I/O Parallelism, Interquery Parallelism, Intraquery Parallelism.

Distributed Databases: Architectures for Distributed Database, Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design, Query processing in .Distributed Databases; Concurrency Control and Recovery in Distributed Databases.

Unit – IV

Enhanced Data Models for Advanced Applications: Active database-syntax and semantics (DB2,Oracle), applications, design principles for active rules, Temporal database concepts, Spatial databases, Deductive databases.

Emerging Database Technologies: Mobile databases, Multimedia Databases, Geographic Information systems (GIS); XML and Internet Databases: Structured, Semi-structured and Unstructured Data, Web databases.

Suggested Readings:

1. Elmasri and Navathe, Fundamentals of Database Systems, Pearson Education.
2. Korth, Silberchatz, Sudarshan, Database System Concepts, McGraw-Hill.
3. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, McGraw-Hill
4. Peter Rob and Coronel, Database Systems, Design, Implementation and Management, Thomson Learning.
5. C.J.Date, Longman, Introduction to Database Systems, Pearson Education
6. Thomas Connolly,CarolynBegg, Database Systems, Pearson Education
7. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Practicals / Lab Assignments

1. Write SQL commands to create tables (Students, Courses, Instructors) and insert sample data.
2. Retrieve specific data using SELECT, WHERE, ORDER BY, and GROUP BY.
3. Convert an ER diagram of a Library Management System into SQL tables.
4. Define relationships between tables (Students and Courses).
5. Write SQL queries with INNER JOIN, LEFT JOIN, and RIGHT JOIN.
6. Write SQL scripts to create tables (Employees, Employee_Log) and an AFTER INSERT trigger to log new employee entries into the log table.
7. Write a BEFORE UPDATE trigger on the Employees table to store old and new salary values in a Salary_Audit table whenever salary is updated.
8. Create a BEFORE DELETE trigger to archive student records from the Students table into a Students_Archive table before deletion.
9. Write a BEFORE INSERT trigger on the Products table to prevent insertion if the product price is less than zero.
10. Write an AFTER INSERT trigger to automatically update stock quantity in the Products table whenever a new order is added to the Orders table.

Any other Programs/Lab Assignments given by the teachers.