

**Revision – R-2026 (Suggestive Courses for 1st Year BCA)
Course Structure**

Induction Program													
Sl. No.	Course Code	Course Title	Credit Distribution				Assessment					Teaching Department (TD) and Question Paper Setting Board (PSB)	
			Lecture	Tutorial	Practical	Self-Learning	Examination in Hours	Formative Assessment		Summative Assessment	Total Marks		Credits
								Theory	Practical				
1	26IND001	Orientation Session (Induction Program)	Three Days Programme				100	-	100	1	CA		

Pre-Semester												
Sl. No.	Course Code	Course Title	Credit Distribution				Assessment					Teaching Department (TD) and Question Paper Setting Board (PSB)
			Lecture	Tutorial	Practical	Self-Learning	Examination in Hours	Formative Assessment	Summative Assessment	Total Marks	Credits	
			L	T	P	SL						
1	26FMC002	Foundational Mathematics	1	1	-	2	2½	70	30	100	2	MAT
2	26ENG003	Foundational English (B)	1	-	1	1	1	70	30	100	1	H & S
3	26CLP004	Computing & Life Skills (B)	1	-	-	1	1	70	30	100	1	CA
4	26UHV005	Universal Human Values (UHV) (B)	1	-	-	1	-	100	-	100	1	H & S
5	26MWP006	Mental Health & Wellbeing (B)	1	-	-	1	-	100	-	100	1	CA
Total Marks										500	06	

Department of Computer Applications**I - Semester**

Sl. No.	Course Code	Course Title	Teaching Hours per Week				Assessment						Teaching Department (TD) and Question Paper Setting Board (PSB)
			Lecture	Tutorial	Practical	Self-Learning	Examination in Hours	Formative Assessment		Summative Assessment	Total Marks	Credits	
			L	T	P	SL							
1	26BCA110	Web technologies	2	1	1	2	2½	70		30	100	4	CA
2	26BCA120	Problem Solving through C	2	1	1	2	2½	70		30	100	4	CA
3	26BCA130	Mathematical Foundation for Computer Applications	2	1	1	2	2½	70		30	100	4	BSC
4	26BCA140	Data Communication	2	1	1	2	2½	70		30	100	4	CA
5	26KVK150 / 26KAK150	Vyavaharika Kannda / Adalitha Kannada	0	0	1	1	-	100	-	100	100	1	H & S
Total Marks											500	17	

Department of Computer Applications**II - Semester**

Sl. No.	Course Code	Course Title	Teaching Hours per Week				Assessment						Teaching Department (TD) and Question Paper Setting Board (PSB)
			Lecture	Tutorial	Practical	Self-Learning	Examination in Hours	Formative Assessment		Summative Assessment	Total Marks	Credits	
1	26BCA210	Data Structures using C	2	1	1	2	2½	70	30	100	4	CA	
2	26BCA220	Object Oriented Programming using JAVA	2	1	1	2	2½	70	30	100	4	CA	
3	26BCA230	Discrete Mathematical Structures	2	1	1	2	2½	70	30	100	4	CA	
4	26BCA240	Foundation Course in Trending Technologies	2	1	1	2	2½	70	30	100	4	CA	
5	26CAI250	Foundation of Cyber Security & AI	1	-	-	1	-	100	-	-	100	1	CA
6	26PWS260	Professional Writing Skills in English	1	-	-	1	-	100	-	-	100	1	H & S
Total Marks											500	18	

Abbreviations:

BSC	Basic Science Course
ESC	Engineering Science Course
SEC	Skill Enhancement Course
H&S	Humanities & Science
B	Binary Graded Course
TD	Teaching Department
QPSB	Question Paper Setting Board
FA	Formative Assessment
SA	Summative Assessment
EH	Examination in Hours

Credit Definition:

Structure of Four Credit Course: Lecture (L), Tutorial (T) and Practice (P) are L: T:P is 2:1:1 and

One Tutorial session = Two hours, One Practice session = Two hours

Two hours of Tutorial / Practice per week = 1Credit

WEB TECHNOLOGIES

Course Code	26BCA110	CIE Marks	70
Teaching Hours/Week (L:T:P:SL)	2:2:2:2	SEE Marks	30
Total Hours of Pedagogy (L:T:P:SL)	120 (30:30:30:30)	Total Marks	100
Credits	04	Exam Hours	2 ½

Pre-requisites:

1. Understanding of How the Internet Works
2. Problem-Solving & Logic

Course Learning Objectives (CLOs):

1. To introduce students to the structure and syntax of HTML, enabling them to create and design basic web pages with properly formatted content, links, images, tables, and forms.
2. Identify tags, attributes and the styles among them needed for a finding the solution to specific problem.
3. Demonstrates how to create web applications with XHTML, CSS, and JavaScript.
4. Demonstrate understanding and use of JavaScript, different Event handling mechanisms for faster and efficient application development.
5. Identify and describe common user interface components to design GUI using XHTML, CSS, and JavaScript

Unit-1

Pedagogy	Page No.	From: 1 - 80				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	Week	Hour		Hour		
	1	Internet, WWW, Web Browsers and Web Servers		URLs, MIME,		
	2	HTTP, Security		Introduction to html, XHTML: Basic syntax, Standard structure,		
	3	Basic text mark-up, Images, Hypertext Links,		Lists, Tables, Forms, Frames		
Transactions	1	Identify components of a URL and trace an HTTP request-response cycle.				
	2	Design an HTML page with text formatting headings, images, hyperlinks, and lists				
	3	Create HTML pages with tables and forms				
Practice	1	Inspect HTTP headers and MIME types using browser Developer Tools				
	2	Build a structured web page with images, links, and lists				
	3	Design a web page with a registration form and a data table				

Unit-2

	Page No.	From: 95 - 126				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Introduction, Levels of style sheets, Style specification formats			Selector forms, Property value forms	
	2	Font properties, List properties, Color,			The box model, Background images	
	3	The <div> and tags			Conflict resolution.	
Transactions	1	Write CSS rules using element, class, and ID selectors				
	2	Apply font, color, alignment, and box model properties				
	3	use <div> and for layout ; Resolve CSS conflicts using specificity				

Department of Computer Applications

Practice	1	Design & Style a page using inline, internal, and external CSS				
	2	Design a page layout using box model (margin, padding, border)				
	3	Create a web page section layout using <div> and with background styling				
Unit-3						
	Page No.	From: 133 - 181				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Overview of JavaScript, Object orientation and JavaScript		Syntactic characteristics, Primitives, operations, and expressions		
	2	Screen output and keyboard input, Control statements		Object creation and modification, Arrays		
	3	Functions, Constructors		Pattern matching using regular expressions, Errors in scripts		
Transactions	1	Write JavaScript programs using primitives, operators, and control statements				
	2	Create and manipulate JavaScript objects and arrays				
	3	Write functions with constructors and regular expressions; debug errors				
Practice	1	Design & Implement JS programs with input/output and control flow (if, for, while)				
	2	Write programs using arrays and objects with methods				
	3	Build a form validation script using regular expressions				
Unit-4						
	Page No.	From: 189 - 225				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	The JavaScript execution environment		The Document Object Model, Element access in JavaScript		
	2	Events and event handling, handling events from the Body elements		Button elements, Text box and Password elements		
	3	The DOM 2 event model		the navigator object, DOM tree traversal and modification		
Transactions	1	Access and modify HTML elements using getElementById and querySelector				
	2	Attach event handlers to buttons, text boxes, and body elements				
	3	Traverse and modify the DOM tree; use the navigator object				
Practice	1	Write scripts to dynamically read and update DOM elements				
	2	Build an interactive form with event-driven JavaScript				
	3	Implement DOM traversal and display browser info using navigator object				
Unit-5						
	Page No.	From: 229 - 260				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Introduction, positioning elements		Moving elements, Element visibility		
	2	Changing colors and fonts		Dynamic content, Stacking elements		
	3	locating the mouse cursor, Reacting to a mouse click		slow movement of elements, Dragging and dropping elements.		
Transactions	1	Position and move HTML elements dynamically using JavaScript				
	2	Change element visibility, color, and font at runtime				
	3	Implement mouse tracking and click-based interactions				

Department of Computer Applications

Practice	1	Create a page with elements that show/hide and move on button click
	2	Build a dynamic page changing colors and content on user interaction
	3	Implement a simple drag-and-drop interface with animated movement

Course Outcomes:

- CO-1. Explain the architecture of the Web and describe the role of HTTP, URLs, MIME, and security in web communication.
- CO-2. Design and develop structured web pages using HTML/XHTML with text, images, hyperlinks, lists, tables, forms, and frames.
- CO-3. Apply Cascading Style Sheets to style and layout web pages using selectors, box model, font properties, and conflict resolution.
- CO-4. Develop interactive web applications using core JavaScript concepts including objects, arrays, functions, and regular expressions.
- CO-5. Create dynamic web pages by manipulating the DOM, handling events, and implementing animations and drag-and-drop interactions.

Text Books:

1. Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson Education, 2008.
2. Web Programming: Building Internet Applications — Chris Bates, Wiley, 3rd Edition

References:

1. Internet and World Wide Web: How to Program — Deitel & Deitel, Pearson, 5th Edition
2. JavaScript: The Definitive Guide — David Flanagan, O'Reilly, 7th Edition.
3. Xue Bai et al: The web Warrior Guide to Web Programming, Cengage Learning, 2003.
4. M Srinivasan: Web Technology Theory and Practice, Pearson Education, 2012.

Problem Solving through C

Course Code	26BCA120	CIE Marks	70
Teaching Hours/Week (L:T:P:SL)	2:2:2:2	SEE Marks	30
Total Hours of Pedagogy (L:T:P:SL)	120 (30:30:30:30)	Total Marks	100
Credits	04	Exam Hours	2 ½

Pre-requisites:

1. Basic Computer Literacy
2. Logical Thinking and Problem-Solving Skills
3. Fundamental Mathematical Knowledge
4. Introduction to Programming Concepts

Course Learning Objectives (CLOs):

1. Understand the fundamentals of algorithms, flowcharts, and the C programming environment.
2. Learn the syntax, semantics, data types, operators, and input/output functions in C programming.
3. Apply decision-making statements, loops, and arrays to solve programming problems effectively.
4. Design modular programs using user-defined functions and string-handling functions.
5. Develop efficient C programs using structures, unions, and pointers.

Unit-1

Pedagogy	Page No.	From: 1 - 50				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	Week	Hour		Hour		
	1	Algorithm: definition, characteristics; Flowchart: symbols, types		Draw flowcharts for: finding largest of 3 numbers, factorial, Fibonacci series		
	2	Introduction to C: history, features, structure of a C program		Creating and executing a C program; understanding the compilation process		
	3	C tokens: keywords, identifiers, constants, variables; data types		Declaration & initialization of variables; symbolic constants using #define		
Transactions	1	Design algorithm for a given problem (e.g., check Armstrong number); trace execution step-by-step as a group activity				
	2	Code tracing: read a C program, identify its structural components (header, main, declarations, statements, return); spot and correct syntax errors				
	3	Case-based task: given a real problem (calculate electricity bill), write algorithm, draw flowchart, identify required tokens and data types				
Practice	1	Write algorithms and draw flowcharts for: (a) largest of three numbers (b) factorial of a number (c) Fibonacci series. Compile each and observe output.				
	2	Write and execute a C program to read two numbers and print their sum. Label each structural component in the code (headers, main(), variables, I/O, return).				
	3	Write programs to demonstrate: (a) keywords and identifiers, (b) integer, float, char, and double data types, (c) symbolic constants using #define and const.				

Unit-2

	Page No.	From: 51 - 85				
		L	T	P	SL	Total

Department of Computer Applications

	Number of hours of engagement	6	6	6	6	24
Learning	1	Formatted I/O: printf, scanf — format specifiers, control strings, escape sequences		Unformatted I/O: getchar, putchar, gets, puts — usage and comparison		
	2	Arithmetic operators; Relational operators; Logical operators — syntax and examples		Assignment, Increment/Decrement, Bitwise, Conditional (ternary), Special operators		
	3	Operator precedence and associativity; evaluation of arithmetic expressions		Type conversion (implicit and explicit) mathematical functions using math.h		
Transactions	1	Code tracing: predict output of printf/scanf programs with different format specifiers and escape sequences; identify and fix mismatched specifier bugs				
	2	Problem-solving: evaluate complex expressions manually applying precedence rules; compare output of prefix vs postfix increment in programs				
	3	Case-based activity: build a simple calculator using operators; group code review for operator misuse, precedence errors, and type mismatch				
Practice	1	Write programs to read and display int, float, and char values using printf/scanf. Demonstrate escape sequences: newline (\n), tab (\t), backslash (\).				
	2	Write programs demonstrating: (a) unformatted I/O functions — getchar/putchar for characters, gets/puts for strings, (b) all arithmetic operators on two numbers.				
	3	Write programs to demonstrate: (a) relational and logical operators for condition checking, (b) bitwise operators (AND, OR, XOR, shift), (c) conditional operator and mathematical functions from math.h (sqrt, pow, fabs).				
Unit-3						
	Page No.	From: 86 - 120				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Decision-making: simple if, if-else, nested if-else; real-world examples		Else-if ladder; switch-case statement; goto statement and its limitations		
	2	Looping: while loop and do-while loop — syntax, differences, entry vs exit control		for loop; nested loops; break and continue statements with use cases		
	3	Arrays: 1-D arrays — declaration, initialization, memory representation, processing		2-D arrays — declaration, initialization, traversal, memory representation		
Transactions	1	Problem-solving: decide between if-else and switch-case for given scenarios; trace nested if-else execution with specific input values; group debugging of flawed decision programs				
	2	Code tracing: trace while/for loop execution step-by-step; predict output when break/continue are used; rewrite a while loop as a for loop and vice versa				
	3	Case-based activity: solve a student mark-processing problem — input marks for 5 students, find max, min, average using arrays; debug a 2-D matrix traversal program				
Practice	1	Write programs: (a) find largest of three numbers using if-else, (b) classify student grade using nested if-else and else-if ladder, (c) display day of the week using switch-case.				
	2	Write programs: (a) print first 10 natural numbers using while loop, (b) generate Fibonacci series using for loop, (c) print multiplication table using nested loops, (d) demonstrate break and continue with examples.				
	3	Write programs: (a) read and display elements of a 1-D array, (b) find sum and average of array elements, (c) read and display a 2-D matrix, (d) add two matrices				

Department of Computer Applications

		using 2-D arrays.				
Unit-4						
	Page No.	From: 121 - 150				
	Number of hours of engagement	L	T	P		Total
		6	6	6	6	24
Learning	1	Strings: declaring and initializing string variables; character arrays vs strings		String handling functions: strlen, strcmp, strcpy, strcat — syntax and examples		
	2	User-defined functions: need, format, components — return type, name, parameters, body, return statement		Categories: without parameters/return; with parameters/return; mixed categories		
	3	Function call mechanism: actual vs formal parameters; scope and lifetime of variables		Integrating functions with strings: design a modular text-processing program		
Transactions	1	Code tracing: trace strlen, strcmp, strcpy, strcat operations on given strings; predict output; debug programs with incorrect string function usage (e.g., missing null terminator)				
	2	Problem-solving: decompose a given problem (student result system) into functions; identify what each function should receive and return; peer review of function designs				
	3	Case-based activity: build a menu-driven C program using functions for different operations; evaluate modularity and reusability of each function through group discussion				
Practice	1	Write programs demonstrating string functions: (a) strlen — find length of a string, (b) strcmp — compare two strings and display result, (c) strcpy — copy one string to another, (d) strcat — concatenate two strings.				
	2	Write programs: (a) function to find square of a number without parameters and return, (b) function to check even or odd with parameters and return value, (c) function to display a formatted menu without parameters and return.				
	3	Write programs: (a) calculate sum of two numbers using function with parameters and return type, (b) build a mini calculator (add, subtract, multiply, divide) using separate functions and a menu-driven main function.				
Unit-5						
	Page No.	From: 151 - 180				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Structures: definition, advantages, declaring and initializing structure variables, accessing members		Array of structures; embedded (nested) structures; comparing structure variables		
	2	Unions: definition, declaring union variables, accessing members, initialization; difference between structures and unions		Pointers: declaring and initializing pointers; accessing address (&) and value (*) of variables		
	3	Pointer arithmetic: increment, decrement, addition, subtraction on pointers		Pointers and arrays — traversal using pointers; advantages and disadvantages of pointers		
Transactions	1	Code tracing: trace structure member access and nested structure programs; debug programs with incorrect member access syntax (wrong dot/arrow operator usage)				
	2	Problem-solving: compare memory usage of structure and union using sizeof; analyze when to use union over structure; trace pointer declaration and dereference				

Department of Computer Applications

		operations
	3	Case-based activity: design a student database using array of structures; use pointers to traverse and display student records; group evaluation of memory efficiency
Practice	1	Write programs: (a) define Student structure (roll no., name, marks) — initialize and access members, (b) create array of structures for 5 students and display all details, (c) demonstrate embedded structures (Address structure inside Student structure).
	2	Write programs: (a) define union for int, float, char — demonstrate initialization and access of members, (b) compare memory occupied by a structure and a union with same members using sizeof operator.
	3	Write programs: (a) declare and initialize pointer variables, access variable value and address using pointer, (b) perform pointer arithmetic (++ , -- , +n), (c) demonstrate relationship between pointers and arrays — traverse array using pointer arithmetic.
Course Outcomes: CO-1. Design algorithms and draw flowcharts for given problems and implement them as C programs CO-2. Read, understand, trace, and write C programs using variables, data types, operators, and I/O functions. CO-3. Implement decision-making and looping constructs; use arrays to store and process data. CO-4. Design modular programs using user-defined functions; apply string-handling functions. CO-5. Implement programs using structures, unions, and pointers for organized and efficient data management.		
Text Books: 1. Balaguruswarny: Programming in ANSI C, Tata McGraw-Hill.		
References: 1. C — The Complete Reference, 4th Edition. 2. https://nptel.ac.in/courses/106105085 3. https://www.w3schools.com/c/		

**DATA COMMUNICATION**

Course Code	26BCA140	CIE Marks	70
Teaching Hours/Week (L:T:P:SL)	2:2:2:2	SEE Marks	30
Total Hours of Pedagogy (L:T:P:SL)	120 (30:30:30:30)	Total Marks	100
Credits	04	Exam Hours	2 ½

Pre-requisites:

1. Basic knowledge of computer fundamentals , operating systems, digital data representation and binary number systems
2. Basic networking concepts and Internet usage, communication devices and computer hardware basics.

Course Learning Objectives (CLOs):

1. Understand the fundamental concepts of data communication, computer networks, network topologies, protocols, and communication standards.
2. Analyze analog and digital transmission techniques, encoding methods, multiplexing, and transmission media used in data communication systems.
3. Identify and evaluate various network connectivity devices, switching technologies, and network security devices used in modern computer networks.
4. Understand LAN technologies, NIC architecture, Ethernet standards, wireless LAN concepts, and IEEE networking standards.
5. Develop knowledge of Internet architecture, IP addressing, DNS, web technologies, Bluetooth, cellular communication, and Internet-based services.

MODULE – I

Pedagogy	Page No.	From: 3 - 54, 1 - 90				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	Week	Hour		Hour		
	1	Introduction to Data Communication, components.		Data representation, data flow, and communication process.		
	2	Networks: Network criteria, topology, physical structure.		Network classification, Internet concepts, protocols, and standards.		
	3	Switching techniques: Message, Packet, and Circuit Switching.		Types of Servers, Peer-to-Peer and Client/Server Architecture.		
Transactions	1	Explanation of data communication components, data representation methods, and communication models with examples.				
	2	Discussion on network topologies, classifications, Internet concepts, protocols, and standards.				
	3	Demonstration of switching techniques, server types, and comparison of Peer-to-Peer and Client/Server architecture.				
Practice	1	Identify components of data communication and explain data flow methods.				
	2	Draw different network topologies and classify computer networks.				
	3	Differentiate switching techniques and compare Peer-to-Peer and Client/Server architectures.				

MODULE – II

	Page No.	From: 57 - 210, 91 - 250				
		L	T	P	SL	Total

Department of Computer Applications

	Number of hours of engagement	6	6	6	6	24
Learning	1	Digital representation of information, Analog and Digital Signals.			Transmission Modes, and basic Digital Transmission concepts.	
	2	Line Coding, Digital Encoding Techniques, Modulation concepts.			Multiplexing techniques: FDM, TDM, and WDM.	
	3	Transmission Media: Guided Media – Twisted Pair, Co-axial Cable, Optical Fiber.			Unguided Media concepts	
Transactions	1	Explanation of analog and digital signals, transmission modes, and digital transmission techniques with examples.				
	2	Discussion on line coding, encoding techniques, modulation, multiplexing methods, and SONET architecture.				
	3	Demonstration of guided and unguided transmission media and comparison of communication channels.				
Practice	1	Differentiate analog and digital signals and identify transmission modes.				
	2	Compare FDM, TDM, and WDM multiplexing techniques with suitable examples.				
	3	Identify different transmission media and compare twisted pair, coaxial cable, and optical fiber characteristics.				
MODULE – III						
	Page No.	From:395 - 466, 251 - 420				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Categories of Connectivity Devices, Passive and Active Hubs.			Repeaters, and basic networking devices.	
	2	Bridges: Local Bridge, Remote Bridge, Wireless Bridge, Transparent Bridge.			Source Routing Bridge concepts.	
	3	Switches: Layer-2 and Layer-3 Switches, Gateways, Firewalls, Proxy Servers,			Real-world network security applications.	
Transactions	1	Explanation of networking devices, hubs, and repeaters with working principles and applications.				
	2	Discussion on different types of bridges and switching techniques used in computer networks.				
	3	Demonstration of gateways, firewalls, proxy servers, and network security concepts in real-world environments.				
Practice	1	Identify different connectivity devices and compare hubs and repeaters.				
	2	Differentiate types of bridges and switches with suitable examples.				
	3	Explain the role of gateways, firewalls, and proxy servers in network security.				
MODULE – IV						
	Page No.	From: 487 - 620 , 421 - 560				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Network Interface Card (NIC), Network Adapters.			Components and Functions of NIC, and Types of NIC.	
	2	LAN Standards: Ethernet and IEEE 802.3 Standard.			Token Ring and IEEE 802.5 Standard.	
	3	FDDI, Wireless LANs, and IEEE 802.11 LAN Standards .			with basic wireless networking concepts.	
Transactions	1	Explanation of NIC architecture, network adapters, functions, and types used in LAN communication.				

**Department of Computer Applications**

	2	Discussion on Ethernet, IEEE 802.3, Token Ring, and IEEE 802.5 LAN standards with applications.
	3	Demonstration of FDDI and Wireless LAN standards including IEEE 802.11 communication methods.
Practice	1	Identify components and functions of NIC and compare different NIC types.
	2	Identify components and functions of NIC and compare different NIC types.
	3	Explain FDDI and Wireless LAN standards and identify IEEE 802.11 applications.
MODULE – V		
	Page No.	From: 741 - 820, 561 - 820
	Number of hours of engagement	L T P SL Total
		6 6 6 6 24
Learning	1	Introduction to Internet, Internet Growth, Architecture. Internet Accessing methods, and Internet Service Providers (ISP).
	2	Internet Addressing System: IP Address, DNS, URL. Concepts of World Wide Web (WWW), Web Servers, Web Browsers, and Search Engines.
	3	Concept of Intranet, Bluetooth technology Cellular Telephony basics.
Transactions	1	Explanation of Internet architecture, ISP functions, and methods of accessing the Internet.
	2	Discussion on IP addressing, DNS, URL structure, WWW services, web browsers, and search engines.
	3	Demonstration of intranet concepts, Bluetooth communication, and cellular telephony technologies.
Practice	1	Identify Internet components and explain the role of ISPs.
	2	Differentiate IP Address, DNS, and URL with suitable examples.
	3	Explain the applications of WWW, Bluetooth, Intranet, and Cellular Telephony.

Course Outcomes:

- CO-1. Understand and explain the fundamental concepts of data communication, computer networks, communication models, protocols, standards, and network architectures used in modern communication systems.
- CO-2. Analyze analog and digital transmission techniques, encoding methods, modulation, multiplexing, and different types of guided and unguided transmission media.
- CO-3. Identify, compare, and evaluate various networking and connectivity devices such as hubs, repeaters, bridges, switches, gateways, firewalls, and proxy servers used in computer networks.
- CO-4. Understand LAN technologies, Network Interface Cards (NIC), Ethernet standards, Token Ring, FDDI, Wireless LAN concepts, and IEEE networking standards for local area communication.
- CO-5. Internet concepts, IP addressing techniques, DNS, URL structure, web technologies, Bluetooth, and cellular communication technologies to real-world networking environments.

Text Books:

1. Data Communications and Networking – Behrouz A. Forouzan, McGraw-Hill Education.
2. Computer Networks – Andrew S. Tanenbaum and David J. Wetherall, Pearson Education.

References:

1. Data and Computer Communications – William Stallings, Pearson Education.
2. Computer Networking: A Top-Down Approach – Pearson Education.
3. Introduction to Data Communications and Networking – Pearson Education.

T.E.H.R.D Trust®

ಕಿಷ್ಕಿಂಡ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬಳ್ಳಾರಿ

Kishkinda University, Ballari

Established under Karnataka Act No. 20 of 2023 | Recognised by Govt. of Karnataka & UGC, New Delhi

www.kishkindauniversity.edu.in

Department of Computer Applications



advancing knowledge
transforming lives

4. https://nptel.ac.in/courses/106105183?utm_source
5. <https://www.geeksforgeeks.org/computer-networks/computer-network-tutorials/>
6. https://www.tutorialspoint.com/data_communication_computer_network/index.htm?utm_source

☎ 6364482226, 6364482227

✉ info@kishkindauniversity.edu.in

📍 Mount View Campus, off 28km, Ballari-Siruguppa Road, Near Sindhigeri,
GP No. 735, Hagalur, Siruguppa Taluk, Ballari - 583120 Karnataka.

DATA STRUCTURES USING C

Course Code	26BCA210	CIE Marks	70
Teaching Hours/Week (L:T:P:SL)	(2:2:2:2)	SEE Marks	30
Total Hours of Pedagogy (L:T:P:SL)	120 (30:30:30:30)	Total Marks	100
Credits	04	Exam Hours	2 ½

Pre-requisites:

1. Programming fundamentals.
2. Mathematical background
3. Problem solving and logical thinking
4. Knowledge on pointers and dynamic memory allocation.

Course Learning Objectives (CLOs):

1. To understand concepts about searching and sorting techniques
2. To understand basic concepts about stacks, queues, lists, trees and graphs.
3. To understanding about writing algorithms and systematic approach in solving problems with the help of fundamental data structures.
4. To understand concepts about trees and graphs.

MODULE – I

Pedagogy	Page No.	From: 1 - 30				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	Week	Hour		Hour		
	1	Introduction to Data Structures		Operations on Data Structures		
	2	Algorithm Specification		Performance Analysis		
	3	Recursion Basics		Recursion Applications & Comparison		
Transactions	1	App Performance Thinking				
	2	Folder Inside Folder (Recursion)				
	3	Data Organization in Apps				
Practice	1	Why do different types of data (images, text, numbers) require different data structures? Explain with example.				
	2	Why is it important to analyze the performance of an algorithm before implementing it in real-world applications?				
	3	In which situations is recursion more useful than iteration? Give a real-world example.				

MODULE – II

	Page No.	From: 51 - 85				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Basics of Arrays		Array Representation & Traversal		
	2	Insertion & Deletion		Searching Techniques		
	3	Sorting Techniques(Basics)		Sorting Techniques(Advanced)		
Transactions	1	Student Marks Management				
	2	E-commerce Product Sorting				
	3	Image Storage (2D Arrays)				

**Department of Computer Applications**

Practice	1	Why is array suitable for storing student marks when the number of students is fixed?				
	2	Why is binary search faster than linear search in e-commerce applications?				
	3	Why is it inefficient to store full matrix data for images with many empty pixels?				
MODULE – III						
	Page No.	From: 86 - 120				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Stack Basics		Stack Applications		
	2	Expression Handling		Recursion		
	3	Queue Basics		Types of Queues		
Transactions	1	Undo/Redo System (Stack)				
	2	Call Stack in Mobile Apps (Stack)				
	3	Queue in Real Systems				
Practice	1	Why are stacks used in undo/redo operations instead of queues?				
	2	Why does a mobile app return to the previous screen when you press the back button? How is stack useful here?				
	3	Why is queue preferred over stack in ticket booking systems?				
MODULE – IV						
	Page No.	From: 121 - 150				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Memory Management Basics		Dynamic Memory Functions		
	2	Introduction to Linked Lists		Types of Linked Lists		
	3	Operations on Singly Linked List		Memory Management in Linked Lists		
Transactions	1	Chat Application Messages (Dynamic Memory)				
	2	Train Reservation System (Linked List)				
	3	Image/Video Buffer Management				
Practice	1	Why is dynamic memory allocation preferred over static allocation in applications where data size changes frequently?				
	2	Why are linked lists preferred over arrays when frequent insertions and deletions are required?				
	3	What happens if memory is not deallocated after use? Explain memory leak with example.				
MODULE – V:						
	Page No.	From: 151 - 180				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Introduction to Trees & Terminology		Binary Trees & Types		
	2	Array Representation of Binary Tree		Tree Traversals		
	3	Comparison of traversals		Reconstruction of Binary Tree		
Transactions	1	Website Menu Structure				
	2	Tournament Bracket System				
	3	Data Processing Order				
Practice	1	Why is a tree structure suitable for representing website menus instead of arrays or linked lists?				
	2	How does a binary tree help in organizing knockout tournaments efficiently?				

**Department of Computer Applications**

	3	In which situations is postorder traversal more useful than preorder or inorder? Explain with example.
Course Outcomes: CO-1. Describe how arrays, linked structures, stacks, queues, trees are represented in memory and used by algorithms CO-2. Describe common applications for arrays, linked structures, stacks, queues, trees CO-3. Write programs that use arrays, linked structures, stacks, queues, trees, CO-4. Compare alternative implementations of data structures with respect to performance CO-5. Describe the concept of recursion; give examples of its use.		
Text Books: 1. Kamthane: Introduction to Data Structure in C, Pearson Education 2005. 2. Langsam, Ausenstein Maoshe & M. Tanenbaum Aaron, Data Structure using C and C++ Pearson Education.		
References: 1. Weiss: Data Structure and Algorithm Analysis in C, IInd Edition, Pearson Education. 2. Lipschutz: Schaum's outline series Data Structures, Tata McGraw Hill. 3. Tenenbaum: Data Structures using C, Pearson Education.		

OBJECT ORIENTED PROGRAMMING USING JAVA

Course Code	26BCA220	CIE Marks	70
Teaching Hours/Week (L:T:P:SL)	(2:2:2:2)	SEE Marks	30
Total Hours of Pedagogy (L:T:P:SL)	120 (30:30:30:30)	Total Marks	100
Credits	04	Exam Hours	2 ½

Pre-requisites:

1. Basic computer knowledge
2. Familiarity with programming fundamentals
3. Understanding of variables, loops, and functions

Course Learning Objectives (CLOs):

1. Students will learn the fundamentals of Java programming
2. Students will understand classes, objects, and Object-Oriented Programming concepts
3. Students will develop programs using inheritance and polymorphism
4. Students will create simple GUI applications with event handling
5. Students will understand multithreading and file handling concepts

MODULE – I

Pedagogy	Page No.	From: 1 - 140				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	Week	Hour		Hour		
	1	Introduction to Java and its features		Operators in Java		
	2	JVM, JDK and JRE		Decision making, Control Statements		
	3	Variables and Data Types		Arrays and Strings		
Transactions	1	A small shop owner wants to quickly calculate bills for customers. Can you create a calculator using arithmetic operators? How can Java operators and conditional statements help in solving this problem?				
	2	A school teacher wants to calculate total marks and grades for students. Can you design a program that stores marks and displays average and grade? How can arrays and loops make this task easier				
	3	A family wants to know their monthly electricity bill based as per the usage ie 0-100 2 rs/unit ,100-200 – 3/unit 200-400 5/unit etc . Can you build a simple Java program to calculate the bill amount? How can conditional statements help in applying different electricity charges?				
Practice	1	Explain how Java handles different data types and variables in real-world applications.				
	2	Discuss the importance of loops and arrays in solving repetitive tasks efficiently.				
	3	Compare decision-making statements such as if-else and switch with suitable examples from daily life.				

MODULE – II

	Page No.	From: 141 – 260				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
	1	Classes and Objects		Method Overriding and Dynamic Binding		

Department of Computer Applications

Learning	2	Constructors and Method Overloading	Packages and Access Specifiers
	3	Inheritance and Types of Inheritance	Interfaces and Abstract Classes
Transactions	1	A college office wants to store details of students like name, roll number, and marks. create a Java class and objects to manage multiple student information.	
	2	A bank wants a small application to store account details and perform deposits or withdrawals. Can you create objects for different bank customers and manage their balances? Use methods and constructors simplify this task.	
	3	An online shopping company wants to support different payment methods such as UPI, Credit Card, and Net Banking. Can you design a Java program using interfaces to implement multiple payment options? How can packages be used to organize payment-related classes?	
Practice	1	Explain how classes and objects help in organizing real-world data in software applications.	
	2	Discuss the role of constructors and methods in improving program structure and readability.	
	3	Explain how packages and access specifiers improve security and modularity in Java applications.	

MODULE – III

	Page No.	From: 261 – 340				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Inheritance and Types		Polymorphism		
	2	Method Overriding		Abstract Classes		
	3	super keyword		Interfaces		
Transactions	1	A university examination system handles Undergraduate, Postgraduate, and Research students differently.Can you implement inheritance and method overriding to calculate grades for each category? How can runtime polymorphism improve flexibility in result processing?				
	2	A ride-booking application supports Mini, Sedan, and SUV vehicles. Can you design a Java program using abstract classes and interfaces for fare calculation and booking services? How can dynamic method dispatch help manage different vehicle types efficiently?				
	3	A banking application supports Savings, Current, and Loan accounts. Can you design an abstract class for account operations and override interest calculation methods in subclasses? How can the super keyword be used to reuse common banking functionalities?				
Practice	1	Explain how inheritance promotes code reusability in large software systems.				
	2	Discuss the advantages of polymorphism in designing flexible and maintainable applications.				
	3	Compare abstract classes and interfaces with suitable real-world examples.				

MODULE – IV

	Page No.	From: 720 – 980				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Introduction to GUI Programming		Event Handling		
	2	AWT Components		ActionListener and Mouse Listener		
	3	Swing Components		Layout Managers		

**Department of Computer Applications**

Transactions	1	A college library wants a secure login system for students and staff. Can you design a GUI-based login form using buttons, labels, and text fields? How can event handling validate user credentials dynamically?
	2	A hospital wants patients to register and book appointments digitally. Can you design a registration form using frames, panels, and combo boxes? How can GUI programming simplify appointment management?
	3	A restaurant wants customers to place orders through a graphical interface. Can you create a Java Swing application that allows users to select food items and calculate the final bill? How can action events improve user interaction during order processing?
Practice	1	Explain the importance of GUI programming in developing user-friendly applications.
	2	Discuss how event handling improves interaction between the user and the application.
	3	Compare different Swing components and explain their roles in GUI design.

MODULE – V:

	Page No.	From: 41 – 520 , 620 – 690				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Introduction to Threads		Synchronization Basics		
	2	Creating Threads		Exception Handling		
	3	Thread Life Cycle		File Handling		
Transactions	1	A railway department wants to allow multiple passengers to book tickets simultaneously. Can you create a multithreaded Java application to handle concurrent bookings? How can synchronization prevent double booking of seats?				
	2	A university wants to store and retrieve student marks from files. Can you design a Java program using file handling and streams to manage examination records?				
	3	An organization wants to automatically create backup copies of important files. Can you develop a multithreaded file-copy program using byte streams? How can threads improve backup speed and efficiency?				
Practice	1	Explain how multithreading improves the performance of modern applications.				
	2	Discuss the importance of file handling and streams in managing data efficiently.				
	3	Explain the challenges of concurrent execution and the need for synchronization in Java.				

Course Outcomes:

- CO-1. Understand and apply the fundamentals of Java programming including variables, operators, control statements, arrays, and basic program development.
- CO-2. Develop Java applications using classes, objects, constructors, methods, packages, and object-oriented programming concepts.
- CO-3. Implement inheritance, polymorphism, abstract classes, and interfaces to create reusable and flexible Java programs.
- CO-4. Design and develop GUI-based and event-driven applications using Java Swing and event handling mechanisms.
- CO-5. Create multithreaded and file-handling applications using threads, exception handling, and input/output streams in Java.



Text Books:

1. Herbert Schildt, Java: The Complete Reference, McGraw Hill.
2. E. Balagurusamy, Programming with Java, McGraw Hill.
3. Herbert Schildt, Java 2: The Complete Reference, McGraw Hill.

References:

1. Cay S. Horstmann, Core Java Volume I – Fundamentals, Pearson.
2. Deitel & Deitel, Java How to Program, Pearson.
3. Bruce Eckel, Thinking in Java, Prentice Hall.
4. The syllabus topics are aligned with concepts covered in Herbert Schildt's Java: The Complete Reference.

FOUNDATION COURSE IN TRENDING TECHNOLOGIES

Course Code	26BCA240	CIE Marks	70
Teaching Hours/Week (L:T:P:SL)	2:2:2:2	SEE Marks	30
Total Hours of Pedagogy (L:T:P:SL)	120 (30:30:30:30)	Total Marks	100
Credits	04	Exam Hours	2 ½

Pre-requisites:

1. Students should have computer knowledge.
2. Students should have Basic idea about Internet, AI, Cloud Computing

Course Learning Objectives (CLOs):

1. To understand the fundamentals of Artificial Intelligence and types of intelligent agents.
2. To explore the Internet of Things architecture, definitions, frameworks, and real-world applications.
3. To comprehend the characteristics, types, sources, and analytics techniques of Big Data.
4. To analyze various computing paradigms and understand cloud computing fundamentals and services.
5. To understand the concepts of cyberspace, cybersecurity issues, internet infrastructure, and governance.

Unit-1

Pedagogy	Page No.	From: 1 - 51				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	Week	Hour		Hour		
	1	What is Artificial Intelligence, Foundations of AI, History of AI		Applications of AI in real world		
	2	Intelligent Agents: Environments, Specifying the task environment, Properties of task environments		Types of agents overview		
	3	Simple reflex agents, Model-based reflex agents		Goal-based agents, Utility-based agents		
Transactions	1	Identify and classify types of AI environments with examples (fully/partially observable, deterministic, stochastic)				
	2	Compare simple reflex, model-based, goal-based, and utility-based agents with real-world scenarios				
	3	Analyze agent behavior for a given task environment and determine the most suitable agent type				
Practice	1	Explore AI tools and platforms (e.g., ChatGPT, Google Bard) and document their capabilities and limitations				
	2	Simulate a simple reflex agent behavior using pseudocode or flowchart for a given environment				
	3	Design a goal-based agent for a real-world problem (e.g., route planning, medical diagnosis)				

Unit-2

	Page No.	From: 1 - 67				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24

**Department of Computer Applications**

Learning	1	What is The Internet of Things? Overview and Motivations, Examples of Applications	Areas of Development and Standardization, Scope of the Present Investigation
	2	Things Definitions and frameworks- IoT Definitions, IoT Frameworks, Basic Nodal Capabilities	Internet of Things Application Examples Overview
	3	Smart Metering/Advanced Metering Infrastructure, E Health/Body Area Networks,	City Automation, Automotive Applications, Home Automation.
Transactions	1	Explain IoT architecture and identify motivations behind IoT adoption with examples	
	2	Describe IoT frameworks and nodal capabilities; compare smart metering with traditional metering	
	3	Analyze IoT applications in e-health, city automation, automotive, and home automation domains	
Practice	1	Identify and document IoT devices and sensors available in daily life; map them to IoT layers	
	2	Simulate an IoT use case (e.g., smart home) using an online IoT simulator (e.g., Cisco Packet Tracer / Tinkercad)	
	3	Design an IoT solution for a smart city or health monitoring system with a block diagram	

Unit-3

	Page No.	From: : 1 - 50				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Need Of Big Data, Big Data Definition, Big Data Characteristics		Big Data Types, Data Sources, Data Quality		
	2	Data Preprocessing, Data Storing and Analysis		Big Data Handling Techniques		
	3	Big Data Analytics and its applications.				
Transactions	1	Explain the 5 Vs of Big Data with real-world examples; identify Big Data sources and types				
	2	Describe data quality dimensions and preprocessing steps; compare structured vs unstructured data				
	3	Analyze Big Data handling techniques and identify suitable analytics tools for given scenarios				
Practice	1	Explore a Big Data dataset (e.g., Kaggle) and identify its characteristics (volume, variety, velocity)				
	2	Perform basic data preprocessing tasks (handling missing values, normalization) using Excel or Python				
	3	Use a Big Data tool (e.g., Hadoop demo / Hive) to store and query a sample dataset				

Unit-4

	Page No.	From: 1 - 22				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Introduction: Different Computing Paradigms Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing, Cloud Computing		Comparison of various Computing Technologies		
	2	Defining Cloud Computing: NIST		5-4-3 principles of Cloud Computing,		

Department of Computer Applications

		Definition of Cloud Computing, Cloud Computing is a service, Cloud Computing is platform	Cloud Eco System
	3	Requirements of Cloud Services, Cloud Application	Benefits and Drawbacks
Transactions	1	Compare parallel, distributed, cluster, grid, and cloud computing with suitable diagrams	
	2	Explain NIST definition of cloud computing; differentiate IaaS, PaaS, and SaaS with examples	
	3	Analyze the 5-4-3 principles of cloud computing; list benefits and drawbacks for a given scenario	
Practice	1	Explore free-tier cloud platforms (AWS / Google Cloud / Azure) and document available services	
	2	Deploy a simple static website or application using a cloud platform (e.g., Google Sites / AWS S3)	
	3	Compare at least three cloud service providers on cost, features, and performance using a table	

Unit-5

	Page No.	From: 1 - 30				
	Number of hours of engagement	L	T	P	SL	Total
		6	6	6	6	24
Learning	1	Defining Cyberspace and Overview of Computer and Web-technology, Architecture of cyberspace,		Communication and web technology, Internet, World wide web		
	2	Advent of internet, Internet infrastructure for data transfer and governance, Internet society		Regulation of cyberspace,		
	3	Concept of cyber security		Issues and challenges of cyber security		
Transactions	1	Define cyberspace and explain the architecture of the internet and web technology				
	2	Describe internet infrastructure for data transfer; explain the role of ICANN and internet governance				
	3	Explain the concept of cyber security and identify current issues and challenges in securing cyberspace				
Practice	1	Analyze a real-world cyber attack case study (e.g., phishing, ransomware) and document findings				
	2	Explore cybersecurity tools (e.g., Wireshark) to observe internet traffic patterns				
	3	Identify cybersecurity vulnerabilities in a sample web scenario and suggest preventive measures				

Course Outcomes:

- CO-1. Students will explain about AI History and AI Agents.
- CO-2. Students will discuss about IOT and applications of IOT.
- CO-3. Students will be familiar about Big Data analytics.
- CO-4. Students will have basic idea about Cloud Computing Fundamentals.
- CO-5. Have an intelligent conversation on the key concepts and applications Cybersecurity.

T.E.H.R.D Trust®

ಕಿಷ್ಕಿಂಡ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬಳ್ಳಾರಿ

Kishkinda University, Ballari

Established under Karnataka Act No. 20 of 2023 | Recognised by Govt. of Karnataka & UGC, New Delhi

www.kishkindauniversity.edu.in

Department of Computer Applications



advancing knowledge
transforming lives

Text Books:

1. Start Russel, Peter Norvig: Artificial Intelligence A Modern Approach, 2nd Edition, Pearson Education, 2003

References:

1. Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications
2. Big Data Analytics by Raj kamal and Preeti Saxena
3. K Chandrashekar: "Essentials of Cloud Computing", CRC Press, 2015. 5.
4. Cyber Crime Impact in the New Millennium, by R. C Mishra, Auther Press. Edition 2010

☎ 6364482226, 6364482227

✉ info@kishkindauniversity.edu.in

📍 Mount View Campus, off 28km, Ballari-Siruguppa Road, Near Sindhigeri,
GP No. 735, Hagalur, Siruguppa Taluk, Ballari - 583120 Karnataka.