KUMARAGURU COLLEGE OF TECHNOLOGY

An autonomous Institution affiliated to Anna University, Chennai

COIMBATORE – 641 049.

B.TECH. INFORMATION TECHNOLOGY

REGULATION 2024 (2026 Batch)

I to VIII Semesters

Department of Information Technology

Signature of BOS Chairman, IT

VISION

The department of Information Technology aspires to become a school of excellence in providing quality education, constructive research and professional opportunities in Information Technology.

MISSION

- To provide academic programs that engage, enlighten and empower the students to learn technology through practice, service and outreach
- To educate the students about social responsibilities and entrepreneurship
- To encourage research through continuous improvement in infrastructure, curriculum and faculty development in collaboration with industry and institutions.

PROGRAM SPECIFIC OUTCOMES (PSOs)

The Program Specific Outcomes of Information Technology Undergraduate Program are:

PSO1: IT Systems Innovation and Management

Graduates will design and manage secure, scalable IT systems that address real-world challenges with technological and social relevance.

PSO2: Emerging Tech for Business Impact

Graduates will apply cutting-edge technologies to create innovative, ethical solutions that drive business transformation and community advancement.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will demonstrate career progression in Information and Communication Technologies by acquiring higher qualifications or industry certifications and advancing in professional roles.

PEO2: Graduates will be leaders in their chosen field

PEO3: Graduates will utilize the acquired technical skills and knowledge for the benefit of society

Signature of BOS Chairman, IT

PROGRAM OUTCOMES (POs)

Graduates of the Information Technology Undergraduate Program should have the ability to:

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions.

PO5: Engineering Tool Usage: Create, select, and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems.

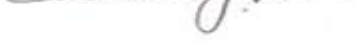
PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture, and environment.

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.

PO8: Individual and Collaborative Teamwork: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

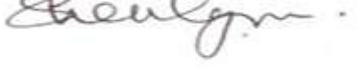
PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences.

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.



Signature of BOS Chairman, IT

PO11: Life-Long Learning: Recognize the need for and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.



Signature of BOS Chairman, IT

KUMARAGURU COLLEGE OF TECHNOLOGY

INFORMATION TECHNOLOGY
REGULATION 2024
B. Tech. Information Technology – Curriculum

2026 Batch Structure

Semester I

S. No	Course code	Course Title	Course Mode	Course Type	L	T	P	J	C
1	24HSI108 / 24HSI109 /	Essential English for Everyday Communication/ Professional Communication for Academic and Workplace/	Theory	HS	1	0	2	0	2
	24HSJ110	Project-Based English for Fluency and Impact	Practical		0	0	0	4	
2	24MAI118	Linear Algebra and Calculus for Computing	Embedded	BS	3	0	2	0	4
3	24PHI108	Physics of Computing and Emerging Technologies	Embedded	BS	3	0	2	0	4
4	24CSI022	Programming in C	Embedded	ES	1	0	4	0	3
5	24CSI023	Digital Logic Circuits	Embedded	ES	3	0	2	0	4
6	24HST101	Heritage of Tamils	Theory	HS	1	0	0	0	1
7	24INP102	Innovation Practicum - 1	Practical	ES	0	0	2	0	1
8	24HSP111	Holistic Wellness - 1	Practical	HS	0	0	2	0	1
9	24INO101	FCLF- General Stack-1	Practical	OE	0	0	2	0	1
								Total Credits	21
								Total Contact Hours/week	30

Semester II

S.No	Course code	Course Title	Course Mode	Course Type	L	T	P	J	C
1	24HSP011	Strategic Conversation Skills	Practical	HS	0	0	2	0	1
2	24MAI128	Multivariate Calculus and Statistical Forecasting	Embedded	BS	3	0	2	0	4
3	24CSI007	Python for Computational Problem Solving	Embedded	ES	1	0	4	0	3
4	24CSI024	Data Structures and Algorithms	Embedded	PC	2	0	4	0	4
5	24ITT203	Computer Organization and Architecture	Embedded	PC	3	0	0	0	3

Signature of BOS Chairman, IT

6	24HST102	Tamils and Technology	Theory	HS	1	0	0	0	1
7	24INP103	Innovation Practicum- 2	Practical	ES	0	0	2	0	1
8	24HSP112	Holistic Wellness-2	Practical	HS	0	0	2	0	1
9	24INO102	FCLF- General Stack-2	Practical	OE	0	0	2	0	1
10	24CST028	Emerging Technologies	Theory	HS	1	0	0	0	1
								Total Credits	**20**
								Total Contact Hours/week	**31**

Semester III

S.No	Course code	Course Title	Course Mode	Course Type	L	T	P	J	C
1	24MAI234	Computational Probability and Statistics	Embedded	BS	3	0	2	0	4
2	24HSP012	Developing Collaborative Discussion and Presentation Skills	Practical	PC	0	0	2	0	1
3	24ITI005	Operating Systems	Embedded	PC	3	0	2	0	4
4	24CSI025	Object Oriented Programming	Embedded	PC	1	0	4	0	3
5	24CSI014	Design and Analysis of Algorithms	Embedded	PC	3	0	2	0	4
6	24CSI030	Foundations of Web Development and UI/UX	Embedded	PC	1	0	4	0	3
7	24CSI009	Database Management Systems	Embedded	PC	3	0	2	0	4
8	24INP201	Innovation Practicum- 3	Practical	ES	0	0	2	0	1
9	24INOXYY	FCLF- General Stack-3	Practical	OE	0	0	2	0	1
10	24INM201	Universal Human Values-II	Theory	HS	1	0	0	0	1
								Total Credits	**26**
								Total Contact Hours/week	**32**

Semester IV

S.No	Course code	Course Title	Course Mode	Course Type	L	T	P	J	C
1	24MAT243	Discrete Mathematics and Optimization	Theory	BS	3	0	0	0	3
2	24CSI013	Full Stack Development	Embedded	PC	3	0	2	0	4
3	24___	Artificial Intelligence and Machine Learning	Embedded	PC	3	0	2	0	4
4	24CSI011	Computer Networks and Security	Embedded	PC	3	0	2	0	4

Signature of BOS Chairman, IT

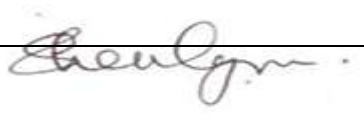
5	24CSI015	Software Engineering and Agile Practices	Embedded	PC	3	0	0	2	4
6	24HSP013	Mastering Career Readiness	Practical	HS	0	0	2	0	1
7	24INP202	Innovation Practicum – 4	Practical	ES	0	0	2	0	1
								Total Credits	**21**
								Total Contact Hours/week	**27**

Semester V

S.No	Course code	Course Title	Course Mode	Course Type	L	T	P	J	C
1	24ITI006	Big Data Technology	Embedded	PC	3	0	2	0	4
2	24ITI007	Mobile Application Development	Embedded	PC	2	0	2	0	3
3	24____	Cloud and DevOps	Embedded	PC	3	0	0	2	4
4	24HSTXYY	Foreign Language	Theory	HS	0	0	4	0	2
5	24------	Professional Elective I	Theory	PE	3	0	0	0	3
6	24------	Professional Elective II	Theory	PE	3	0	0	0	3
7	24ITOXYY	Open Elective 2	Theory	OE	3	0	0	0	3
8	24INM202	Environmental Science and Sustainability	Embedded	HS	1	0	2	0	2
								Total Credits	**24**
								Total Contact Hours/week	**32***

Semester VI

S.No	Course code	Course Title	Course Mode	Course Type	L	T	P	J	C
1	24____	Generative AI and Agentic AI Systems	Embedded	PC	3	0	2	0	4
2	24CST031	Foundations of Blockchain Technology	Theory	PC	3	0	0	0	3
3	24------	Professional Elective III	Theory	PE	3	0	0	0	3
4	24------	Professional Elective IV	Theory	PE	3	0	0	0	3
5	24------	Professional Elective V	Theory	PE	3	0	0	0	3
6	24------	Professional Elective VI	Theory	PE	3	0	0	0	3
7	24ITOXYY	Open Elective 3	Theory	OE	3	0	0	0	3
8	24INMXYY	Constitution of India	Theory	HS	2	0	0	0	0
								Total Credits	**22**
								Total Contact Hours/week	**26***

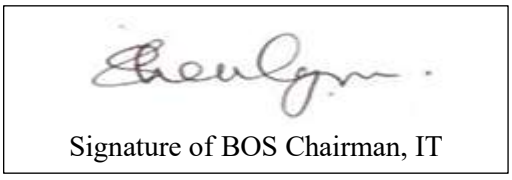


Signature of BOS Chairman, IT

Semester VII									
S.No	**Course code**	**Course Title**	**Course Mode**	**Course Type**	**L**	**T**	**P**	**J**	**C**
1	24ITOXYY	Open Elective 4	Theory	OE	3	0	0	0	3
2	24------	Professional Elective VII	Theory	PE	3	0	0	0	3
3	24------	Professional Elective VIII	Theory	PE	3	0	0	0	3
4	24INM102	Indian Knowledge System in Science and Engineering	Theory	HS	1	0	0	0	1
5	24ITJ409	Project Phase-I	Project	PRJ	0	0	0	6	3
6	24ITJ415	Professional Internship/ Certification	Project	PRJ	0	0	0	6	3
								Total Credits	**16**
								Total Contact Hours/week	**26***
Semester VIII									
S.No	**Course code**	**Course Title**	**Course Mode**	**Course Type**	**L**	**T**	**P**	**J**	**C**
1	24ITJ411	Project Phase-II	Project	PRJ	0	0	0	24	12
								Total Credits	**12**
								Total Contact Hours/week	**24**
								Grand Total Credits	**160**

Signature of BOS Chairman, IT

SEMESTER I



Signature of BOS Chairman, IT

24CSI022	PROGRAMMING IN C	L	T	P	J	C
		1	0	4	0	3
ES		SDG	9			

Pre-requisite courses	Nil	Data Book / Codes / Standards (If any)	Nil

Course Objectives:	The purpose of taking this course is to:
1	Gain proficiency in logical reasoning and problem-solving using algorithms and flowcharts.
2	Acquire knowledge of fundamental programming constructs in C, including data types, variables, control structures, and input/output operations.
3	Attain the capability to address real-world problems by utilizing arrays, strings, and modular programming with functions.
4	Introduce concepts of pointers, storage classes, and structures as tools for effective data representation and management.
5	Provide hands-on experience in building C-based applications that incorporate file operations, debugging strategies, and best coding practices.

Course Outcomes:	After successful completion of this course, the students shall be able to	Bloom's Taxonomy Level (BTL)
CO 1	Illustrate the algorithmic problem-solving techniques using flowcharts and pseudocode.	U
CO 2	Solve computational problems by applying decision-making constructs and iterative controls using basic C programming.	Ap
CO 3	Apply arrays, strings, and functions to solve programming challenges using modular programming.	Ap
CO 4	Utilize pointers, storage classes, and structures for efficient data manipulation and memory management, incorporating file operations in problem-solving scenarios.	Ap
CO 5	Apply debugging techniques and tools to identify, trace, and correct syntax, logical, and runtime errors in C programs.	Ap

	Program Outcomes (PO) (Strong-3, Medium – 2, Weak-1)											Program Specific Outcomes (PSO)
	1	2	3	4	5	6	7	8	9	10	11	

Signature of BOS Chairman, IT

Course Outcomes (CO)	Engineering Knowledge	Problem Analysis	Design/Development of Solutions	Conduct Investigations of Complex	Engineering Tool Usage	The Engineer and The World	Ethics	Individual and Collaborative Team work	Communication	Project Management and Finance	Life-Long Learning	PSO-1	PSO-2
1	2				2							3	
2	3	2										3	
3	3		2									3	
4	3		2									3	
5		3			2							3	

Course Content

INTRODUCTION TO PROBLEM SOLVING AND C BASICS Introduction to computer problem-solving – The problem-solving aspect- Top-down design -Building blocks, pseudocode notation, flowcharts – Implementation of algorithms –Simple algorithmic problem-solving scenarios – C programming: data types, variables, operators and expressions, input and output operations, control flow using if and switch statements. **Practical Component** Drawing flowcharts, writing pseudocode for algorithmic problem scenarios, Programs using arithmetic, logical, and relational operators, Simple logic-based programs using conditionals	**3 Hours** **12 Hours**
ARRAYS AND STRINGS Looping and Control Statements: for, while, break, continue -Nested Loops - Defining an array - Processing an array - Multidimensional Arrays-Character Arithmetic - Defining a string - NULL character - Initialization of Strings - Reading and Writing Strings - Processing Strings -Searching and Sorting of Strings. **Practical Component** Pattern printing using loops - Programs for defining and processing 1D and 2D arrays, Searching (linear and binary search), Sorting (bubble sort, selection sort), Matrix operations (addition, subtraction, transpose), String handling with/without standard functions	**3 Hours** **12 Hours**

Signature of BOS Chairman, IT

FUNCTIONS AND STORAGE CLASSES Defining a function - Accessing a function - Function prototypes -User defined Function – Actual and Formal parameters - Passing arrays to functions - Function with string - Recursion - Storage classes **Practical Component** Programs using functions with arguments and return values, Recursive functions (e.g., factorial, Fibonacci, sum of numbers), Passing arrays and strings to functions, Programs demonstrating storage classes: auto, static, extern, register	**3 Hours** **12 Hours**
POINTERS, STRUCTURES AND UNIONS Pointer Fundamentals - Pointer Declaration - Passing Pointers to a Function, call by value, call by reference- Pointers and arrays - Operations on pointers - Dynamic memory allocation- Structures and Unions: Defining a Structure - Processing a Structure - User defined data types (Typedef) -Unions **Practical Component** Using Pointers - operations and arithmetic, swapping values using pointers, array manipulation- searching. Dynamic memory allocation using malloc, calloc, and free, Programs using structures and unions, Accessing structure members using pointers	**3 Hours** **12 Hours**
FILES AND DEBUGGING File Operations -Open,close, read,write- Processing a data file -Unformatted data files - Concept of binary files - Accessing a file randomly using fseek- Types of errors: Syntax, Logical, Runtime – Debugging techniques: Tracing with printf(), setting breakpoints, step-by-step execution – Introduction to debugging tools: gcc -g, GDB- Integrated Development Environment – Best practices in debugging C programs. **Practical Component** Text and Binary File read/write operations with Structures and Union – Search and Replace using fseek, Debugging C programs using printf() tracing and GDB, Identifying and fixing errors- integer division, incorrect loop condition, reading a non-existing file, Divide by zero, Accessing array out of bounds.	**3 Hours** **12 Hours**

Theory Hours: 15	**Tutorial Hours:**	**Practical Hours: 60**	**Project Hours:**	**Total Hours: 75**

Learning Resources
Textbooks
1. Yashavant Kanetkar, Let Us C – Authentic Guide to C Programming Language, 20th Edition, BPB Publications, 2024.

Signature of BOS Chairman, IT

2. Byron S Gottfried and Jitendar Kumar Chhabra, “Programming with C”, Tata McGraw Hill Publishing Company, Third Edition, New Delhi, 2018.
3. E. Balagurusamy, Programming in ANSI C, 8th Edition, McGraw-Hill Education, 2019.

Reference books/ Web Links

1. R. G. Dromey, How to Solve It by Computer, Pearson Education, First Edition, New Delhi, 2008.
2. Pradip Dey and Manas Ghosh, Programming in C, 2nd Edition, Oxford University Press, 2011.
3. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, 2nd Edition, Pearson Education, 2006.
4. Ashok N. Kamthane, Computer Programming, Pearson Education, 2018.
5. Reema Thareja, Programming in C, 2nd Edition, Oxford University Press, 2011.

Online Resources

1. https://www.geeksforgeeks.org/c-programming-language/
2. https://www.w3schools.com/c/
3. https://www.programiz.com/c-programming
4. https://www.coursera.org/learn/c-for-everyone
5. https://www.codingninjas.com/studio/library/c-programming
6. https://www.w3schools.com/c/c_debugging.php
7. https://www.onlinegdb.com/

Assessment (Embedded course)

MCQ, Experimental Cycle tests and End Semester Examination (Lab)

Course Curated by

Expert(s) from Industry	Expert(s) from Higher Education Institution	Internal Expert
-	-	Ms. R. Nivetha Department of Computer Science and Engineering

Recommended by BoS on	06.06.2026		
Academic Council Approval	No: 30	**Date**	19.06.2026

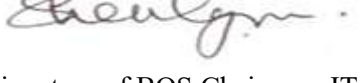
Signature of BOS Chairman, IT

24CSI023	DIGITAL LOGIC CIRUCITS	L	T	P	J	C
		3	0	2	0	4
ES		SDG	9			

Pre-requisite courses	Nil	Data Book / Codes / Standards (If any)	

Course Objectives:	The purpose of taking this course is to:
1	Provide a strong foundation in digital systems, number systems, binary codes, and Boolean algebra.
2	Develop the ability to represent and simplify Boolean functions using Karnaugh map techniques and universal gates.
3	Enable students to analyze and design combinational logic circuits for arithmetic and data processing applications.
4	Equip students with the knowledge of sequential logic circuits, including flip-flops, counters, and registers.
5	Introduce digital logic families, programmable logic devices, and hardware description languages for digital system implementation.

Course Outcomes:	After successful completion of this course, the students shall be able to	Bloom's Taxonomy Level (BTL)
CO 1	Apply the concepts of digital systems, number systems, binary codes, and Boolean algebra to solve problems in digital logic representation.	Ap
CO 2	Apply Boolean function representation and minimization techniques to obtain optimized logic expressions for digital circuit implementation.	Ap
CO 3	Design combinational logic circuits for arithmetic and data processing applications based on given specifications.	Ap
CO 4	Demonstrate sequential logic circuit concepts including flip-flops, counters, and registers in digital system operations.	Ap
CO 5	Illustrate digital system realization using logic families, programmable logic devices, and basic VHDL modeling techniques.	Ap



Signature of BOS Chairman, IT

	Program Outcomes (PO) (Strong-3, Medium – 2, Weak-1)											Program Specific Outcomes (PSO)	
	1	2	3	4	5	6	7	8	9	10	11		
Course Outcomes (CO)	Engineering Knowledge	Problem Analysis	Design/Development of Solutions	Conduct Investigations of Complex Problems	Engineering Tool Usage	The Engineer and The World	Ethics	Individual and Collaborative Team	Communication	Project Management and Finance	Life-Long Learning	PSO-1	PSO-2
1	3												
2	3	3											
3	3	2	3									3	
4	3	2										3	
5	3		3										

Course Content

DIGITAL SYSTEMS AND BOOLEAN ALGEBRA Introduction to Digital Systems, Analog and Digital Signals, Number Systems, Number Base Conversions, Complements-Signed Binary Numbers, Binary Codes, Boolean Algebra and its Properties, Boolean Functions, Simplification of Boolean Functions using Boolean Algebra. **Practical Component:** 1. Study of logic gates. 2. Verify Boolean algebra properties such as commutative, associative, distributive, and De Morgan's theorems using logic gates and truth tables.	**9 Hours** **4 Hours**
BOOLEAN FUNCTION REPRESENATTION AND MINIMIZATION Canonical Forms-Standard Forms of Boolean Function-Karnaugh Maps (2, 3 and 4 Variables), Don't care condition-Simplification of Boolean Functions using K-Maps, Universal Gates, Realization of Logic Gates using NAND and NOR Gates. **Practical Component:** 1. Implement Boolean functions using logic gates and validate the outputs with truth tables. 2. Design logic circuits for simple real-world problem statements by deriving Boolean expressions, simplifying them using Karnaugh maps, and verifying the outputs.	**9 Hours** **4 Hours**
COMBINATIONAL LOGIC CIRCUITS Combinational Circuit-Analysis of Combinational Logic Circuits, Design of Combinational Logic Circuits.Half Adder, Full Adder,Half Subtractor, Full Subtractor-Magnitude Comparator,Code ~~Converters, Decoders, Encoders,~~ Multiplexers,	**9 Hours**

Signature of BOS Chairman, IT

Demultiplexers-Parity checker and Generator-Applications of Combinational Logic Circuits. **Practical Component:** 1. Design and construct half adders, full adders, half subtractors, and full subtractors. 2. Develop a code converter circuit using logic gates. 3. Design and implement decoder and encoder circuits. 4. Construct multiplexers and demultiplexers and incorporate them into circuit designs. 5. Design and implement a circuit for a parity generator and parity checker using logic gates.	**12 Hours**
SEQUENTIAL LOGIC CIRCUITS Introduction to Sequential Logic Circuits, Latches and Flip-Flops, SR Flip-Flop, JK Flip-Flop, D Flip-Flop, T Flip-Flop, Edge Triggering and Level Triggering, Flip-Flop Conversions, Asynchronous Counters, Synchronous Counters, Decade Counter, Ring Counter, Registers, Shift Registers (SISO, SIPO, PISO, PIPO), Applications of Sequential Logic Circuits. **Practical Component:** 1. Implement and test various types of shift registers. 2. Design and build a synchronous and asynchronous counter.	**9 Hours** **6 Hours**
LOGIC FAMILIES AND PROGRAMMABLE DEVICES Digital Logic Families, RTL, DTL, TTL, ECL and CMOS Logic Families, Characteristics and Comparison of Logic Families, Programmable Logic Devices, Programmable Logic Array (PLA), Programmable Array Logic (PAL), Complex Programmable Logic Devices (CPLD), Field Programmable Gate Arrays (FPGA), Implementation of Combinational Logic Circuits using PLA and PAL, Introduction to Hardware Description Languages, Basics of VHDL Modeling and Simulation for Arithmetic Circuits. **Practical Component:** 1. Implement and simulate arithmetic circuits using VHDL models.	**9 Hours** **4 Hours**

Theory Hours:45	Tutorial Hours:	Practical Hours:30	Project Hours:	Total Hours:75

Learning Resources*
Textbooks
1. M. Morris R. Mano, Michael D. Ciletti, "Digital Design: With an Introduction to the Verilog HDL, VHDL, and System Verilog", 6th Edition, Pearson, 2018.
Reference Books
1. C. H. Roth Jr., Larry L. Kinney "Fundamentals of Logic Design", 8th Edition, Cengage Learning, 2018. 2. John F. Wakerly, "Digital Design: Principles and Practices", 5th Edition, Pearson, 2018.

Signature of BOS Chairman, IT

Online Resources
1. https://www.coursera.org/learn/digital-systems 2. https://nptel.ac.in/courses/108105113
Weblinks:
1. https://www.nesoacademy.org/ee/05-digital-electronics

Assessment (Embedded course)
SA-I, SA-II, Activity and Learning Task(s), Mini project, MCQ, End Semester Examination (ESE) Lab Workbook, Experimental Cycle tests, viva-voce

Course Curated By			
Expert(s) from Industry	**Expert(s) from Higher Education Institutions**	**Internal Expert(s)**	
		Ms.P.Anitha Assistant Professor Dept of CSE	
Recommended by BoS on	06.06.2026		
Academic Council Approval	No: 30	**Date**	19.06.2026

Signature of BOS Chairman, IT

Semester – II

Signature of BOS Chairman, IT

24CSI007	PYTHON FOR COMPUTATIONAL PROBLEM SOLVING	L	T	P	J	C
		1	0	4	0	3
PC		SDG	9			

Pre-requisite courses	NIL	Data Book / Code book (If any)	NIL

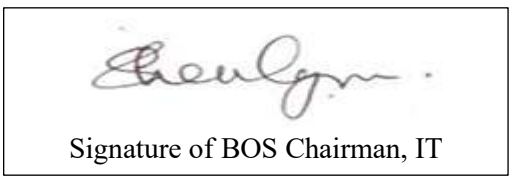
Course Objectives:	
The purpose of taking this course is to:	
1	Introduce the basics of Python, including data types, operators, and control structures, with a focus on practical implementation.
2	Develop proficiency in iterative statements, string handling, and fundamental data structures like lists, tuples, and dictionaries through extensive lab work.
3	Provide foundational knowledge on modules, functions, object-oriented programming (OOP) concepts, and file handling.
4	Familiarize students with exception handling and Python's standard libraries like NumPy and Pandas for data manipulation.
5	Enable students to apply Python concepts to solve real-world problems through practical, hands-on experiments.

Course Outcomes		
After successful completion of this course, the students shall be able to		Revised Bloom's Taxonomy Levels (RBT)
CO1	Apply Python syntax, data types, operators, and basic control structures through practical programs.	Uzzzzzz
CO2	Apply conditional and iterative constructs for solving logical problems.	Ap
CO3	Implement modular programming concepts, handle strings, and utilize data structures efficiently.	Ap
CO4	Demonstrate object-oriented programming, file handling, and exception handling in Python.	Ap
CO5	Utilize libraries like NumPy and Pandas for data analysis and manipulation.	Ap

Signature of BOS Chairman, IT

Course Outcomes (CO)	Program Outcomes (PO) (Strong-3, Medium – 2, Weak-1)											Program Specific Outcomes (PSO)	
	1	2	3	4	5	6	7	8	9	10	11		
	Engineering Knowledge	Problem Analysis	Design/Development of Solutions	Conduct Investigations of	Engineering Tool Usage	The Engineer and The World	Ethics	Individual and Collaborative	Communication	Project Management and Finance	Life-Long Learning	PSO-1	PSO-2
1	3				2							3	
2	3	2			2							3	
3		2	2		3							3	
4			2		3							3	
5		2			3							3	

Course Content	
Python Basics and Control Structures Python Basics: Python syntax – Variables - Data Types - Type Casting - Arithmetic & Logical Operators. Conditional Branching Statements: if, if-else, if-elif-else, nested conditions. Iterative Statements: for loop, while loop, Loop Control Statements. **Practical Component:** Implement basic Python programs to demonstrate data types, operators, and input/output operations. Develop programs using all forms of conditional statements. Write extensive Python scripts using for and while loops to generate complex patterns, various series like Fibonacci, arithmetic progression, and solve simple mathematical/logical problems.	**3 Hours** **10 Hours**
Python Functions, Strings, and Data Structures Defining functions, Function arguments - Return values - Anonymous Lambda Functions. Modules - built-in and user-defined modules - String Handling: String slicing, concatenation, formatting, and methods - List: Creating lists - List indexing - List methods - List comprehension. Tuple - Characteristics of tuples - Tuple indexing, Tuple methods. Dictionary: Key-value pairs, Dictionary methods. Sets in Python: Create, Add, Remove with Examples. Python Regular Expressions (RegEx) basics	**3 Hours**



Signature of BOS Chairman, IT

Practical Component: Create Python scripts to demonstrate advanced string manipulation techniques, including RegEx for basic pattern matching. Perform complex operations on lists, tuples, sets, dictionaries, and combinations of all datatypes using various methods. Develop and utilise user-defined modules.	**10 Hours**
Object-Oriented Programming OOP concepts - Creating classes and objects - Self parameter - __init__ method. Inheritance & Polymorphism: Types of inheritance, Method Overriding - Concept of Method Overloading in Python. **Practical Component:** Develop a robust OOP-based application to practically demonstrate the core concepts: creating multiple classes and objects, implementing all major types of inheritance, and showcasing polymorphism through method overriding. Implement a solution to simulate method overloading using variable arguments or default parameters	**3 Hours** **10 Hours**
File Handling and Exception Handling File Handling: Reading and writing files, File modes - Working with text and CSV files. Exception Handling: Understanding exceptions - Handling exceptions using try-except-finally - Raising custom exceptions **Practical Component:** Read and write large amounts of data to text and CSV files using different Python file handling modes. Create a practical utility that processes data from a file, writes results to another file, and is fully protected using robust exception handling. Implement and raise meaningful custom exceptions for validation or specific error conditions	**3 Hours** **10 Hours**
Python Libraries – NumPy and Pandas NumPy Library: Introduction to NumPy, Arrays, Array creation and manipulation, Array operations. Analyzing different pipeline structures involving storage - processing, ML components based on requirements. Pandas Library: Series: Creating Series – Indexing - Operations on Series. Data Frames: Creating Data Frames, Data Manipulation. Handling Missing Data: dropna(), fillna(), isnull() functions **Practical Component:** Perform an extensive array creation, manipulation, and mathematical operations on NumPy arrays. Manipulate real-world or mock datasets using Pandas Series and Data Frames, including data cleaning, filtering, sorting, and handling missing values using appropriate functions	**3 Hours** **10 Hours**

Signature of BOS Chairman, IT

Theory Hrs: 15	Tutorial Hrs:	Practical Hrs: 50	Project Hrs:	Total Hrs: 65

Learning Resources
Textbooks:
1. Mark Lutz, “Learning Python,” O’Reilly Media, 6th Edition, 2025. 2. Y. Daniel Liang, “Introduction to Programming Using Python,” Pearson, 1st Edition, 2017.
References:
1. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist,” Green Tea Press, 2nd Edition, 2015. 2. Zed A. Shaw, “Learn Python the Hard Way,” Addison-Wesley, 5th Edition, 2024.
Online Educational Resources:
1. https://docs.python.org/3/tutorial/index.html 2. https://learn.microsoft.com/en-us/shows/intro-to-python-development/ 3. https://www.udacity.com/course/introduction-to-python--ud1110 4. https://www.geeksforgeeks.org/courses/python-course-certification-free 5. https://onlinecourses.nptel.ac.in/noc24_cs57/preview 6. https://www.coursera.org/projects/python-for-data-analysis-numpy

Assessment (Embedded course)
MCQ, Experimental Cycle tests and End Semester Examination (Lab)

Course Curated by

Expert(s) from Industry	Expert(s) from Higher Education Institution	Internal Expert
-	-	Dr. C Rajankrupa Ms. R K Shalini
Recommended by BoS on		
Academic Council Approval	**Date**	

Signature of BOS Chairman, IT

24CSI024	DATA STRUCTURES AND ALGORITHMS	L	T	P	J	C
		3	0	2	0	4
ES		SDG	9			

Pre-requisite courses	Nil	Data Book / Code book (If any)	-

Course Objectives:

The purpose of taking this course is to:

1	Provide foundational knowledge of data structures, searching and sorting algorithms, string processing techniques, and algorithm analysis for developing efficient software solutions.
2	Develop the ability to implement and analyze linear data structures such as stacks, queues, and linked lists for real-world applications.
3	Apply non-linear data structures including trees and heaps for efficient storage, searching, and data organization.
4	Analyze graph algorithms for solving complex computational and optimization problems.
5	Understand and apply hashing techniques, hash functions, and collision resolution methods for efficient data storage, retrieval, and management in computing applications.

Course Outcomes

After successful completion of this course, the students shall be able to		Revised Bloom's Taxonomy Levels (RBT)
CO 1	Apply searching, sorting, and string processing algorithms using appropriate data structures and analyze their efficiency using asymptotic notations.	Ap
CO 2	Apply linear data structures such as arrays, stacks, queues, and linked lists for solving computational problems.	Ap
CO 3	Implement and analyze tree-based data structures including BST, AVL trees, and heaps for efficient data processing.	Ap
CO 4	Apply graph algorithms such as BFS, DFS, shortest path, and minimum spanning tree algorithms for problem solving.	Ap
CO5	Design efficient hashing techniques with suitable collision resolution methods and evaluate their performance in data storage and retrieval applications.	Ap

	Program Outcomes (PO) (Strong-3, Medium – 2, Weak-1)											Program Specific Outcomes (PSO)	
	1	2	3	4	5	6	7	8	9	10	11		
Course Outcomes (CO)	Engineering Knowledge	Problem Analysis	Design/Development of Solutions	Conduct Investigations of Complex Problems	Engineering Tool Usage	The Engineer and The World	Ethics	Individual and Collaborative Team work	Communication	Project Management and Finance	Life-Long Learning	PSO-1	PSO-2
1	3			2							2	3	2
2	3	3	2									3	2
3	3	3	2	2							2	3	3
4	3	3		2							2	3	3

Signature of BOS Chairman, IT

5	2	2											2

INTRODUCTION TO DATA STRUCTURES AND ALGORITHMS **Theory Component:** Introduction to Data Structures and Algorithms, Importance of data structures in programming and software development, data types vs data structures, Abstract Data Types (ADTs), arrays, two-pointer technique, sliding window, prefix sums. Searching: Linear, Binary search. Sorting Algorithms: Bubble Sort, Selection sort, insertion sort. Merge sort. string algorithms, anagram problems, palindrome problems. Algorithm analysis: Big O, Big Theta, Big Omega notations **Practical Component** Implementation of List ADT using arrays (insertion, deletion, Linear and Binary Search, Sorting)	**6 Hours** **12 Hours**
LINEAR DATA STRUCTURES **Theory Component:** Stacks: Implementation of stack and Applications (infix to postfix, expression evaluation, parenthesis matching) Queues: Implementation of Queue, Circular Queue, Applications, Linked Lists (Single, Double, and Circular linked implementation), Linked list implementation of Stack & Queue **Practical Component** Implement the applications of Linear Data structures: reverse and cycle detection problems, solve balanced parentheses and next greater element problems	**6 Hours** **12 Hours**
NON-LINEAR DATA STRUCTURES TREES **Theory Component:** Trees: Binary Trees, Binary Search Trees, Tree Traversals, AVL Trees, Heap tree, Complete Binary Tree, **Practical Component** Implement BST and AVL trees, solve tree traversal problems, implement heap operations, solve top-K problems.	**6 Hours** **12 Hours**
NON-LINEAR DATA STRUCTURES GRAPHS **Theory Component:** Graphs: Terminologies, Representation (Adjacency Matrix, List), Graph Traversal (BFS, DFS), Spanning Trees Prim's algorithm, Kruskal's algorithm, Shortest Path Algorithms (Dijkstra, Floyd-Warshall). **Practical Component:** Implement graph traversals, solve shortest path problems, implement MST algorithms, solve advanced graph problems including course scheduling and cheapest flight problems	**6 Hours** **12 Hours**
HASHING TECHNIQUES **Theory Component:** Hashing: Hash Functions, Collision Resolution Techniques, Linear probing, Quadratic probing, random probing, Double hashing and rehashing. **Practical Component** Implementation of Hash hashing techniques and collision resolution techniques	**6 Hours** **12 Hours**

AI Tools to be Used • GitHub Copilot • Agent Smith • ChatGPT API / LLM-assisted debugging • Online Competitive Coding Platforms • Complexity Visualizers	

Signature of BOS Chairman, IT

Theory Hrs: 45	Tutorial Hrs:0	Practical Hrs: 30	Project Hrs:0	Total Hrs: 75

Learning Resources

Textbooks

1. Horowitz, Ellis, Sartaj Sahni, and Susan Anderson-Freed. Fundamentals of Data Structures in C. Universities Press, Hyderabad (2021).
2. Tenenbaum, Aaron M., Yedidyah Langsam, and Moshe J. Augenstein. Data Structures Using C. Pearson, New York (2021).
3. Weiss, Mark Allen. Data Structures Using C. Pearson Education Asia, Singapore (2007).

Reference books

1. Tremblay, Jean Paul, and Paul G. Sorenson. An Introduction to Data Structures with Applications. Tata McGraw-Hill, New Delhi (2014).
2. Mehlhorn, Kurt, and Peter Sanders. Algorithms and Data Structures: The Basic Toolbox. Springer, Berlin (2011).
3. Aho, Alfred V., John E. Hopcroft, and Jeffrey D. Ullman. Data Structures & Algorithms. Pearson Education, New Delhi (2009).

Online Resources (Weblinks)

1. https://open.umn.edu/opentextbooks/textbooks/an-open-guide-to-data-structures-and-algorithms
2. https://www.oreilly.com/library/view/data-structures-and/9780133437483/
3. https://www.khanacademy.org/computing/computer-science/algorithms
4. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/
   https://www.geeksforgeeks.org/data-structures/

Assessment (Embedded course)

SA-I, SA-II, Activity and Learning Task(s), Mini project, MCQ, End Semester Examination (ESE) Lab Workbook, Experimental Cycle tests, viva-voce

Course Curated by

Expert(s) from Industry	Expert(s) from Higher Education Institution	Internal Expert(s)

Recommended by BoS on				
Academic Council Approval		**Date**		

Signature of BOS Chairman, IT

Signature of BOS Chairman, IT

24ITT203	COMPUTER ORGANIZATION AND ARCHITECTURE (Common to CS, IT)	L	T	P	J	C
		3	0	0	0	3
PC		SDG	4,9			

Pre-requisite courses	Nil	Data Book / Codes / Standards (If any)	Nil

Course Objectives:

The purpose of taking this course is to:

1	Introduce the basic structure and functional components of a computer system, including its evolution, Von Neumann architecture, and performance parameters.
2	Explain the internal organization of a basic computer, focusing on instruction formats, addressing modes, and the distinction between CISC and RISC architectures.
3	Explore arithmetic and logic unit (ALU) operations including integer and floating-point computations.
4	Examine memory hierarchy, cache, and I/O system designs and their impact on performance.
5	Explore modern advancements in computer architecture, including multicore processors, parallel architectures, and multiprocessor systems with interconnection structures and synchronization techniques.

Course Outcomes:

After successful completion of this course, the students shall be able to		Bloom's Taxonomy Level (BTL)
CO 1	Apply computer organization concepts to explain the functional units, performance characteristics and processor architectures.	Ap
CO 2	Solve computational problems by applying arithmetic algorithms for addition, subtraction, multiplication, and division with accuracy and efficiency.	Ap
CO 3	Utilize the concepts of instruction execution, multiple bus organization, and control unit design to implement control logic using hardwired and microprogrammed approaches.	Ap

Signature of BOS Chairman, IT

CO 4	Apply the principles of I/O mechanisms and memory subsystems to interpret data transfer processes and memory management in computer systems.	AP
CO 5	Use the principles of parallel processing and multicore architectures to explain data hazards, instruction hazards, and performance issues in modern computing systems	Ap

	Program Outcomes (PO) (Strong-3, Medium – 2, Weak-1)											Program Specific Outcomes (PSO)	
	1	2	3	4	5	6	7	8	9	10	11		
Course Outcomes (CO)	Engineering Knowledge	Problem Analysis	Design/Development of Solutions	Conduct Investigations of Complex Problems	Engineering Tool Usage	The Engineer and The World	Ethics	Individual and Collaborative Team work	Communication	Project Management and Finance	Life-Long Learning	PSO-1	PSO-2
1	3	2											
2	3	2											
3	3	2	2									2	
4	3	2										2	
5	3	2	2									1	

Course Content

FUNDAMENTALS OF COMPUTER ORGANIZATION Organization and Architecture- Structure and Function- Functional Units-Basic Operations and Concepts-Bus Structure-Performance. Memory Locations and Addresses - Memory Operations - Instruction and Instruction Sequencing - Addressing Modes - Basic I/O Operations. Case Study: Evolution of Intel, ARM Architectures/Processor Families	**7 Hours**
COMPUTER ARITHMETIC Addition and Subtraction of Signed Numbers-Design of Fast Adders-Multiplication of Positive Numbers-Signed Operand Multiplication-Fast Multiplication-Integer Division-Floating Point Numbers and Operation.	**11 Hours**
BASIC PROCESSING UNIT	**9 Hours**

Signature of BOS Chairman, IT

Fundamental Concepts - Execution of a Complete Instruction - Multiple Bus Organization - Hardwired Control – Microprogrammed Control – Microinstructions-Microprogram Sequencing-Wide Branch Addressing	
I/O AND MEMORY The I/O Subsystem, -Programmed I/O- I/O Interrupts -Direct Memory Access (DMA) - The Components of Memory System – RAM Structure: The Logic Designer's Perspective – Memory Boards and Modules-Memory Hierarchy – The Cache- The Memory Subsystem.	**9 Hours**
PIPELING AND PARALLEL PROCESSING Basic Concepts - Data Hazards - Instruction Hazards – Influence on instruction sets - Data path and control considerations - Superscalar operation. Multiple Processor Organization-Types, Parallel Organization. Multicore Computers Introduction – Organization- Software and Hardware performance issues. Case Study: GPU Acceleration in Real-Time System.	**9 Hours**

Theory	Tutorial	Practical	Project	Total
Hours: 45	**Hours: 0**	**Hours: 0**	**Hours: 0**	**Hours: 45**

Learning Resources
Textbooks
1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Edition, McGraw-Hill, (2014). 2. Vincent P Heuring, Harry F Jordan, TG Venkatesh., Computer Systems Design and Architecture, 2nd Edition, Pearson, (2014). 3. William Stallings, Computer Organization and Architecture - Designing for Performance, 9th Edition, Prentice Hall, (2012).
Reference books/ Web Links
1. M. Moris Mano, Computer System Architecture, Revised 3rd Edition, Pearson, (2017). 2. John P.Hayes, Computer Architecture and Organization, 3rd Edition, McGraw Hill, (2002). 3. David A.Patterson and John L.Hennessy, Computer Organization and Design: The hardware / software interface, 5th Edition, Morgan Kaufmann, (2014). 4. Andrew S. Tanenbaum, Structured Computer Organization, 6th EditionPearson Education, (2016)
Online Resources (Weblinks)

Signature of BOS Chairman, IT

1. https://profile.iiita.ac.in/bibhas.ghoshal/teaching_coa_2021.html
2. https://www.cse.iitd.ac.in/~srsarangi/archbooksoft.html
3. https://onlinecourses.nptel.ac.in/noc22_cs88/preview
4. https://www.coursera.org/learn/comparch
5. https://www.udemy.com/course/computer-organization-and-architecture-j/?srsltid=AfmBOoo3L-F9CWPP3HioEoAzx57cX2iGGEU3Bi0UOLTD0aLRh_BAREfE&couponCode=ST21MT30625G1

Assessment (Theory course)
SA - I, SA - II, Activity and Learning Task(s), MCQ, End Semester Examination (ESE)

Course Curated by			
Expert(s) from Industry	**Expert(s) from Higher Education Institution**	**Internal Expert**	
-	-	Mr. M. Sathish, Department of Computer Science and Engineering	
Recommended by BoS on	09.05.2025		
Academic Council Approval	No: 28	**Date**	26.06.2025

Signature of BOS Chairman, IT

Signature of BOS Chairman, IT