GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

Department of Computer Science and Engineering

Course Structure and syllabus of B.Tech CSE 6th Semester

Sl. No	Course Code	Course Name	Hours per week			Credit
			L	T	P	C
1.	BCS23221T	Theory of Computation	3	1	0	4
2.	BCS23222T	Computer Networks	3	0	0	3
3.	BCS23222P	Computer Networks Lab	0	0	2	1
4.	BCS23223T	Cryptography & Data Security	3	0	0	3
5.	BCS23223P	Cryptography & Data Security Lab	0	0	2	1
6.	BCS23224T	Microprocessor	3	0	0	3
7.	DSE23220T	Image Processing	3	0	0	3
8.	DSE23221T	Speech & Natural Language Processing	3	0	0	3
9.	BCM25102T	Accountancy	2	0	0	2
10.	DCS23201R	Mini Project	0	0	6	3
TOTAL CREDIT			**20**	**1**	**10**	**23**

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

BCS23221T	**THEORY OF COMPUTATION**	**L**	**T**	**P**	**C**
		3	**1**	**0**	**4**

Pre-requisite: Discrete Mathematics

Course Objectives: After completing this course, students will be able to:

1. Define and differentiate between various types of formal languages, including regular languages, context-free languages, and context-sensitive languages.
2. Demonstrate an in-depth understanding of different automata models, such as finite automata, pushdown automata, and Turing machines, and analyze their computational capabilities.
3. Apply automata theory to design and analyze algorithms for language recognition, parsing, and other computational tasks.

Course Outcome:

CO1: Design automata and regular expressions to accept a certain word or generate a certain language. − **CREATE**
CO2: Construct context-free grammar and language and their different forms. − **APPLY**
CO3: Create push down automata as acceptors of context free languages. − **CREATE**
CO4: Develop Turing machines performing simple tasks. − **CREATE**

MODULE 1	**10 hours**

Introduction to the concept of Automata, Basics of Strings and Alphabets, Deterministic finite automata (DFA), Transition graphs, Problem solving using various DFAs, Solving various numerical problems on DFAs, Non-deterministic FA, Equivalence of DFA and NDFA, Minimization of finite automata, Mealy and Moore machines

MODULE 2	**6 hours**

Regular grammars & regular expressions, Solving various numerical problems on regular expressions, Equivalence between regular languages, properties of regular languages, Pumping lemma of regular sets, Using different examples prove whether a language is regular or not

MODULE 3	**12 hours**

Concept of context free grammar (CFG), it's importance, Derivation trees and sentential forms, Leftmost and rightmost derivation of strings, Constructing CFG for different languages, Constructing context-free languages for different grammars, Problem solving on different numerical questions on CFG & CFL, Ambiguity in context free grammars.

MODULE 4	**8 hours**

Push down automata (PDA), model definition and basic concepts, Acceptance of CFL by final state and acceptance by empty state and its equivalence, Solving different problems to show whether a language is accepted by PDA, Designing PDA for different languages and solving different problems.

MODULE 5	**9 hours**
Turing Machine (TM), definition, Model, design of TM, Designing TMs for different languages, Solving different problems to show whether a language is accepted by TM, Recursively enumerable languages Church's hypothesis, counter machine, Types of Turing machines, Decidability of problems, Universal Turing Machine	
Total Lecture hours	**45 hours**

Text Book

1. Mishra & Chandrasekharan, Theory of computer science: Automata language and computation, Prentice Hall of India, 3rd Ed, 2007.

2. P. Linz, Introduction to Formal Language and Computation, Narosa, 2nd Edition, 2006.

3. Nasir & Sirmani, A Text Book on Automata Theory, Cambridge University Press, 2008.

4. H. R. Lewis & C. H. Papadimitriou, Elements of the Theory of Computation, Prentice Hall of India, 2nd Edition, 2006.

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

BCS23222T	**COMPUTER NETWORKS**	L	T	P	C
		3	0	0	3

Pre-requisite: Basics of Data Communication

Course Objectives:

After completing this course, the student will be able to:

1. **Introduce** the basic concepts of computer networks, network architectures, reference models (OSI and TCP/IP), and common networking terminology used in Internet-based systems.
2. **Explain** the principles of data transmission, including physical media, signal encoding, switching techniques, network topologies, and data link layer protocols for reliable communication.
3. **Develop understanding of** network layer and transport layer functionalities such as routing algorithms, IP addressing schemes, subnetting, congestion control, and transport protocols (TCP and UDP).
4. **Familiarize** students with application layer protocols and services such as HTTP, DNS, e-mail, peer-to-peer systems, and multimedia streaming over best-effort IP networks.

Course Outcome:

After successful completion of the course, the students will learn

CO1 (Understand – Bloom's Level: L2): Explain the fundamental concepts of computer networks, including network types, reference models (OSI and TCP/IP), and basic networking terminology.

CO2 (Apply – Bloom's Level: L3): Apply concepts of the Physical and Data Link layers to analyze transmission media, encoding techniques, error detection and correction methods, and multiple access protocols.

CO3 (Analyze – Bloom's Level: L4): Analyze network layer and transport layer mechanisms such as routing algorithms, IP addressing and subnetting, congestion control, and transport protocols (TCP and UDP).

CO4 (Evaluate – Bloom's Level: L5): Evaluate application layer protocols and services, including HTTP, DNS, e-mail, and peer-to-peer applications, and assess challenges in multimedia streaming over IP networks.

Module 1: Introduction	4 Hours

Network, Uses of Networks, Types of Networks, Reference Models: TCP/IP Model, The OSI Model. Understand basic terminology like browser, web server, URL, domain name.

Module 2: Physical Layer	7 Hours
Guided transmission media, Wireless transmission media, Switching techniques, Network topologies, Network devices. Signal Transmission and Encoding, Network Performance and Devices.	
Module 3: Data Link Layer	10 Hours
Design issues, Error Detection & Correction, Elementary Data Link Layer Protocols, Sliding window protocols Multiple Access Protocols - ALOHA, CSMA,CSMA/CD, CSMA/CA, Collision free protocols.	
Module 4: Network Layer	10 Hours
Network Layer Design issues, store and forward packet switching, connection less and connection oriented networks-routing algorithms-optimality principle, shortest path, flooding, Distance Vector Routing, Count to Infinity Problem, Link State Routing, Path Vector Routing, Hierarchical Routing; Congestion control algorithms, IP addresses, CIDR, Sub netting, Supernetting, IPv4, Packet Fragmentation, IPv6 Protocol, Transition from IPv4 to IPv6, ARP, RARP.	
Module 5: Transport Layer	8 Hours
Services provided to the upper layers elements of transport protocol addressing connection establishment, Connection release, Error Control & Flow Control, Crash Recovery. The Internet Transport Protocols: UDP, Introduction to TCP, The TCP Service Model, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Sliding Window, The TCP Congestion Control Algorithm.	
Module 6: Applications layer	6 Hours
Applications layer paradigms: Client server model, HTTP, E-mail, WWW, TELNET, DNS, Peer-to-peer applications. P2P file distribution. Audio and video streaming. Challenges of streaming over best effort IP.	
Total Lecture hours	45 Hours

Text Book

1. Computer Networks - Andrew S Tanenbaum, Pearson Education.
2. Data Communications and Networking - Behrouz A. Forouzan

Reference Books

1. J.F. Kurose and K.F. Ross, Computer networking: a top-down approach.
2. 2. Bhavneet Sidhu, An Integrated Approach to Computer Networks, Khanna Publishing House, 2021.

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

BCS23222P	**COMPUTER NETWORKS LAB**	L	T	P	C
		0	0	2	1

Pre-requisite: Basics of Data Communication

Course Objectives:

The objectives of this course are to:

1. **Provide hands-on experience** with networking devices, cabling techniques, and network topology design using simulation tools.
2. **Enable students to implement and test** data link layer protocols, including error detection and correction methods, sliding window protocols, and multiple access techniques.
3. **Develop practical understanding** of network layer operations such as routing algorithms, IP addressing, subnetting, and address resolution protocols through simulations and packet analysis.
4. **Expose students to transport and application layer protocols** by analyzing TCP/UDP behavior, congestion control mechanisms, and implementing basic client–server communication using socket programming.

Course Outcome:

After successful completion of the course, the students will learn

CO1 (Apply – Bloom's Level: L3): Apply fundamental networking concepts to identify network devices, prepare UTP cables, and design basic network topologies using simulation tools.

CO2 (Apply – Bloom's Level: L3): Implement data link layer protocols and mechanisms such as error detection and correction techniques, sliding window protocols, and multiple access protocols.

CO3 (Analyze – Bloom's Level: L4): Analyze network layer functionalities including routing algorithms, IP addressing, subnetting, and address resolution protocols using simulation and packet analysis tools.

CO4 (Analyze / Evaluate – Bloom's Level: L4–L5): Analyze and evaluate transport and application layer protocols by examining TCP/UDP behavior, congestion control mechanisms, and implementing client–server communication using sockets.

Module 1: Network Fundamentals & Physical Layer

1. **Study of Networking Devices and Cables**
   - Identification and working of Hub, Switch, Router, Modem
   - UTP cable preparation (straight-through & crossover)

Question: Study the working principles of networking devices such as **Hub, Switch, Router, and Modem**. Prepare **straight-through and crossover UTP cables**, test their connectivity, and record the observations.

2. **Network Topologies using Simulation Tools**
   - Star, Bus, Ring, Mesh topology using Packet Tracer / NS2 / NS3

Question: Design and simulate **Star, Bus, Ring, and Mesh topologies** using **Packet Tracer / NS2 / NS3**. Analyze the advantages, limitations, and performance of each topology.

3. **Transmission Media and Signal Encoding**
   - Study and simulation of guided and unguided media
   - Digital encoding techniques (NRZ, Manchester, etc.)

Question: Study guided and unguided transmission media. Simulate **digital signal encoding techniques** such as **NRZ and Manchester encoding**, and compare their characteristics.

Module 2: Data Link Layer

4. **Error Detection and Correction Techniques**
   - Implementation of Parity Check, Checksum, CRC, Hamming Code

Question: Implement **Parity Check, Checksum, CRC, and Hamming Code** techniques for error detection and correction. Analyze their effectiveness in detecting transmission errors.

5. **Sliding Window Protocol Simulation**
   - Stop-and-Wait ARQ
   - Go-Back-N ARQ / Selective Repeat ARQ

Question: Simulate **Stop-and-Wait ARQ**, **Go-Back-N ARQ**, and **Selective Repeat ARQ** protocols. Compare their efficiency in terms of throughput and reliability.

6. **Multiple Access Protocols Simulation**
   - ALOHA, Slotted ALOHA
   - CSMA, CSMA/CD, CSMA/CA

Question: Simulate **Pure ALOHA, Slotted ALOHA, CSMA, CSMA/CD, and CSMA/CA** protocols and analyze their collision handling and channel utilization.

Module 3: Network Layer

7. **Routing Algorithms Implementation**
    - Shortest Path Algorithm (Dijkstra's)
    - Distance Vector Routing Algorithm
    - Link State Routing Algorithm

Question: Implement and analyze **Dijkstra's shortest path algorithm**, **Distance Vector routing**, and **Link State routing** algorithms for packet routing in a network.

8. **IP Addressing and Subnetting**
    - Classful and Classless addressing
    - CIDR and subnet calculations
    - Supernetting problems

Question: Perform **Classful and Classless IP addressing**, **CIDR-based subnetting**, and **supernetting** for a given network scenario and verify the results.

9. **ARP and RARP Protocol Study:** ARP request and reply analysis using packet analyzer (Wireshark)

Question: Capture and analyze **ARP request and reply messages** using **Wireshark**. Study the role of ARP and RARP in IP-to-MAC address resolution.

Module 4: Transport Layer

10. **UDP and TCP Protocol Analysis**

- Study TCP segment and UDP datagram
- Connection establishment and termination (3-way & 4-way handshake)

Question: Analyze **TCP and UDP protocol behavior** by capturing packets using **Wireshark**. Study **TCP 3-way handshake** and **4-way connection termination**.

Module 5: Application Layer

11. **Client–Server Communication using Sockets**

- TCP client-server program
- UDP client-server program

Question: Develop and execute **TCP and UDP client–server programs** using socket programming and analyze data transmission between client and server.

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

Course Code	Cryptography and Data Security	L	T	P	C
BCS23223T		3	0	0	3

Pre-requisite:
Strong foundation in Computer Science, Programming Skills, Networking knowledge and Operating System concepts.

Course Objective:

1. Introduce the fundamental principles of cryptography and secure communication
2. Provide in-depth knowledge of symmetric and asymmetric cryptographic algorithms
3. Explain hashing, digital signatures, and public key infrastructure
4. Develop understanding of data security threats and access control mechanisms
5. Familiarize learners with data protection techniques, privacy, and compliance requirements

Course Outcome:

After completion of this course Students will be able to -

1. Explain fundamental concepts and security services of cryptography
2. Apply symmetric key cryptographic algorithms for secure data protection.
3. Make use of asymmetric key cryptography, hashing, and digital signatures to provide secure, authenticated communication
4. Apply Data protection techniques to safeguard sensitive information from damage, loss, theft, and understand legal & ethical issues in data security

Module-1. Introduction	**8 Hours**

Security Goals, Security services: confidentiality, integrity, authentication, non-repudiation, types of security attacks (passive and active attacks), Basic cryptographic concepts: plaintext, ciphertext, encryption, decryption, keys.

Module-2. Symmetric Key Cryptography	**10 Hours**

Classical and Modern Encryption, Data Encryption Standard, The Strength of DES, Triple DES, Advanced Encryption Standard, Principles of Pseudorandom number generation, Key distribution and key management issues.

Module-3. Public Key Cryptography	**10 Hours**

Public Key Infrastructure - Mathematics of Asymmetric Key Cryptography, Encryption, Principles of Public Key Cryptosystems, Key Exchange, Message Integrity and Message Authentication, Cryptographic Hashing & Digital Signature, Public Key Distribution.

Module-4. Data Security Fundamentals	**10 Hours**
Data security concepts; data states including data at rest, data in transit, and data in use; authentication and authorization mechanisms; Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC); data security threats including data breaches, insider threats, malware, and ransomware; data security architecture.	
Module-5. Data Protection, Privacy, and Compliance	**10 Hours**
Data protection techniques including encryption, data masking, anonymization, and tokenization; Data Loss Prevention (DLP); backup and recovery strategies; secure data storage; data privacy principles; legal and regulatory frameworks including GDPR and provisions of the Indian IT Act; ethical issues in data security.	

Text / Reference Books:

1) Stallings, Cryptography & Network Security - Principles & Practice, Prentice Hall.
2) Debdeep Mukhopadhyay, Behrouz A Forouzan, Cryptography and Network Security, 3rd Edition, McGraw Hill.
3) Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Sunit Belapur and Nina Godbole, Wiley India Pvt. Ltd.
4) Michael E. Whitman & Herbert J. Mattord – *Principles of Information Security*
5) Ravi Sandhu & Latifur Khan – Handbook of Data Security

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

Course Code	Cryptography and Data Security Lab	L	T	P	C
BCS23223P		0	0	2	1

List of Experiments – Cryptography and Data Security Lab [Using C Language]

1. Implementation of Caesar cipher.
2. Perform a Brute force attack on a given Ciphertext. You can assume that the cipher text is encrypted using CAESER Cipher algorithm
3. Construct a Playfair Matrix with a given key (Suppose the Key is - OCCURRENCE). Make a reasonable assumption about how to treat redundant letters in the key.
4. Implementation of transposition ciphers.
5. Implement S-Box substitution of DES. Make Necessary assumptions for S-Box table.
6. Implementation on - Mathematics on public key cryptography (Euler's Totient Function / Fermat's Theorem etc.).
7. Generate public and private key using RSA algorithm. Also implement encryption and Decryption for a given plain text using the keys.
8. Implementation of Diffie–Hellman key exchange.
9. Implementation of cryptographic hash functions (MD5, SHA-1, SHA-2).
10. Implementation of digital signature generation and verification.
11. Implementation of authentication and access control mechanisms.

Teaching Cryptography Lab using the C language enables students to understand cryptographic algorithms at the implementation level, ensures industry relevance, improves performance awareness, and strengthens core computer science and security fundamentals.

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

BCS23224T	**MICROPROCESSOR**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Pre-requisite: Digital Logic Design, Computer Organization Fundamentals, Programming Fundamentals

Course Objectives: After completing this course, students will be able to:

1.Explain the architecture, register organization, and evolution of the 8086 microprocessor and compare RISC and CISC processors.
2.Apply addressing modes, instruction sets, and assembler directives to develop assembly language programs using 8086.
3.Analyze system bus structure, timing, and system design concepts of 8086-based microprocessor systems.
4.Demonstrate the interfacing of memory, I/O, interrupts, and programmable peripheral devices in microprocessor-based systems.

Course Outcome:

CO1: Describe the architecture, register organization, and functional operation of the 8086 microprocessor and differentiate between RISC and CISC architectures.
CO2: Develop and execute assembly language programs using various addressing modes, stacks, procedures, and string manipulation instructions.
CO3: Analyze the system bus structure, timing diagrams, I/O programming, and multiprocessor configurations of the 8086 microprocessors.
CO4: Design and implement memory and peripheral interfacing, interrupt handling, serial communication, and DMA operations using programmable devices.

MODULE 1	**12hours**

Introduction to 8086 microprocessor, Microprocessor architecture, Salient features of advanced microprocessors, RISC & CISC processors. Review and evolution of advanced microprocessors:8086,8088, 80186/286/386/486/Pentium, Register organization of 8086 , Addressing modes, Instruction set and assembler directives.

Module 2	**12hours**

Assembly language programming, Stacks, Procedures, Byte and String Manipulation,8086 System Bus Structure, Basic configurations, System bus timing, System design using 8086, IO programming, Introduction to Multi-programming, System Bus Structure, Multiprocessor configurations.

Module 3	**12hours**
Introduction to the various interfacing chips like 8155, 8255, Interfacing with Key board, LED, Memory Interfacing to 8086, Interrupt Structure of 8086, Vector Interrupt Table, Interrupt Service Routine, Serial Communication Standards, Serial Data Transfer Schemes, 8251 USART Architecture and Interfacing.	
Module 4	**9hours**
General purposes programmable peripheral devices (8253), 8254 programmable interval timer, 8259A programmable interrupt controller & 8257 DMA controller, USART, serial I/O & data Communication.	
Total Lecture hours	**45hours**

Text Book

1. Microprocessor Architecture, Programming, and Applications with the 8085 by R.S. Gaonkar

2. Microprocessor and Interfacing – D.V.Hall, McGraw Hill.

3. The Intel microprocessor - Barry B. Brey, Pearson

Reference Books

1. The 8086 & 8088 Microprocessor- LIU and Gibson, Tata McGraw Hill

2.Yu-Cheng Liu, Glenn A.Gibson, "Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design", Second Edition, Prentice Hall of India, 2007

3. Advance microprocessor and peripheral –A.K. Ray and K. M. Bhurchandi, Tata Mcgraw Hill

4. Micro Computer System 8086/8088 Family Architecture, Programming and Design - Liu and GA Gibson, PHI, 2nd Ed

5. Advanced Microprocessors" – A.K. Gautam

6. "Microprocessor Architecture: Programming and Applications with the 8085/8080A" – R.S. Gaonkar

7. "Microprocessors and Microcontrollers" – Krishna Kant

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

DSE23220T	**IMAGE PROCESSING**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Prerequisite: Basic knowledge of Mathematics/ Statistics

Course Objectives:

The objectives of this course are to enable students to:

1. Explain key concept behind the fundamentals of digital images and visual perception.
2. Comprehend on the analysis of image enhancement, restoration, and compression techniques.
3. To develop skills in image segmentation, feature extraction, and pattern recognition.
4. To apply image processing methods in real-world applications like medical, industrial, and satellite imaging.

Course Outcome:

Learning Outcomes Aligned with Bloom's Taxonomy, this course is designed to measure student outcomes at various cognitive levels. Upon successful completion, students will be able to:

Outcome 1: Explain the fundamentals of digital image representation, visual perception, and basic image processing operations. (Understand level).

Outcome 2: Apply various image enhancement and restoration techniques to improve image quality using mathematical models.

Outcome 3: Analyze different image segmentation and feature extraction methods for object detection and recognition.

Outcome 4: Evaluate image processing algorithms for performance and suitability in real-world applications such as medical, satellite, and surveillance systems by utilizing appropriate image processing tools.

Outcome 5: Design and implement a small application of digital image processing by choosing any real life problems like medicine, forensics, remote sensing , communications etc. with programming skill using Software like Python etc.

MODULE 1:Introduction	**5 Hours**

Light, brightness adaptation and discrimination, pixels, coordinate conventions, Imaging geometry, perspective projection, Sampling and quantization.

MODULE 2:Spatial Domain Filtering	**8 Hours**
Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian	
MODULE 3:Filtering in the Frequency Domain	**7 Hours**
Fourier Transforms and properties, FFT, Convolution, Correlation, 2-D sampling, Discrete Cosine Transform, Frequency domain filtering	
MODULE 4: Image Segmentation	7 hours
Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding, Otsu's method, Moving averages, Multivariable thresholding, Region-based segmentation, Watershed algorithm	
MODULE 5:Image Restoration	**5 Hours**
Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections	
MODULE 6:**Image Compression**	7 hours
Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an informationsource, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, LZW coding, Transform Coding, Sub-image size selection, blocking artifacts, DCT implementation using FFT, Run length coding, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG,Lossless predictive coding, Lossy predictive coding, Motion Compensation.	
MODULE 7: Wavelet based Image Compression	3 hours
Expansion of functions, Multi-resolution analysis, Scaling functions, Wavelet series expansion, Discrete Wavelet Transform (DWT),Continuous Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding, Digital Image Watermarking	
MODULE 8: Morphological Image Processing	3 hours
Basics, SE, Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion	
Case Studies	3 hours
Different case studies on applications of Image Processing	
Total hours	**48 hours**
Text Book	

1.	Digital Image Processing by Rafael C Gonzalez &Richard E Woods, 3rd Edition
2.	Fundamentals of Digital Image Processing by Anil K Jain
Reference Books	
1.	Digital Image Processing by William K Pratt

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

DSE23221T	Speech And Natural Language Processing	L	T	P	C
		3	0	0	3

Prerequisite: Machine Learning

Course Objectives:

Students will be able to–

1. Introduce fundamental concepts of speech processing and NLP.
2. Develop understanding of text and speech analysis techniques.
3. Familiarize students with statistical and machine learning approaches for language processing.
4. Enable students to design basic speech and NLP applications.

Course Outcome:

At the end of successful completion of the course, students will be able to :

CO1: Explain fundamentals of speech signals and language structure.

CO2: Apply text preprocessing and linguistic analysis techniques.

CO3: Implement statistical models for NLP tasks.

CO4: Analyze speech recognition and synthesis systems.

CO5: Design simple NLP-based applications.

Module:1	**8 Hours**

Overview of Speech and Natural Language Processing; Phases in speech and natural language processing, applications, Challenges in Speech and Language Understanding, Levels of Linguistic Analysis.

Module:2	**10 Hours**

Introduction to corpus, elements in balanced corpus, TreeBank, PropBank, WordNet, VerbNet etc. Resource management with XML, Management of linguistic data with the help of GATE, NLTK.

Module:3	**10 Hours**

Text Processing, Tokenization, Normalization, Stemming, Lemmatization, Stop-word

Removal, Part-of-Speech Tagging, Syntax and Parsing, Named Entity Recognition.	
Module:4	**10 Hours**
N-gram Language Models, Hidden Markov Models (HMM), Naïve Bayes Classifier Machine Learning Approaches, Word Embeddings: Word2Vec, GloVe, Deep Learning Techniques.	
Module:5	**8 Hours**
Automatic Speech Recognition (ASR), Text-to-Speech (TTS) Systems, Machine Translation Sentiment, Analysis, Question Answering, Systems Chatbots and Virtual Assistants.	
Total Lecture hours	**46 hours**

Text Book	
1	Daniel Jurafsky and James H. Martin. Speech and Language Processing, 2e, Pearson Education, 2009.
2	Christopher D. Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing. MIT Press, 1999.
Reference Books	
1	James Allen, Natural language Understanding 2e, Pearson Education, 1994.
2	Steven Bird, Ewan Klein and Edward Loper, Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit.
3	Bharati A., Sangal R., ChaitanyaV., Natural language processing: a Paninian perspective, PHI, 2000.
4	Siddiqui T., Tiwary U. S., Natural language processing and Information retrieval, OUP, 2008.
5	Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers
6	Jacob Perkins, Python Text Processing with Nltk 2.0 Cookbook

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

BCM25102T	ACCOUNTANCY	L	T	P	C
		2	0	0	2

Course Objectives:

1. Explain basic accounting concepts and apply relevant accounting standards.
2. Demonstrate the process of recording business transactions and preparing financial statements.
3. Explore and evaluate new and emerging practices in the preparation and presentation of final accounts.

Course Outcomes:

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

CO1: Understand the meaning, objectives, and principles of accounting including the double-entry system of bookkeeping.

CO2: Apply the process of journalizing transactions and posting them to the ledger.

CO3: Identify and prepare various types of subsidiary books, including cash books.

CO4: Analyze the trial balance and prepare financial statements including Trading Account, Profit & Loss Account, and Balance Sheet.

CO5: Evaluate and apply methods for calculating depreciation under SLM and WDVM.

Module 1: Theory Base of Accounting	**06 hours**

Meaning of Accounting, Objectives, Advantages of Accounting, Generally Accepted Accounting Principles , Basic accounting concepts, Accounting Standards, Double Entry system of Book Keeping, Golden rules of Debit and Credit

Module 2: Recording of transaction I	**06 hours**

Journal: Definition, advantages, Procedure of Journalising, Ledger: advantages, rules regarding Posting, Balancing of Ledger account

Module 3: Recording of Transaction II	**06 hours**

Name of Subsidiary Books, Cash Book- definition, advantages, objectives, types of Cash Book, preparation of different types of cash books

Module 4: Trial Balance and Financial Statements	**06 hours**

Trial Balance: Definition and preparation of Trial Balance, Final Account: Preparation of

Trading Account, Profit and Loss Account with adjustments, preparation of Balance Sheet	
Module 5: Depreciation Accounting	**06 hours**
Meaning, Causes, Factors, Methods of calculating depreciation under SLM and WDVM.	
Total Lecture hours	**30 hours**
Suggested Readings:	

Text Books:

1. B.B.Dam, R. a. (2012). *Theory and Practice of Accountancy.* Capital Publishing Company.

Reference Books:

1. Grewal, T. (2018). *Accountancy Theory and Practice.* Sultan Chand & Sons Pvt Ltd.
2. Das, A. (2015). *Theory and Practice of Accountancy.* Navyug Publicatins

GIRIJANANDACHOWDHURYUNIVERSITY
Hatkhowapara, Azara, Guwahati - 781017, Assam

Department of Computer Science and Engineering

Course Structure of B.Tech CSE 5th Semester

Sl. No	Course Code	Course Name	Hours per week			Credit
			L	T	P	C
1.	BCS23213T	Introduction to Database System	3	0	0	3
2.	BCS23213P	Introduction to Database System Lab	0	0	4	2
3.	BCS23214T	AI and Machine Learning	2	1	0	3
4.	BCS23214P	AI and Machine Learning Lab	0	0	4	2
5.	BCS23215T	Design and Analysis of Algorithms	3	0	0	3
6.	BCS23215P	Design and Analysis of Algorithms Lab	0	0	2	1
7.	BCS23216T	Web Technology	2	0	0	2
8.	BCS23216P	Web Technology Lab	0	0	2	1
9.	BCS23217T	Signals and Systems	3	0	0	3
10.	BLW25101T	Constitution of India	2	-	-	0
11.	BCS23218P	Internship (Academia)	-	-	4	2
TOTAL CREDIT			**15**	**1**	**16**	**22**

* 4 weeks Mandatory Academic internship needed to be done in the 4th semester summer break and the report is to be submitted and evaluated in 5th semester.

BCS23213T	Introduction to Database System	L	T	P	C
		3	0	0	3

Prerequisite:

1. Knowledge of Programming Languages

2. Basic mathematical backgrounds (elementary set theory, concepts of relations and functions etc
3. Data Structure and algorithms (Trees, B-tree, B+ tree)

Course Objectives:

1. To understand the relational database design principles, database storage structures, basic issues of transaction processing and concurrency control.
2. To provide in depth knowledge for designing and implementing database queries using Structured Query Language.

Course Outcome:

After Successful completion of the course, students will be able to:

1. Identify the basic need of database management system, different types of databases and query languages (Bloom's L3 Applying)
2. Design entity relationship diagrams, relational DBMS.(Bloom's L6 Creating)
3. Analyze the concepts of functional dependencies, normalization and storage strategies. (Bloom's L4 Analyzing)
4. Apply the principles of database transaction management, database recovery etc. (Bloom's L3 Applying)

Module 1: Introduction	**3 hours**

Data Abstraction, Data Independence, Introduction to Data Models. Entity-relationship model, network model, relational, semi structured model etc

Module 2: Relational Databases	**4 hours**

Relational Data Model, Relational Algebra, Relational Calculus

Module 3: Interacting with database (SQL)	**9 hours**

Data Definition Language (DDL), Data Manipulation Language (DML). Insert/Delete/Update

Simple Queries (select/project/join/ aggregate queries)

Complex queries (With Clause, Nested Subqueries, Views)

Module 4: Big Data	**3 hours**

Key-value Stores and Semi- structured Data, using JSON and MongoDB, or other combinations

Module 5: Database Design	**8 hours**

Introduction to ER model, Mapping from ER to relational model. Functional Dependencies, Normalization.	
Module 6: Physical Design and Storage Strategies	**6 hours**
Overview of Physical Storage (Hard Disks, Flash/SSD/RAM), sequential vs random I/O, Reliability via RAID Storage Organization (Records, Pages and Files), Indexing, B+-Trees	
Module 7: Query Processing and Optimization	**6 hours**
Query Processing: External sort, Joins using nested loops, indexed nested loops Overview of Query Optimization: equivalent expressions, and concept of cost- based optimization	
Module 8: Transaction Processing	**6 hours**
Concept of transactions and schedules, ACID properties. Conflict-serializability. Concurrency control: locks, 2PL, Strict 2PL. Recovery using undo and redo logs.	
Total Hours	**45 hours**

Text Book

1.	Database System Concepts, 7th Edition, Silberschatz, Korth and Sudarshan, McGraw-Hill. Indian Edition released 2021
2.	Fundamentals of Database Systems, 7th Edition, Elmasri and Navathe, Pearson Pubs, 2017
3.	Principles of Database Management, Lemahieu, Broucke and Baesens, Cambridge University Press, 2018
4.	Database Management Systems, RP Mahapatra, Khanna Publishing House, 2020.
5.	Database Management Systems, Krishnan, McGraw Hill.
6.	"Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
7.	"Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

Suggested reference books / Online resources:

1.	Software a. Relax Relational algebra calculator: https://dbis-uibk.github.io/relax/landing b. SQL: PostgreSQL/MySQL/MariaDB, or SQLite in browser c. B+-tree visualization:

	https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html d. MongoDB e. Various DB systems playground: https://www.pdbmbook.com/playground

BCS23213P	INTRODUCTION TO DATABASE SYSTEM LAB	L	T	P	C
		0	0	4	2

Prerequisite: Basic programming skills

Course Objectives:

1. To provide students with hands-on experience to design and create databases, write SQL queries, and perform data manipulation tasks, enabling them to translate theoretical knowledge into practical skills.

2. To gain proficiency in setting up and configuring database management systems, managing user access, and implementing security measures

3. To enhance students' abilities to optimize database queries and improve system performance

Course Outcome:

After Successful completion of the course, students will be able to

1 : Make use of Data Definition Language(DDL) statement using SQL for Creating, Deleting and Modifying Relations Schemas.

2 :Make use of Data Manipulation Language(DML) statement using to SQL Retrieve, Store,Modify,Delete,Insert and Update data in Database.

3 : Apply the concept of trigger using SQL.

4 : Apply the concept of procedure using SQL.

Experiments

1. Write statements for creation/alteration/view of relational database schema along with necessary integrity constraints. Insert tuples into the created tables.
2. Write SQL statements for selection, updation and deletion of data
3. Write SQL statements for joins, grouping, nested sub query.
4. Write statements for creating and using stored procedures.
5. Write statements for creating and using triggers.

Total Lab hours	**30 hours**

Reference Books

1.	"Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
2.	"Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
3.	"Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
4.	"Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

	Introduction to AI\ML	L	T	P	C
		2	1	0	3

Prerequisite:

1. A course on Python (Anaconda)

Course Objectives:

1. COB1: To introduce basic concepts and techniques of Artificial Intelligence and Machine Learning.
2. COB2: To develop understanding of problem-solving, knowledge representation, and learning algorithms.
3. COB3: To implement and evaluate machine learning models using Python and relevant libraries.

Course Outcome:

After Successful completion of the course, students will be able to:

1. CO1: Understand fundamental concepts of Artificial Intelligence and its applications.
2. CO2: Apply search and reasoning techniques to solve real-world problems.
3. CO3: Demonstrate knowledge of machine learning algorithms and evaluate their performance.
4. CO4: Implement ML algorithms using Python and interpret the results.
5. CO5: Design a simple AI/ML project that addresses a real-world use case.

Module 1: Introduction of AI & ML	**8 hours**

Definition of AI & ML, Applications of AI & ML, Types of Learning (Supervised, Unsupervised, Reinforcement), Hypothesis Space and Inductive Bias, Evaluation Metrics: Accuracy, Loss Functions, Cross-Validation Techniques, Overfitting and Underfitting

Module 2: Knowledge Representation: Introduction	**6 hours**

Knowledge Representation: Introduction, Propositional Logic, Predicate Logic, Rule-Based Systems, Inference Engine and Reasoning Mechanisms.

Module 3: AI vs ML vs Data Science	**8 hours**

Data Preprocessing: Cleaning, Encoding, Feature Handling Missing Data, Model Evaluation Metrics: Precision, Recall, F1, ROC, Introduction to pandas and scikit-learn.

Module 4: Linear Regression – Concept and Implementation	**8 hours**

Logistic Regression, Naïve Bayes, Decision Trees – ID3, Entropy, Gini, Decision Trees Day, K-Nearest Neighbors (KNN), Support Vector Machines (SVM) , K-Means Clustering , Hierarchical Clustering Confusion Matrix, Precision, Recall (Summary & Practice)

Module 5: Introduction to Neural Networks	**8 hours**

Neural Networks, Perceptron Algorithm, Activation Functions, Multilayer Networks Forward Propagation, Gradient Descent, Backpropagation Algorithm

Module 6: Model Deployment and Training	**7 hours**

Overview: Chatbots, Fraud Detection, Use Case: Sentiment Analysis, Model Training, Model Deployment.	
Total Hours	**45 hours**
Text Book	
1.	Russell, S., & Norvig, P. (2020). Artificial intelligence: A modern approach (4th ed.). Pearson.
2.	Muller, C.A & Guido, S. (2017). Introduction to Machine Learning with Python. (3rd ed.). Sebastopol, CA: O'Reilly.
3.	Raschka, S. & Mirjalili, V. (2017). Python Machine Learning (2 nd ed.). : Packt Publishing
4.	Aurelien, G. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, (3 rd ed.). : O'Reilly.
5.	Mitchell, Tom (1997), Machine Learning. (1 st Ed.), McGraw- Hill.

	Introduction to AI\ML (Practical)	L	T	P	C
		0	0	4	4

Prerequisite:

1. A course on Python (Anaconda)

Course Objectives:

1. COB1: To introduce basic concepts and techniques of Artificial Intelligence and Machine Learning.
2. COB2: To develop understanding of problem-solving, knowledge representation, and learning algorithms.
3. COB3: To implement and evaluate machine learning models using Python and relevant libraries.

Course Outcome:

After Successful completion of the course, students will be able to:

1. CO1: Understand fundamental concepts of Artificial Intelligence and its applications.
2. CO2: Apply search and reasoning techniques to solve real-world problems.
3. CO3: Demonstrate knowledge of machine learning algorithms and evaluate their performan ce.
4. CO4: Implement ML algorithms using Python and interpret the results.
5. CO5: Design a simple AI/ML project that addresses a real-world use case.

Module 1: Explore basic AI/ML	**8 hours**
Explore basic AI/ML using Pythons, Identify real-world AI/ML applications, Implement and visualize different types of learning in scikitlearn, Demonstrate overfitting/underfitting with polynomial degrees.	
Module 2: Implement propositional logic	**6 hours**
Propositional logic in Python using Boolean operators, Basic rule-based systems using if-else conditions	
Module 3: AI vs ML vs Data Science	**8 hours**
Compare AI, ML, Data Science model, Perform data cleaning and encoding using pandas, Apply normalization and missing value imputation , Evaluate models using precision, recall, F1-score	
Module 4: Concept of Implementation	**8 hours**
Linear Regression on a real-world dataset , Build Logistic Regression model, Naïve Bayes classifier, Decision Tree algorithm, KNN, SVM and compare model performance	
Module 5: Analysis of datasets	**6 hours**
Preprocess dataset, Train sentiment analysis model, Deploy model using Flask or Streamlit	
Total Hours	**36 hours**

Text Book

1. Russell, S., & Norvig, P. (2020). Artificial intelligence: A modern approach (4th ed.). Pearson.

2.	Muller, C.A & Guido, S. (2017). Introduction to Machine Learning with Python. (3rd ed.). Sebastopol, CA: O'Reilly.
3.	Raschka, S. & Mirjalili, V. (2017). Python Machine Learning (2 nd ed.). : Packt Publishing
4.	Aurelien, G. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, (3 rd ed.). : O'Reilly.
5.	Mitchell, Tom (1997), Machine Learning. (1 st Ed.), McGraw- Hill.

GIRIJANANDACHOWDHURYUNIVERSITY
Hathkhowapara, Azara, Guwahati - 781017,Assam

BCS23215T	Design and Analysis of Algorithms	L	T	P	C
		3	0	0	3

Pre-requisite: A solid foundation in programming fundamentals and data structures and algorithms.

Course Objectives:

1. To understand the fundamental characteristics and performance analysis of algorithms, including time and space complexity using asymptotic notations and recurrence relations.
2. To study and apply various algorithmic design strategies such as Divide and Conquer, Greedy, Dynamic Programming, Backtracking, and Branch & Bound to solve real-world problems efficiently.
3. To explore classical graph and tree algorithms and apply them in problem-solving using techniques like BFS, DFS, shortest path algorithms, minimum spanning tree, and network flow.
4. To analyze the theoretical foundations of computational complexity, differentiate between tractable and intractable problems, and understand NP-completeness and related concepts.
5. To examine advanced algorithmic approaches including approximation, randomized, and evolutionary algorithms, and develop insights into problems beyond NP such as PSPACE.

Course Outcomes: After successful completion of the course, the students will learn:

CO1: Understand the fundamental concepts and mathematical analysis of algorithms, including time and space complexity, and recurrence relations. → *Bloom's Level*: Understand, Analyze

CO2: Apply various algorithm design techniques such as Divide and Conquer, Greedy, Dynamic Programming, Backtracking, and Branch & Bound to solve computational problems. → *Bloom's Level*: Apply, Analyze

CO3: Develop efficient algorithms for graph and tree-based problems, and evaluate their performance using appropriate algorithmic strategies. → *Bloom's Level*: Apply, Evaluate

CO4: Analyze computational complexity classes (P, NP, NP-Complete, NP-Hard), and explore advanced approaches like approximation, randomized, and evolutionary algorithms for solving hard problems. → *Bloom's Level*: Analyze, Create

Module 1: Introduction Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds –best, average and worst case behavior; Time and space complexity, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.	12 Hours
Module 2: Fundamental Algorithmic Strategies Divide and Conquer, Brute Force, Greedy, Dynamic Programming, Branch and Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem Solving, Bin Packing, Knap Sack, TSP, Heuristics characteristics and their application domains.	12 Hours

Module 3: Graph and Tree Algorithms Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.	11 Hours
Module 4: Tractable and Intractable Problems Computability of Algorithms, Computability classes –P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP complete problems and Reduction techniques	5 Hours
Module 5: Advanced Topics Approximation algorithms, Randomized algorithms, Evolutionary algorithms, Class of problems**,** beyond NP –P SPACE	5 Hours
Total Lecture hours	45 Hours

Text Books:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.

Reference Books:

1. Ellis Horowitz, Satraj Sahni, Rajasekharam, Fundamentals of Computer Algorithms, University Press, New Delhi.
2. R. C. T. Lee, S. S. Tseng, R.C. Chang and T. Tsai, Introduction to Design and Analysis of Algorithms A strategic approach, McGraw Hill, India.

Internal Member/BoS

External Member/BoS

GIRIJANANDACHOWDHURYUNIVERSITY
Hathkhowapara, Azara, Guwahati - 781017,Assam

BCS23215P	Design and Analysis of Algorithms Lab	L	T	P	C
		0	0	2	1

Pre-requisite: Prerequisite: Good understanding of data structures and basic programming in C/C++/Java/Python.

Course Objectives:

To implement and analyze various algorithms using different design strategies such as Divide and Conquer, Greedy, Dynamic Programming, Backtracking, Branch and Bound, and Graph algorithms. The students will also gain hands-on experience with problems related to NP-Completeness and advanced algorithmic approaches.

Course Outcomes: After successful completion of the course, the students will learn:

CO1: Implement and analyze algorithms for solving problems using recursion and evaluate their time complexity using asymptotic notations. (Apply, Analyze)

CO2: Design and apply algorithmic strategies such as Divide & Conquer, Greedy, Dynamic Programming, Backtracking, and Branch & Bound to solve real-world problems. (Apply, Create)

CO3: Develop and simulate graph algorithms like shortest path, MST, and traversal methods using appropriate data structures and evaluate their performance. (Apply, Evaluate)

CO4: Demonstrate the behavior of NP-complete problems and explore the use of approximation, randomized, or evolutionary algorithms for problem-solving. (Understand, Analyze, Create)

Module 1: Introduction to Algorithm Analysis — 8 Hours

Experiment 1: Write programs to compute time complexity of different operations (e.g., sorting, searching).

Experiment 2: Implement recurrence relations using:

- Substitution method
- Recursion Tree method
- Master's Theorem

Module 2: Algorithm Design Strategies — 8 Hours

Experiment 3: Implement Merge Sort and Quick Sort using Divide and Conquer.

Experiment 4: Solve the Knapsack Problem.

Experiment 5: Solve the Travelling Salesman Problem.

Experiment 6: Implement a solution for the N-Queens problem using

Backtracking. **Experiment 7**: Implement Branch and Bound for: o 0/1 Knapsack o TSP	
Module 3: Graph Algorithms **Experiment 8**: Implement DFS and BFS for a graph. **Experiment 9**: Implement Dijkstra's and Bellman-Ford shortest path algorithms. **Experiment 10**: Implement Kruskal's and Prim's algorithms for Minimum Spanning Tree. **Experiment 11**: Implement Topological Sorting. **Experiment 12**: Implement Network Flow using Ford-Fulkerson algorithm.	8 Hours
Module 4: NP-Completeness and Advanced Topics **Experiment 13**: Demonstrate NP-complete behavior by solving Subset Sum problem using: • Backtracking • Dynamic Programming (Analyze time complexity for both) **Experiment 14**: Implement a basic Approximation Algorithm for Vertex Cover or TSP. **Experiment 15**: Implement a simple Randomized algorithm (e.g., Randomized Quick Sort). **Experiment 16**: (Optional) Implement a Genetic Algorithm for optimization problem.	8 Hours
Total Lecture hours	32 Hours

Syllabus	Web Technology BCS23216T	L	T	P	C
		3	0	0	3

Pre-requisite: None

Course Objectives:

- Design and develop static and responsive web pages using HTML, CSS, and JavaScript
- Build backend services using Node.js and Express
- Create modern frontend applications using React.js
- Integrate backend, frontend, and database into a working full-stack application
- Develop mini-projects such as Student and Health Information Systems

Course Outcome:

After successful completion of the course, the students will be able to

CO1 Design and develop static and responsive web pages using HTML, CSS, and JavaScript for enhanced user experience and accessibility.
CO2 Construct and manage backend services using Node.js and Express to handle server-side logic and RESTful API operations.
CO3 Develop modern, component-based frontend applications using React.js with effective state management and routing.
CO4 Integrate frontend, backend, and database layers to build and deploy full-stack web applications with seamless data communication.

Module1:Introduction — **10 hours**

Introduction to HTML: Elements, Lists, Tables, Images, Forms, Frames
Cascading Style Sheets (CSS)
Introduction to DHTML
XML Basics: DTD (Document Type Definition), XML Schemas
DOM (Document Object Model)

Module2:JavaScript — **12 hours**

JavaScript Basics: Variables, Data types, Control Structures
Functions and Arrays
Objects and Events
Dynamic HTML using JavaScript

Module3:Web Servers — **10 hours**

Web Servers: Apache (XAMPP/LAMPP), Node.js Environment
Node.js Basics:
Modules, File System, HTTP, Express.js
Handling Forms, Routing, Middleware
Session Management using Express
Cookies and Authentication
Introduction to REST API with Node.js

Module4: Frontend & Database Integration — **7 hours**

React.js: Components, Props, State, Events Forms and Hooks Axios for API Calls Database Connectivity: MySQL Integration with Node.js CRUD operations Case Study Implementations (mandatory): Student Information System Health Management System

Books

- Paul Deitel, Harvey Deitel, Abbey Deitel – Internet & World Wide Web: How to Program, 5th Edition, Pearson, 2012.
- Ethan Brown – Web Development with Node and Express, O'Reilly, Latest Edition
- Robin Wieruch – The Road to React, Leanpub

Syllabus	Web Technology LAB BCS23216P	L	T	P	C
		0	0	2	1

Pre-requisite: Basic knowledge of Programming

CourseObjectives:

- Design and develop static and responsive web pages using HTML, CSS, and JavaScript
- Build backend services using Node.js and Express
- Create modern frontend applications using React.js
- Integrate backend, frontend, and database into a working full-stack application
- Develop mini-projects such as Student and Health Information Systems

CourseOutcome:

CO1 Design and develop static and responsive web pages using HTML, CSS, and JavaScript for enhanced user experience and accessibility.
CO2 Construct and manage backend services using Node.js and Express to handle server-side logic and RESTful API operations.
CO3 Develop modern, component-based frontend applications using React.js with effective state management and routing.
CO4 Integrate frontend, backend, and database layers to build and deploy full-stack web applications with seamless data communication.
CO5 Design and implement mini-projects (e.g., Student Information System, Health Information System) demonstrating full-stack development practices.

Suggested Practical Lists:

1. Create a basic static web page using HTML with proper structure and semantic elements.
2 Design a responsive web layout using CSS Grid and Flexbox.
3 Implement a form with validation using HTML5 and JavaScript (e.g., registration form).
4 Develop an image gallery or portfolio using CSS transitions and animations.
5 Create a JavaScript program to manipulate the DOM dynamically (add, edit, remove elements).
6 Build a To-Do List application using HTML, CSS, and JavaScript with local storage.
7 Develop a RESTful API using Node.js and Express to perform CRUD operations on student data.
8 Connect the backend API with a MySQL database for data persistence.
9 Create a React.js frontend to consume RESTful APIs and display data dynamically.
10 Implement user authentication (login/register) using Express sessions or JWT.
11 Integrate React frontend with Node.js backend and MySQL database to form a full-stack mini-project.
12 Develop and present a mini project (e.g., Student Management / Health Record System) demonstrating full-stack integration.

BCS23217T	SIGNALS AND SYSTEMS	L	T	P	C
		3	0	0	3

Pre-requisite: Basics of Mathematics

Course Objectives:

1. To introduce the fundamental concepts of signals and systems and their classifications in both continuous and discrete time domains.
2. To develop analytical skills required to analyze Linear Time-Invariant (LTI) systems using convolution, differential and difference equations.
3. To provide a comprehensive understanding of signal transformation techniques including Fourier, Laplace, and Z-transforms, along with the implications of signal sampling.

Course Outcome:

After successful completion of the course, the students will learn

1. CO1 : Explain the characteristics of different types of signals and systems including linearity, causality, and stability.
2. CO2 : Apply convolution and system equations to determine the output of continuous and discrete time LTI systems.
3. CO3 : Analyze signals using Fourier, Laplace, and Z-transform techniques for both time and frequency domain representations.
4. CO4 : Evaluate the effects of sampling, aliasing, and reconstruction in signal processing systems.

MODULE 1 — **15hours**

Signals and systems as seen in everyday life and in various branches of engineering and science, Signal properties: periodicity, determinism and stochastic character. Standard signals- Step, Ramp, Pulse, Impulse, Real and complex exponentials and Sinusoids_ Classification of signals — Continuous time (CT) and Discrete Time (DT) signals, Periodic & Aperiodic signals, Deterministic & Random signals, Energy & Power signals — Classification of systems- CT systems and DT systems- — Linear & Nonlinear, Time-variant & Time-invariant, Causal & Non-causal, Stable & Unstable.

Module 2 — **10hours**

Impulse response and step response, convolution, cascade interconnections, Characterization of causality and stability of LTI systems, System representation through differential equations and difference equations. Periodic inputs to an LTI system, the notion of a frequency response and its relation to the impulse response.

Module 3 — **15hours**

Fourier series for periodic signals, Calculation of Fourier Coefficients, Fourier Transform — properties- Laplace Transforms and properties Impulse response — Difference equations-Convolution sum- The Discrete-Time Fourier Transform (DTFT) and the Discrete Fourier Transform (DFT) and Z Transform Analysis of Recursive & Non-Recursive systems-DT systems connected in series and parallel.

Module 4	**5 hours**
The Sampling Theorem and its implications, Spectra of sampled signals, Aliasing and its effects, Relation between continuous and discrete time systems.	
Total Lecture hours	**45 hours**

Text Book

1. A. V. Oppenheim, A. S. Willsky and S. H. Nawab, “Signals and systems”, Prentice Hall India,1997.
2. A Nagoor Kani , “SIGNALS AND SYSTEMS”,Tata McGraw-Hill
3. B.P. Lathi "Principles of signal processing and linear systems" , Publication- New Delhi Oxford University Press c2009

Reference Books

1. J. G. Proakis and D. G. Manolakis, “Digital Signal Processing: Principles, Algorithms, and Applications”, Pearson, 2006.

2. H. P. Hsu, “Signals and systems”, Schaum’s series, McGraw Hill Education, 2010.

3.S. Haykin and B. V. Veen, “Signals and Systems”, John Wiley and Sons, 2007.

4.A. V. Oppenheim and R. W. Schafer, “Discrete-Time Signal Processing”, Prentice Hall, 2009.

5.M. J. Robert “Fundamentals of Signals and Systems”, McGraw Hill Education, 2007.7.B. P. Lathi, “Linear Systems and Signals”, Oxford University Press, 2009.

6. Ramesh Babu & Anandh Natarajan, "Signals and Systems", Publisher: Scitech Publications.

7. Simon Haykin and Barry Van Veen, "Signals and Systems", Publisher: Wiley India

Internal Member/BoS External Member/BoS

Course Code	Course Title	Hours per week L-T-P	Credit C
	Constitution of India	2-0-0	0 (PP/NP)

Course Objectives: Students will be able to:

1. Understand the importance of Constitution in a country also understand the nature of the Constitution of India.
2. Understand the provisions of equality, liberty and freedom under the Constitution of India.
3. To address the importance of DPSP and its relation with Fundamental Rights.
4. To address the provisions of emergency and amendment.

Course Outcomes: Students will be able to:

1. Discuss the characteristics and nature of the Constitution of India .
2. Discuss about the relation among FR, DPSP and Fundamental Duties.
3. Explain the power, function of the different organs of the Constitution.
4. Discuss various provisions of Emergency and its effect.

Module 1. Introduction :

a) Definition and Classification of Constitution
b) Salient features of the Constitution of India
c) Nature of Constitution of India- Whether federal or unitary
d) Preamble of the Constitution of India
e) Article 1
f) Citizenship – Acquirement and Termination (Article 5---11)

Module 2: Fundamental Rights:

a) What is Fundamental Rights?
b) Definition of Law, State (Article 12, 13)

c) Part III of the Constitution of India (Article 14, 15, 16, 19, 20, 21, 21-A, 23, 24, 29 30, 32)

Module 3: Directive Principles of State Policy and Fundamental Duties

a) Nature and Significance of DPSP

b) Differences between FR and DPSP

c) Part IV – (Articles 38, 29, 39A, 41, 42, 44, 45, 47, 48A)

d) Fundamental Duties

Module 4: **Organs of Government under the Constitution of India**

a) Parliament- Composition

b) Executive- Centre and State

c) Judiciary- Supreme Court and High Court

Module 5: Emergency provisions and Amendment of the Constitution of India

a) Different types of emergency

b) Effect of emergency

c) Amendment- meaning, importance, procedures and limitation

Text books/ references

1. **The Constitution of India, 1950 (Bare Act)**
2. **Constitution of India, J. N. Pandey**
3. **Constitutional Law of India, Narender Kumar**
4. **Constitution of India, S. N. Myneni**

GIRIJANANDA CHOWDHURY UNIVERSITY,ASSAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE STRUCTURE OF B.TECH 4TH SEMESTER

SL. NO.	COURSE CODE	COURSE NAME	L	T	P	C	CA	ESE
1	BCS23206T	DATA COMMUNICATION	3	0	0	3	40	60
2	BCS23207T	COMPUTER ORGANIZATION AND ARCHITECTURE	3	1	0	4	40	60
3	BCS23208T	OPERATING SYSTEM	3	0	0	3	40	60
4	BCS23209T	JAVA PROGRAMMING	3	0	0	3	40	60
5	BCS23209P	JAVA PROGRAMMING LAB	0	0	2	1	20	30
6	BCS23210T	GRAPH THEORY	3	0	0	3	40	60
7	BCS23211T	ADVANCED PYTHON PROGRAMMING	3	0	0	3	40	60
8	BCS23211P	ADVANCED PYTHON PROGRAMMING LAB	0	0	2	1	20	30
9	BEL23212T	IKS IV: Indian Science, Engineering & Technology (Past , Present & Future)	3	0	0	3	40	60
TOTAL CREDIT						24	320	480

	DATA COMMUNICATION (BCS23206T)	L	T	P	C
		3	0	0	3

Pre-requisite:

Basic mathematics, Basic electronics

Course Objectives:

1. To introduce basic concepts of data communication and networking.
2. To explore data transmission techniques including analog and digital communication, and different transmission media, guided and unguided.
3. To study different encoding, modulation and multiplexing techniques, along with error detection/correction techniques.
4. To learn standard LAN protocols and switching techniques.

Course Outcome:

After successful completion of the course, the students will be able to

CO1: **Understand** the data communication principles and networking basics.

CO2: **Evaluate** different analog and digital transmission techniques, including modulation, encoding, and multiplexing, and error detection/correction techniques.

CO3: **Analyze** transmission mediums used for communication, and different medium access control mechanisms.

CO4: **Understand** LAN protocols and switching techniques.

MODULE 1: Introduction — **5 hours**

ISO - OSI reference model; Protocol; Interface and service concepts; Layer wise functionality.

MODULE 2: Physical layer — **8 hours**

Concepts of data transmission; Signals: Analog, Digital: Signal properties; Periodic and Aperiodic Signals; Composite Signals; Time-domain and frequency-domain representations of signal; Channel capacity; Transmission impairments

MODULE 3: Encoding, Modulation, Multiplexing — **12 hours**

Line coding: NRZ, RZ, Manchester, Differential Manchester, Bipolar; Digitization: PCM, Delta modulation; Analog modulation: AM, FM, PM, Digital Modulation: ASK, FSK, PSK, QAM; Multiplexing: FDM, TDM, WDM, CDM, OFDM, Spread spectrum techniques.

MODULE 4: Transmission media — **7 hours**

Guided: Twisted pair, Coaxial cables, Optic fibers; Unguided: EM waves: Radio, Microwaves, Infrared; Satellite communication; Propagation models; SONET, Frame relay, ATM etc.

MODULE 5: Data link layer — **8 hours**

Types of Errors, Error Detection, Parity Check, Vertical Redundancy Check Longitudinal Redundancy Check, Cyclic Redundancy Check, Checksum, Error Correction; MAC protocols: ALOHA, CSMA, CSMA/CD, CSMA/CA, Token ring.

MODULE 6: Standard LAN protocols	**5 hours**
Network devices: Repeaters, Hub, Bridges, Switches, Routers, Gateways; Switching techniques: Circuit, Packet, Message; FDDI; VLAN; WLAN; PAN; WiMax.	
Total Lecture hours	**45 hours**

Text Book

(1) Forouzan B. A, Data Communication and Networking, 5e, Tata McGrawHill

(2) Stalling, Data and Computer Communication, 8e, PHI (EEE)

(3) Horak, R, Communication Systems and Networks, Wiley

Reference Books

(1) Tanenbaum A.S., Computer Network, 5e, PHI (EEE)

(2) Leon-Garcia, Widijaja, I., Communication, 5e, PHI (EEE)

(3) Kurose J. F, and Ross, K W, Computer Networking: A Top-Down Approach

	COMPUTER ORGANISATION AND ARCHITECTURE	L	T	P	C
		3	1	0	4

Pre-requisite: Basic Digital Logic Design, Fundamentals of Computer Systems

Course Objectives:

1. To understand the structure and functional organization of computers, including instruction sets, CPU, memory, and I/O subsystems.
2. To explore arithmetic operations and CPU design principles, such as control units, micro-operations, and microprogramming.
3. To analyze memory hierarchy, cache performance, virtual memory concepts, and data transfer mechanisms in I/O subsystems.

Course Outcome:

After successful completion of the course, the students will learn

1. CO1: To understand the basic functionality and operation of computer fundamentals including architecture and arithmetic functions.
2. CO2: To analyse CPU design principles, including control units, micro-operations, and microprogramming.
3. CO3: To analyse the memory organization, types of memory and memory management and interfacing between the CPU and peripherals.
4. CO4: To apply the concepts of parallel processing, pipelining and interprocessor communication.

MODULE 1 — **12hours**

Structure of Computer: Introduction to Computer Architecture, Functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Introduction to Instruction Set Types of ISA; RISC, CISC. Instruction execution cycle, RTL interpretation of instructions.
Computer Arithmetic : signed number representation, fixed and floating point representations, character representation. Computer arithmetic –integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication –shift-and add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.

Module 2 — **9 hours**

Introduction to x86 architecture and instruction set, Registers, addressing modes, instruction set. CPU control unit design: hardwired and micro-programmed design approaches, Micro-Program example, MICRO-OPERATIONS: Arithmetic Micro-Operations, Logic Micro-Operations.

Module 3 — **8hours**

Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards, Instruction Pipeline, Parallel Processing

Module 4 — **12hours**

MEMORY SYSTEM: Memory Hierarchy, Semiconductor Memories, RAM (Random Access Memory), Read Only Memory (ROM), Types of ROM, Cache Memory, cache size vs. block size, Performance considerations, Virtual memory, Paging, mapping functions, replacement algorithms.

Module 5	**4 hours**
Peripheral devices and their characteristics: Input-output subsystems, I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA. privileged and non-privileged instructions, software interrupts and exceptions.	
Total Lecture hours	**45 hours**

Text Book

1."Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.

2."Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraw Hill Higher Education.

3."Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill

Reference Books

1."Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.

2."Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

3. Computer System Architecture: By M. Morris Mano.

4. Structured Computer Organization: By Tanenbaum.

5. Computer Organization: By Stallings.

6. Computer Architecture and Organization: By Hayes.

7. Microprocessor Architecture, Programming, and Applications with the 8085 Ramesh S. Gaonkar Pub: Penram International.

Internal Member/BoS External Member/BoS

BCS23208T	Operating System	L	T	P	C
		3	0	0	3

Prerequisite:

1. Knowledge of data structures, algorithms, and computer architecture.
2. Proficiency in programming languages like C or C++.

Course Objectives:

1. To make aware of different types of Operating System and their services.
2. To provide in depth knowledge on different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
3. To familiarize students deadlocks prevention and detection techniques
4. To provide various mechanisms involved in memory management techniques and managing virtual memory.
5. To familiarize students with the disk management functions and techniques

Course Outcome:

After Successful completion of the course, students will be able to

CO1: Apply concepts of operating system, process.
CO2: Analyse the concepts of process synchronization, inter process communication and Deadlock
CO3: Apply Memory management and virtual memory concept.
CO4: Analyse I/O Management, File Systems Management, Disk Management, Protection and Security Mechanism.

Module: 1	**8 hours**

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls

Module: 2	**9 hours**

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads. Process creation system call.
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive

Module: 3	**7 hours**

Inter-process Communication: Communication models, Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer-Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

Module: 4	**8 hours**

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention,

Detection and Avoidance, Banker's Algorithm, Recovery from Deadlock.	
Module: 5	**8 hours**
Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation –Hardware support for paging, Protection and sharing, Disadvantages of paging. Virtual Memory: Basics of Virtual Memory – Hardware and control structures –Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, first in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU)	
Module: 6	**5 hours**
I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software File Management: File Concept, Access Methods, Directory Structure, File System Structure, Allocation Methods, and Free-Space Management. Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, RAID Structure	
Total hours	**45 hours**

Text Book

1.	Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2.	Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.
3.	Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
4.	Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
5.	Design of the UNIX Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
6.	Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Reference Books

1.	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2006), Operating System Principles, 7th edition, Wiley India Private Limited, New Delhi.
2.	Deitel & Deitel (2008), Operating systems, 3rd edition, Pearson Education, India.
3.	Modern Operating Systems, Tanenbaum, Fourth Edition. Pearson Publication.
4.	Operating Systems, Internals and Design Principles, Stallings, Seventh Edition, Pearson Publication.

GIRIJANANDA CHOWDHURY UNIVERSITY
Hathkhowapara, Azara, Guwahati - 781017, Assam

BCS23209T	Java Programming	L	T	P	C
		3	0	0	3

Pre-requisite: Concept of OOP using C++

Course Objectives:

1. Apply the concepts of Java to solve problems.
2. Understand the principles of inheritance and polymorphism.
3. Implementation of packages and interfaces.
4. Understand the concepts of exception handling and multithreading.
5. Implementation of the design of Graphical User Interface using applets and swing controls.
6. Understand the concepts of JDBC, JSP.

Course Outcomes: After successful completion of the course, the students will learn:

CO1: **Understand** the foundational concepts of Java, including data types, variables, control structures, object-oriented principles, and the execution flow of programs. **(Cognitive Level: Understanding)**

CO2: **Apply** object-oriented programming concepts such as inheritance, polymorphism, interfaces, and exception handling to develop modular and robust Java applications. **(Cognitive Level: Applying)**

CO3: **Design** interactive GUI-based applications using AWT and Swing components, incorporating event handling and layout management. **(Cognitive Level: Creating)**

CO4: **Utilize** JDBC and JSP to connect Java applications with databases and create dynamic, web-based solutions. **(Cognitive Level: Applying)**

Module 1:	**10 hours**
History of java, features of java, data types, variables, static variables, constants, scope and lifetime of variables, operators, structure of java program with example, execution flow of java program, new keyword , type conversion and casting, conditional statements, loops, break and continue statements, arrays, class, object and its methods, constructors and its types, static fields and methods, access control, this reference, final keyword, overloading methods and constructors, recursion, garbage collection, JVM, JRE.	
Module 2:	**10 hours**
Inheritance: Inheritance types, super keyword, final classes and methods. Polymorphism: Method overloading and overriding, abstract classes and methods. Interfaces: Interfaces vs. abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface. Packages: Defining, creating and accessing a package, importing packages.	

Module 3: Exception handling: Define exception, advantages of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, creating own exception sub classes. Multithreading: Define Thread, multithreading, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication, and producer consumer problem.	**8 hours**
Module 4: Applets – Define applets, differences between applets and applications, Life cycle of an applet, Passing parameters to applets. GUI Programming with Java- The AWT class hierarchy, Introduction to Swing, Swing vs. AWT, Hierarchy for Swing components, Overview of some Swing components – Jbutton, JLabel, JTextField, JTextArea, simple Swing applications, Layout management – Layout manager types – border, grid and flow Event Handling- Events, Event sources, Event classes, Event Listeners, Event sources and Listeners, Delegation event model, Examples: Handling a button click, Handling Mouse events, Adapter classes.	**10 hours**
Module 5: Components and architecture of JDBC, Types of JDBC Architecture, Working of JDBC, JDBC Drivers, Establishing JDBC Connection in Java, Types of Statements in JDBC. JSP architecture, Life cycle of JSP, Difference between Servlet and JSP.	**10 hours**
Total Lecture hours	**48 hours**

Text Books/ Reference Books:

1. Java Fundamentals – A Comprehensive Introduction, Herbert Schildt and Dale Skrien, TMH.

2. Core Java: An Integrated Approach – Dr R Nageswara Rao.

3. Java for Programmers, P.J.Deitel and H.M.Deitel, PEA (or) Java: How to Program, P.J.Deitel and H.M.Deitel, PHI.

4. Object Oriented Programming through Java, P. Radha Krishna, Universities Press.

5. Thinking in Java, Bruce Eckel, PE 4. Programming in Java, S. Malhotra and S. Choudhary, Oxford Universities Press.

GIRIJANANDA CHOWDHURY UNIVERSITY
Hathkhowapara, Azara, Guwahati - 781017, Assam

BCS23209L	Java Programming Lab	L	T	P	C
		0	0	2	1

Pre-requisite: Concept of OOP using C++

Course Objectives:

1. Apply the concepts of Java to solve problems.
2. Understand the principles of inheritance and polymorphism.
3. Implementation of packages and interfaces.
4. Understand the concepts of exception handling and multithreading.
5. Implementation of the design of Graphical User Interface using applets and swing controls.
6. Understand the concepts of JDBC, JSP.

Course Outcomes: After successful completion of the course, the students will learn:

CO1: **Develop** Java programs using fundamental and object-oriented concepts such as classes, objects, inheritance, and interfaces to solve real-world problems. (Cognitive Level: Creating)
CO2: **Apply** exception handling and multithreading techniques to create robust and efficient Java applications. (Cognitive Level: Applying)
CO3: **Design** graphical user interfaces (GUI) with AWT and Swing, integrating event-handling mechanisms and layout management. (Cognitive Level: Creating)
CO4: **Construct** database-driven and dynamic web-based applications using JDBC and JSP. (Cognitive Level: Creating)

Module 1: Basics of Java Programming	**6 hours**
1. Write a Java program to demonstrate the usage of data types, variables, constants, and type casting. 2. Implement a program to illustrate conditional statements (if, switch) and loops (for, while, do-while). 3. Develop a program to perform array operations such as insertion, deletion, and searching. 4. Create a Java program to define a class, create objects, and demonstrate the usage of constructors and this keyword. 5. Write a program to demonstrate method overloading and constructor overloading.	
Module 2: Object-Oriented Programming Concepts	**6 hours**
6. Implement single, multilevel and hierarchical inheritance in Java. 7. Write a program to demonstrate the usage of super and final keywords in inheritance. 8. Develop a program to illustrate method overriding. 9. Create a Java program to define interfaces, implement them, and access them through interface references.	

10. Demonstrate the creation and usage of packages and importing user-defined packages.	
Module 3: Exception Handling and Multithreading 11. Write a program to handle checked and unchecked exceptions using try, catch, throw, throws, and finally. 12. Develop a user-defined exception class and use it in a program. 13. Create a multithreaded program to demonstrate thread life cycle and priorities. 14. Implement thread synchronization and inter-thread communication (e.g., Producer-Consumer problem).	**6 hours**
Module 4: GUI Programming and Event Handling 15. Create a Java applet to demonstrate the applet life cycle and parameter passing. 16. Design a GUI application using AWT to handle basic events like button clicks. 17. Develop a Swing-based application to demonstrate the use of JButton, JLabel, JTextField, and JTextArea. 18. Implement layout management using BorderLayout, GridLayout, and FlowLayout. 19. Write a program to handle mouse and keyboard events using adapter classes.	**6 hours**
Module 5: Database and Web Programming 20. Write a Java program to connect to a database using JDBC. 21. Develop a program to use different types of JDBC statements (Statement, PreparedStatement, CallableStatement). 22. Create a simple dynamic web application using JSP to display and process user inputs.	**6 hours**
Total Lecture hours	**30 hours**

	GRAPH THEORY	L	T	P	C
		3	0	0	3

Pre-requisite:

Basic mathematics

Course Objectives:

1. To introduce basic concepts of graph theory and their various properties.
2. To study the structural properties of different types of graphs, including trees, planar graphs, and bipartite graphs.
3. To develop problem-solving skills using graph algorithms like BFS, DFS, shortest path algorithms, and spanning tree algorithms.
4. To explore advanced graph theory topics like graph coloring, matching, and network flows, and examine their applications in optimization and resource allocation.

Course Outcome:

After successful completion of the course, the students will be able to

CO1: **Understand** the basics concepts of graph theory.

CO2: **Analyze** different types of graphs based on their properties and real-world applications.

CO3: **Apply** graph algorithms to solve practical problems such as computing shortest paths, minimum spanning trees, and optimized network flows.

CO4: **Evaluate** real life problems by applying the concepts of graph planarity and matching.

MODULE 1: Introduction — **5 hours**

Graph: terminologies; Incidence and degree; Handshaking Lemma; Types of graphs; Complete, regular, bipartite graphs with their basic properties; Distance in graphs: length, eccentricity, diameter, radius; Graph operations; Isomorphism; Subgraphs and Union of graphs; Adjacency matrix and Incidence matrix of graph.

MODULE 2: Walks, Paths, Circuits — **9 hours**

Walks, Paths, Circuits, Cycles; Components and Connectedness algorithms; Eulerian graphs: properties and theorems; Fleury's algorithm and Chinese postman problem; Hamiltonian graph - necessary and sufficient conditions; Labeled trees: Cayley's theorem; Directed graph

MODULE 3: Trees and Connectivity — **10 hours**

Properties of trees; Pedant vertices in a tree; Center of a tree; Rooted binary trees; Spanning trees and forest, Minimum spanning tree algorithms: Prim's and Kruskal's; Fundamental circuits; Spanning trees of a weighted graph; Menger's theorem

Connectivity: Cut-sets and Cut-vertices; Bridges and blocks; Fundamental cut-sets; auto Orphism groups; Connectivity and separativity; Network flow; Max-flow min-cut theorem; Reconstruction problem

MODULE 4: Planar Graph — **7 hours**

Combinatorial and geometric dual; Kuratowski's graph; Euler's theorem; Detection of

planarity; Thickness and Crossings.	
MODULE 5: Graph Coloring	**5 hours**
Chromatic number: Chromatic polynomial; Vertex and Edge coloring**;** The four colour problem.	
MODULE 6: Matching and Cover	**4 hours**
Matching in bipartite graph; Maximal matching; Vertex cover and edge cover; Decomposition and domination	
MODULE 7: Graph Algorithms	**5 hours**
Breadth-First Search (BFS); Depth-First Search (DFS); Dijkstra's Algorithm for shortest paths; Bellman-Ford Algorithm; Floyd-Warshall Algorithm for all-pairs shortest paths.	
Total Lecture hours	**45 hours**

Text Book

(1) J. A. Bondy and U. S. R. Murthy, Graph Theory, Springer Verlag

(2) Douglas B. West. Introduction to Graph Theory, PHI

(3) N. Deo, Graph Theory with Applications to Engineering and Computer Science.

(4) R. Diestel, Graph Theory, Springer Verlag

Reference Books

(1) G. Chatrand, and O.R. Ollermann, Applied and Algorithmic Graph theory, McGraw Hills.

(2) B. Bollobas, Modern Graph Theory, Springer

	ADVANCED PYTHON PROGRAMMING	L	T	P	C
		3	0	0	3

Pre-requisite: Basic knowledge of python programming

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

1. Understand and apply advanced Python features such as assertions, decorators, generators, iterators, and multi-threading using the threading module.
2. Develop the ability to perform database operations in Python using SQLite, including executing and processing DDL and DML queries.
3. Build interactive GUI applications in Python using Tkinter components like buttons, labels, menus, and message boxes.
4. Gain practical skills in image processing, web scraping with Beautiful Soup, and performing data analysis and visualization using Pandas and Matplotlib.

Course Outcome:

After successful completion of the course, the students will learn

CO1: Understand and explain advanced Python concepts such as assertions, decorators, generators, iterators, and multi-threading.
CO2: Apply Python for database connectivity and execute DDL/DML queries with SQLite
CO3: Design and implement interactive GUIs using Tkinter components.
CO4: Analyze data with Pandas, visualize insights with Matplotlib, and perform basic image processing and web scraping.

MODULE 1: Advanced Python Constructs and Multi-threading	**10hours**

Assertion, Decorators, Generators, Iterators, Creation, and Execution of threads using threading module

Module 2: Database programming using Python	**6 hours**

Connecting to a database (SQLite) using Python, Sending DML and DDL queries and processing the result from a Python Program

Module 3: GUI in Python	**6hours**

Button, Canvas, Checkbutton, Entry, Frame, Label, Listbox, Menubutton, Menu, Message, Radiobutton, Scale, Scrollbar, Text, Toplevel, Spinbox, PanedWindow, LabelFrame, tkMessageBox

Module 4: Basic image processing and web scrapping using Python	**6 hours**

Introduction to digital image processing, Basic operations on an image, Crop, Scale, Rotate, Flip, Changing contrast, brightness and color, Edge detection, blur, sharpening, Beautiful Soup

Module 5: Data analysis and data visualization	**10 hours**

Introduction to Pandas, Pandas data structures – Series and DataFrame, Data wrangling using pandas, Loading a dataset into a dataframe o Selecting Columns from a dataframe, Selecting Rows from a dataframe, Adding new data in a dataframe, Deleting data from a dataframe Introduction to Matplotlib, Scatter plot, Line plot, Bar chart, Histogram, Box plot.	
Total Lecture hours	**38hours**

Text Book

1. Beazley, D., & Jones, B. K. (2013). **Python cookbook: Recipes for mastering Python 3** (3rd ed.). O'Reilly Media.
2. Chollet, F. (2021). **Deep learning with Python** (2nd ed.). Manning Publications.
3. Grinberg, M. (2018). **Flask web development: Developing web applications with Python** (2nd ed.). O'Reilly Media

4. McKinney, W. (2022). **Python for data analysis: Data wrangling with Pandas, NumPy, and Jupyter** (3rd ed.). O'Reilly Media.

5. Ramalho, L. (2022). **Fluent Python: Clear, concise, and effective programming** (2nd ed.). O'Reilly Media

Reference Books

1. Lutz, M. (2013). **Programming Python** (4th ed.). O'Reilly Media.
2. Mitchell, R. (2018). **Web scraping with Python: Collecting data from the modern web** (2nd ed.). O'Reilly Media

3. Slatkin, B. (2020). **Effective Python: 90 specific ways to write better Python** (2nd ed.). Addison-Wesley Professional.

4. Sweigart, A. (2020). **Automate the boring stuff with Python: Practical programming for total beginners** (2nd ed.). No Starch Press.

5. Vincent, W. S. (2022). **Django for professionals: Production websites with Python & Django** (3rd ed.). WelcomeToCode

6. Python Software Foundation. (n.d.). **Python 3 documentation**. Retrieved from https://docs.python.org/3/

	ADVANCED PYTHON PROGRAMMING LAB	L	T	P	C
		-	-	-	-

Pre-requisite: Basic knowledge of python programming

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

1. Develop Advanced Python Programs.
2. Implement Data Handling and GUI-Based Applications
3. Perform Image Processing and Web Scraping
4. Analyze and Visualize Data

Programs:

1. Write a Python program that uses assertions to ensure a user-provided number is positive.
2. Create a decorator that measures and prints the execution time of functions.
3. Implement a generator to yield Fibonacci numbers up to a specified limit.
4. Create a custom iterator for iterating over a list of prime numbers.
5. Write a program to demonstrate multi-threading where each thread calculates the square of a number from a list.
6. Create a program where threads access shared data safely using threading.Lock.
7. Write a program using SQLite Database to store student records. Insert, retrieve, update, delete records and execute query to calculate and display the average grade of students.
8. Design a GUI with buttons that change the text of a label on a button click.
9. Create a form using Entry widgets to accept user input (e.g., name, age) and display it in a label.
10. Build an application to display a list of cities in a Listbox with a Scrollbar.
11. Design a menu-driven application with options like "About" and "Exit," using tkMessageBox to display messages.
12. Write a program to perform cropping, scaling, rotation, and flipping of an image using the Pillow library. And adjust the brightness, contrast, and apply sharpening filters to an image.
13. Scrape the latest news headlines from a website using the BeautifulSoup library and save them to a text file. Also scrape table data from a webpage and store it in a CSV file.
14. Load a dataset into a Pandas DataFrame, select specific columns, filter rows, and add new columns. Perform group-by operations and calculate aggregated metrics such as sum and mean for a dataset.
15. Use matplotlib to visualize relationships in a dataset using scatter plots and line plots. Create histograms and box plots to display data distribution and outliers. Build a bar chart with custom titles, labels, legends, and colors.

Text Book

1. Beazley, D., & Jones, B. K. (2013). **Python cookbook: Recipes for mastering Python 3** (3rd ed.). O'Reilly Media.

2. Chollet, F. (2021). **Deep learning with Python** (2nd ed.). Manning Publications.

3. Grinberg, M. (2018). **Flask web development: Developing web applications with Python** (2nd ed.). O'Reilly Media

4. McKinney, W. (2022). **Python for data analysis: Data wrangling with Pandas, NumPy, and Jupyter** (3rd ed.). O'Reilly Media.

5. Ramalho, L. (2022). **Fluent Python: Clear, concise, and effective programming** (2nd ed.). O'Reilly Media

Reference Books

1. Lutz, M. (2013). **Programming Python** (4th ed.). O'Reilly Media.
2. Mitchell, R. (2018). **Web scraping with Python: Collecting data from the modern web** (2nd ed.). O'Reilly Media

3. Slatkin, B. (2020). **Effective Python: 90 specific ways to write better Python** (2nd ed.). Addison-Wesley Professional.

4. Sweigart, A. (2020). **Automate the boring stuff with Python: Practical programming for total beginners** (2nd ed.). No Starch Press.

5. Vincent, W. S. (2022). **Django for professionals: Production websites with Python & Django** (3rd ed.). WelcomeToCode

6. Python Software Foundation. (n.d.). **Python 3 documentation**. Retrieved from https://docs.python.org/3/

Internal Member/BoS

External Member/BoS

	Indian Science, Engineering and Technology (Past, Present and Future)	L	T	P	C
		3	0	0	3

Pre-requisite:

1. Basic of Mathematics.

Course Objectives:

1. To familiarize learners with major sequential development in Indian science, engineering and technology.
2. To review & strengthen the ancient discovery and research in physics, chemistry, maths, metallurgy, astronomy, architecture, textile, transport, agriculture and Ayurveda etc.
3. To help students to trace, identify and develop the ancient knowledge systems to make meaningful contribution to development of science today.
4. To help to understand the apparently rational, verifiable and universal solution from ancient Indian knowledge system for the scientific, technological and holistic development of physical, mental and spiritual wellbeing.

MODULE 1: Indian Traditional Knowledge; Science and Practices — **8 hours**

Introduction to the Science and way of doing science and research in India, Ancient Science in Intra & Inter Culture Dialogue & coevolution. Traditional agricultural practices, Traditional water-harvesting practices, Traditional Livestock and veterinary Sciences Traditional Houses & villages, Traditional Forecasting, Traditional Ayurveda & plant based medicine, Traditional writing Technology

MODULE 2: Ancient Indian Science (Physics, Chemistry, Maths) — **9 hours**

Physics in India: Vaisheshika darshan Atomic theory & law of motion, theory of panchmahabhoota, Brihath Shathaka (divisions of the time, unit of distance), bhaskarachaya (theory of gravity, surya siddhanta & sidhanta shriomani), Lilavati (gurutvakashan Shakti). Chemistry in India Vatsyayana, Nagarjuna,Khanda, Al-Biruni, Vagbhaṭa –building of the ras-shala (laboratory), working arrangements of ras-shala, material and equipment, Yaśodhara Bhaṭṭa-process of distillation, apparatus, saranasamskara, saranataila Mathematics in India: Baudhayana's Sulbasutras, Aryabhaṭa, Bhaskaracharya-I, Severus Sebokht, Syria, Brahmagupta, Bhaskaracharya-II, Jyeṣṭhadeva

MODULE 3: Ancient Indian Science (metallurgy, Astronomy, Architecture) — **9 hours**

Metallurgy in India: Survarṇa(gold) and its different types, prosperities, Rajata(silver), Tamra(copper), Loha(iron), Vanga(tin), Naga / sisa(lead), Pittala(brass) Astronomy in India Vedang Jyotish, aryabhatta siddhanta, Mahabhaskriya, Laghubhaskariya, vatesvarasiddhanta, Sisyadhivrddhida, Grahashyay, Goladhyaya, Karabakutuhala (Aryabhaṭa, Varahamihira, Brahmagupta, Vaṭesvara, Bhaskara, Paramesvara, NilakaṇṭhaSomayaji, Jyeṣṭhadeva, ŚankaraVarman) Architecture in India: Nagara (northern style), Vesara (mixed style), and Dravida (southern style), Indian vernacular architecture, Temple sytle, cave architecture, rock cut architecture, kalinga architecture, chandels architecture, rajput architecture, jain architecture, sikh architecture, Maratha architecture Indo-Islamic architectural, Indo-Saracenic revival architecture, Greco Buddhist style.

MODULE 4: Ancient Indian Science (Textile, Agriculture, Transport) — **9 hours**

Textile Technology in India: Cotton (natural cellulose fiber), silk, wool (natural protein fibers), bast and leaf fibers, mridhudhautadhupitambaram (meaning a practice of fumigating the fabric with incence smoke before use as a part of the finishing process), sitadhautavasanayugala (bleached white–a finishing process); suchhastah, sutradharah (needle and thread – tools for stiching). dyeing, washing spinning and weaving technology, Agriculture in India: krishisuktas,

Krishiparashara, Brihatsamhita, Types of crops, Manures, Types of land- devamatruka, nadimatruka, use of animals in warfare, animal husbandry, Animals for medicines. Ancient transport in India	
MODULE 5: Ancient Indian Science (Ayurveda & Yoga)	**7 hours**
Ayurveda for Life, Health and Well-being: Introduction to Ayurveda: understanding Human body and Pancha maha bhuta, the communication between body & mind, health regimen for wellbeing, introduction to yoga (raja yoga, astang yoga, gyan yoga), understanding of Indian psychological concept, consciousness, tridosha & triguna.	
Total Lecture hours	**42 hours**
Text Book/Reference Book:	
1. Textbook on IKS by Prof. B Mahadevan, IIM Bengaluru.	
2. Kapur K and Singh A.K (Eds) 2005). Indian Knowledge Systems, Vol. 1. Indian Institute of Advanced Study, Shimla. Tatvabodh of sankaracharya, Central chinmay mission trust, Bombay, 1995.	
3. Nair, Shantha N. Echoes of Ancient Indian Wisdom. New Delhi: Hindology Books, 2008.	
4. SK Das, The education system of Ancient hindus, Gyan publication house, India	
5. R P Kulkarni, Glimpese of Indian Engineering and Technology (Ancient & Medieval period, Munshiram Manoharlal Publishers Pvt. Ltd. 2018	
6. AK Pathak, Science and Technology in India, Anshika prakashan pratapgarh, 2016	
7. PB Sharma, S. Narain, Doctors Scientists and Engineers of Ancient India, Kalpaz Publications 2017	
8. NVP, Unithiri, Indian Scientific Traditions (Professor K.N. Neelakantan Elayath Felicitation Volume), publication division unieristy of Calicut, 2006	
9. Anonyms, History of Science in India- Volume-I Part-I (Physics, Mathematics and Statistics), the national academy of science, India & the ramkrishna mission institute of culture, 2014	
10. R N Basu, T K Bose, CS, Cakraborty History of Science in India - Agricultural Science (Volume V), the national academy of science, India & the ramkrishna mission institute of culture 2014	
11. A Gosh, History of Science in India (Volume-I Part-II Astronomy), the national academy of science, India & the ramkrishna mission institute of culture, 2014	
12. Dharmpal, Indian science and technology in the eighteen century, rashtrottahana sahitya, 1983	
13. S Biswal, B L ray, vedic Science and technology, DK Print world, 2009	
14. A.K Bag, Histroy of technology in Indian (Set 3 vol), Indian Nation Science Academy, 1997.	
15. AR vasudev Murty, Science and Technology in Ancient India as Reflected in the Mahabharata, Sanskrit bharati, 2019	

School of Engineering & Technology

B.Tech. –Computer Science and Engineering

Semester III Course Structure & Syllabus

Theory/ Practical	Sl. No	Course Type	Course Code	Course Name	Hours per week			Credit
					L	T	P	C
T	1.	ESC	BCS23201T	Discrete Mathematics	3	1	0	4
T	2.	PCC	BCS23202T	Object Oriented Programming using C++	3	0	0	3
P	3	PCC	BCS23202P	Object Oriented Programming using C++ LAB	0	0	4	2
T	6	PCC	BCS23203T	Data Structure and Algorithms	3	0	0	3
P	7	PCC	BCS23203P	Data Structure and Algorithms Lab	0	0	4	2
T	4	PCC	BCS23204T	Python Programming	3	0	0	3
P	5	PCC	BCS23204P	Python Programming Lab	0	0	2	1
T	8	ESC	BCS23205T	Digital Electronics	3	0	0	3
T	9	HSMC	BCH23112T	Environmental Science	2	0	0	0
				Total	**17**	**1**	**10**	**21**

Discrete Mathematics	L	T	P	C
	4	0	0	4

Pre-requisite:

1. Basic of Mathematics.

Course Objectives:

1. To introduce the concept of set theory.
2. To introduce about the vector spaces
3. Perform basic matrix operations including sums, products, and transpose etc.

Course Outcome:

After successful completion of the course, the students will learn

CO1: Solve and analyses real world engineering problems by applying set theory, relations, construct and use functions and apply these concepts to solve problems

CO2: Be able to apply the fundamental concepts of Partial differential Equations.

CO3: Analyze and apply the concepts of Matrices and propositional logic to solve problems and to construct proofs using mathematical induction

CO4: be able to construct simple mathematical proofs and possess the ability to analyze them.

MODULE 1: Set, Relation and Function — **10 hours**

Sets, relations, properties of binary relations, closures of relation, equivalence relations, equivalence classes and partitions. Partial ordering relations.

MODULE 2: Matrices — **10 hours**

Row and column operations, vectors and matrices, partitioning of matrices, representing relations using matrices, Determinant of a square matrix, minor, cofactor, the Cayley Hamilton theorem, inverse of a matrix, product form of inverse. Rank of a matrix. Solutions of simultaneous linear equations, existence of solutions, solution by Gaussian elimination, Eigen values and Eigen vectors.

MODULE 3: Number Theory — **10 hours**

Basic of counting principles, principle of inclusion exclusion, application of inclusion and exclusion. Pigeonhole principle, generalized Pigeonhole principle and its application, permutations and combinations, permutations with repetitions, combinations with repetitions, permutations of sets with indistinguishable objects.

MODULE 4:Partial Differential Equation — **10 hours**

First order Partial differential equation: Partial differential equation of first order, Linear partial differential equation, Non-linear partial differential equation Homogenous and non-homogeneous partial differential equation with constant coefficient

MODULE 5: Vector Spaces — **10 hours**

Definition and examples of vector spaces. Elementary properties of R as a vector space. Subspaces of a vector space. Union, intersection and sum of two subspaces. Subspaces generated by a subset of a vector space. Definition, example and properties of linearly independent and dependent set of vectors. Basis and dimension of a vector space. Examples of finite dimensional

vector spaces.	
MODULE 6: Logic	**10 hours**
Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers.	
Total Lecture hours	**60 hours**
Text Book/Reference Book:	
1. E. Kreyszig," Advanced Engineering Mathematics:, Eighth Edition, Wiley India.	
2. B.V. Ramana, "Higher Engineering Mathematics", McGraw Hil Education.	
3. N.P.Bali and Manish Goel, " A text book of Engineering mathematics", Laxmi Publication.	
4. B. S. Grewal ,"Higher Engineering Mathematics", Khanna Publication, Delhi.	
5. Discrete Mathematics and Its Applications, by Kenneth H. Rosen, Tata McGraw Hill, 6th edition, ISBN: 0072880082© 2007.	
6. Elements of Discrete Mathematics, by C. L. Liu, Tata McGraw Hill Education Private Limited, 3rd edition,2008	
7. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press, Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson	
8. J.P. Tremblay and R. Manohar, Discrete Mathematical Structure and Its Application to Computer Science", TMG Edition, Tata McGraw-Hill	

	OBJECT ORIENTED PROGRAMMING USING C++	L	T	P	C
		3	0	0	3

Pre-requisite:

1. Basic of computer knowledge.
2. Knowledge of C programming.

Course Objectives:

1. To introduce the concept of object orientation to C++.
2. To introduce solving real world problems.
3. It will help in acquainting the techniques and applications of C++ for programming based onchallenging tasks.

Course Outcome:

After successful completion of the course, the students will learn

CO1: Develop the concept of object-oriented programming, its applications, and its differences with procedure-oriented programming.
CO2: Implement the concept of class, object, and constructor in writing codes for solving problems.
CO3: Apply the concept of polymorphism, inheritance in solving engineering problems.
CO4: Design programmed solution using the concept of files and pointers including templates, exceptions and file handling.

.

MODULE 1: Introduction to Object Oriented Programming — **5 hours**

Introduction to Object Oriented Programming: Computer programming background- C++ overview. First C++ Program -Basic C++ syntax, Object Oriented Programming: Concept of object, Classes, methods and messages, abstraction and encapsulation, inheritance, abstract classes, polymorphism.

MODULE 2: Basics of C++ Programming — **6 hours**

C++ Program Structure, Character Set and Tokens, Data Type, Type Conversion, Preprocessor Directives, Namespace, Input/output Streams, Control Statements.
Keywords, Identifiers and constants, Operators in C++, Scope resolution operator. Function: Function Overloading, Inline Function, Default Argument, Expressions, Call by Value, Call by reference – Return by reference.
Pointers: Pointer variable declaration & initialization, Operators in pointers, Pointers and Arrays, Pointer and Function.

MODULE 3: Class and Objects — **12 hours**

A Simple Class and Object, Accessing members of class, Initialization of class objects (Constructor, Destructor), Default Constructor, Parameterized Constructor, Copy Constructor , Default Constructor, Objects as Function Arguments, Returning Objects from Functions, Memory allocation for Objects, Static members, Member functions defined outside the class.

MODULE 4:Operator Overloading	**3 hours**
Fundamentals of operator overloading, Restriction on operator overloading, Operator function as a class member, Overloading unary and binary operator.	
MODULE 5: Inheritance	**10 hours**
Introduction to inheritance, Derived Class and Base Class, Access Specifiers (private, protected, and public), Types of inheritance, Constructor and Destructor in derived class, Aggregation.	
MODULE 6: Polymorphism, Virtual Function and miscellaneous C++ features	**4hours**
Concept of Virtual functions, Late Binding, Abstract class and pure virtual functions, Virtual Destructor, Virtual Base class, Friend function and static function, inline function, This pointer, Concrete classes, Polymorphism and its roles.	
MODULE 7: File handling, Function Templates and Exception Handling	**5 hours**
Stream Class Hierarchy for Console Input/output, Function template, Function template with multiple arguments, Class template, Exception Handling (Try, throw and catch), Use of exception handling.	
Total Lecture hours	**45 hours**

Text Book:

1.E Balaguruswamy (2013), Object-oriented programming with C++, 6th Edition,
Mc GrawHill Education.

2. Bjarne Stroustrup (2013), The C++ Programming Language, 4th Edition, Addison-Wesly.

3. Herbert Schildt (2017), C++: The Complete Reference, 4th Edition, McGraw Hill Education.

Reference Books:

1. Reema Thareja,(2016),Object Oriented Programming with C++,1st Edition, Oxford University.

OBJECT ORIENTED PROGRAMMING USING C++ LABORATORY		
List of Lab Experiments		
Lab 1	Basics of C++ Programming.	**2hours**
Lab 2	WAP using Switch Case to add, subtract, multiply and divide of two numbers.	**1 hour**
Lab 3	WAP to illustrate Class Declaration, Definition, Member function, objects. (Area of Trapezium, Rhombus, Circle, Triangle).	**4 hours**
Lab 4	WAP to create a simple class named Account and write methods to deposit and withdraw amount from the account.	**2 hours**
Lab 5	WAP to demonstrate the usage of a Constructor and Destructor in a class.	**1 hour**
Lab 6	WAP to illustrate parameterized constructor.(default, copy constructor)	**2 hours**
Lab 7	WAP to demonstrate: a) Operator Overloading b) Function Overloading.	**2 hours**
Lab 8	WAP to demonstrate Hybrid Inheritance. (Single, Multiple, Multi-level, Hierarchical)	**5 hours**
Lab 9	WAP to demonstrate Friend Function and Friend class.	**2 hours**
Lab 10	WAP to demonstrate polymorphism by calculating area of a rectangle and triangle using Shape class.	**2 hours**
Lab 11	WAP to demonstrate Virtual function.	**2 hours**
Lab 12	WAP to overload +operator to add two numbers.	**1hour**
Lab 13	WAP to create a Class Template.	**2 hours**
Lab 14	WAP to demonstrate exception handling.	**2 hours**
	Total	**30hours**

Text Book:

(4) E Balaguruswamy (2013), Object-oriented programming with C++, 6th Edition, McGrawHill Education.

(5) Bjarne Stroustrup (2013), The C++ Programming Language, 4th Edition, Addison-Wesly.

(6) Herbert Schildt (2017), C++: The Complete Reference, 4th Edition, McGraw Hill Education.

Reference Books

(2) Reema Thareja,(2016),Object Oriented Programming with C++,1st Edition, Oxford University.

Course Title	Hours per week L-T-P	Credit C
Data Structure and Algorithms	**3-0-4**	**5**

Course Outcome (Theory)

Course Outcome	Statement
CO1	Analyze the performance of various algorithms
CO2	Make use of the functionality of linear data structure
CO3	Model the solutions using non-linear data structure
CO4	Choose appropriate searching and sorting technique for a given problem
CO5	Implementation of graphs

MODULE 1: (3 Lectures)

Notion of datastructures and algorithms, understanding growth of functions, Worst-case, average case and best case time/space complexity, Asymptotic Notation.

MODULE 2: (8 Lectures)

Abstract data-type (ADTs): arrays and linked list ADTs, Stacks, Queues: ADTs and implementations using arrays, linked lists, Doubly linked lists: ADT and implementation, Dictionary ADT: implementation using array, linked lists, binary search, Tree ADT and examples, Implementation of trees and basic traversal algorithms.

MODULE 3: (3 Lectures)

Priority Queue ADT, Definition of heaps, Implementation of Priority Queues using heaps and running time analysis, Implementation of heaps using arrays, Heap-sort.

MODULE 4: (9 Lectures)

Binary Search Trees: definition and some basic algorithms, Implementation of Dictionary ADTs using Binary Search trees and running time analysis, AVL trees: height balance condition, rotations, and implementation of dictionary ADT, 2-4 Trees: Multi-way search trees, implementation of dictionary ADT, Informal discussion of extension to B-trees.

MODULE 5: (6 Lectures)

Map ADT, Hash Tables and implementation of Map using Hash Tables, Design of hash functions, Collision resolution schemes: chaining, open addressing schemes like linear probing, quadratic probing, double hashing, Applications of Hashing: finding duplicates, set

intersection, etc, Tries: implementation of Map ADT using tries, Compressed tries and suffix tries.

MODULE 6: (4 Lectures)

Bubble sort, insertion sort, selection sort, Merge sort and divide and conquer paradigm, Quick sort: average and worst case analysis, randomized quicksort, Selection based on partitioning ideas used in Quick Sort.

MODULE 7: (8 Lectures)

Graph ADTs and applications, Adjacency list and adjacency matrix representations and relative merits, Basic graph definitions: paths, cycles, trees, spanning trees, connected components, Euler's formula, Depth First Search Traversal algorithm for directed graphs: classification of edges into forward, back and cross edges. Applications to cycle finding, topological sort in directed acyclic graphs, finding connected components. Running time analysis, Breadth first search algorithm: implementation using queues, shortest path tree property. Running time analysis.

Text Books/ Reference Books:

1."Data Structures and Algorithms in Java", by Michael T. Goodrich and Roberto Tamassia, John Wiley & Sons; 3rd Edition.
2."Data Structures and Algorithms in Python", by Michael T. Goodrich and Robert, Tamassia, Wiley, 1st Edition.
3. In case any other programming language is used for this course, some other suitable text book may be chosen

GIRIJANANDACHOWDHURYUNIVERSITY
Hathkhowapara,Azara,Guwahati781017,Assam

BSC23203P	DATA STRUCTURE AND ALGORITHMS	L	T	P	C
		0	0	4	2

LIST OF EXPERIMENTS

1. Write a program to implement bubble sort, insertion sort and selection sort in a menu driven program.
2. Write a program to perform linear search and binary serch.
3. Write a program to perform operations in an array.
4. Write a program to implement stack using array.
5. Write a program to implement queue using array.
6. Write a program to implement circular queue, priority queue.
7. Write a program to implement singly linked list along with operations.
8. Write a program to implement circular doubly linked list along with operations.
9. Write a program to create a binary search tree with operations of searching and deletion.
10. Write a program to perform traversal of a binary search tree.
11. Write a program to represent graph in memory and perform breadth first search and depth first search on this graph.

Python Programming	L	T	P	C
	3	0	0	3

Pre-requisite: Basic knowledge of Programming

Course Objectives:

1. To be able to introduce core programming basics and program design with functions using Python programming language.
2. To understand a range of in-built functions as well as in-depth data and information processing techniques.

Course Outcome:

After successful completion of the course, the students will be able to

CO1: Interpret the fundamental Python syntax and semantics and determine the methods to create and manipulate Python programs by utilizing the data structures like lists, dictionaries, tuples and sets.
CO 2: Express proficiency in the handling of control flow statements, strings and functions.
CO 3: Identify the commonly used operations involving file systems and regular expressions.
CO 4: Evaluate the python in built functions, Scientific computing packages
CO 5: Evaluate the various feature engineering algorithms by python programming.

Module 1: Introduction to Programming in Python	**10 hours**

Introduction to Programming in Python: What Is Python? Features of Python, Python environment set up: Installing Python, Running Python, Python Documentation, Structure of a Python Program Basics of Programming in Python: Input statement, output statement, variables, operators, numbers, Literals, strings, lists and tuples, dictionaries.

Module 2: Conditionals, Loops and Functions.	**12 hours**

Conditionals and Loops: if statement, else Statement, elif Statement, while Statement, for Statement break Statement, continue Statement, pass Statement. Functions: Built-in Functions, User defined functions: Defining a Function, Calling a Function, Various Function Arguments.

Module 3: Files, Modules and Introduction to Advanced Python.	**10 hours**

Files: File Objects, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Scientific computing packages: **Numpy:** -Array object, creating array, matrix, indexing, slicing, resizing, reshaping, arithmetic operations, functions, matrices and vector operations, **Matplotlib**: basic plotting, **Scipy:** Linear algebra operations, equation solving. Regular Expressions.

Module 4: Introduction to advance Python Programming	**7 hours**

OOPS

Module 5: Python GUI & CGI Programming and Python database connectivity.	**6 hours**

Python GUI Programming (Tkinter): Tkinter Programming example, Tkinter widges, standard attributes, geometry management

Books

1. Kamthane, A. N., & Kamthane, A.A. Programming and Problem Solving with Python, McGraw Hill Education. 2017.
2. Balaguruswamy E., "Introduction to Computing and Problem Solving using Python",2nd Edition, McGraw Hill Education, 2018.
3. Taneja, S., Kumar, N. "Python Programming- A modular Approach", Pearson Education India, 2018.

Additional References

(i) Mark Lutz, "Learning Python" O'Reilly Media, 2013.

(ii) Guttag, J. V. Introduction to computation and programming using Python. MIT Press. 2018

(iii) Downey, A. B. Think Python–How to think like a Computer Scientist 2nd Edition. O'Reilly 2015

(iv) Robert Johansson, "Numerical Python: Scientific Computing and Data Science .Applications with Numpy, SciPy and Matplotlib" Apress, 2019.

Python Programming LAB	L	T	P	C
	0	0	2	1

Pre-requisite: Basic knowledge of Programming

Course Outcome:

After successful completion of the course, the students will be able to
CO1: Write, Test and Debug Python Programs.
CO 2: Implement Conditionals and Loops for Python Programs
CO 3: Use functions and represent Compound data using Lists, Tuples and Dictionaries.
CO 4: Read and write data from & to files in Python and develop Application

Suggested Practical List	**15 hours**

1. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:
   Grade A : if Percentage >=80
   Grade B : if Percentage >=60 and Percentage <80
   Grade C : if Percentage >=40 and Percentage <60
   Grade D : if Percentage <=40
2. WAP to print factors of a given number.
3. WAP to add N natural numbers and display their sum.
4. WAP to print the following conversion table (use looping constructs):

   Height(in Feet)
   Height(in inches)
   5.0ft
   60 inches
   5.1ft
   61.2inches
   .
   .

   5.8ft 69.6inches
   5.9ft 70.8inches
   6.0ft 72inches

5. WAP that takes a positive integer n and the produce n lines of output as shown:

   *

   * *

   * * *

   * * * *
   (for n =4)

6. Write a menu driven program using user defined functions to print the area of rectangle, square, circle and triangle by accepting suitable input from user.

7. Write a function that calculates factorial of a number n.

8. WAP to print the series and its sum: (use functions)
   1/1! + 1/2! + 1/3! …….1/n!
9. WAP to perform the following operations on an input string

   a. Print length of the string

   b. Find frequency of a character in the string

   c. Print whether characters are in uppercase or lowercase

10. WAP to create two lists: one of even numbers and another of odd numbers. The program should demonstrate the various operations and methods on lists.
11. WAP to create a dictionary where keys are numbers between 1 and 5 and the values are the cubes of the keys.
12. WAP to create a tuple t1 = (1,2,5,7,2,4). The program should perform the following:
    a. Print tuple in two lines, line 1 containing the first half of tuple and second line having the second half.
    b. Concatenate tuple t2 = (10,11) with t1.

	DIGITAL ELECTRONICS	L	T	P	C
		3	0	0	3

Pre-requisite: Basics of logic circuits, Boolean algebra

Course Objectives:

1. To provide the fundamental concepts associated with the digital logic and circuit design.
2. To introduce the basic concepts and laws involved in the Boolean algebra and logic families and digital circuits.
3. To familiarize with the different number systems, logic gates, and combinational and sequential circuits utilized in the different digital circuits and systems.

Course Outcome:

After successful completion of the course, the students will learn

1. CO1: To solve the Boolean algebra, logic gates, logical variables, the truth table, number systems, codes, and their conversion from to others.
2. CO4: To analyse various types of analog to digital converter, memory elements and programmable logic arrays and properties of Digital logic families and programmable logic devices.
3. CO2: To design the most simplified combinational circuit using mapping techniques.
4. CO3: To design different types of sequential circuits using flip flops and counters.

MODULE 1: Introduction to number systems and Boolean Algebra — **10 hours**

Data and number system: Binary, Octal and Hexadecimal representations and their conversion, BCD, ASCII, EBDIC, Gray codes, code conversion, Error detection and correction codes - parity check codes and Hamming code. Representation of Signed binary numbers with 1's and 2's complement methods, Binary arithmetic, Logic circuits, integrated circuits;

Axiomatic definitions of Boolean Algebra - Basic Theorems and Properties of Boolean Algebra, Boolean Functions- Canonical and Standard forms ,Digital Logic Gates, Simplification of Boolean Expressions: The map method- SOP and POS ,NAND and NOR implementation ,Don't Cares - The Tabulation Method – Determination and Selection of Prime Implicants.

Module 2: Combinational logic circuits — **8hours**

Basic logic operation and logic gates, Combinational logic circuit design using truth-table, Different Adders and Subtractors, Comparator, Encoder, Decoder, Multiplexer, Demultiplexer, BCD arithmetic, carry look ahead adder, serial adder, popular MSI chips, digital comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices, Q-M method of function realization. Synthesis of combinational logic circuits.

Module 3: Sequential Logic systems	**12 hours**
Basic sequential circuits- latches and flip-flops: SR-latch, D-latch, D flip-flop, T flip-flop, JK flip-flop; race around condition, master slave conversion of flip-flops, Edge triggered FF, applications of flip flops, Sequential shift register, sequence generator, serial to parallel converter, parallel to serial converter, ring counter, ripple(Asynchronous) counters, synchronous counters, counters design using flip flop, applications of counters.	
Module 4: Logic Families and Semiconductor Memories	**12hours**
Brief idea about DTL, TTL, ECL, MOS and CMOS families and their comparison based on Parameters: fan-in, fan-out, propagation delay, speed-power product, etc. TTL NAND gate, Tristate TTL, ECL, CMOS families and their interfacing. Memory elements, Read-only memory, read/write memory - SRAM and DRAM Concept of Programmable logic devices like PLAs, PALs and their applications, Introduction to field programmable gate arrays (FPGAs), analog to digital converter: quantization and encoding, different types of conversion, accuracy and resolution.	
Total Lecture hours	**42 hours**

Text Book

1.R.P. Jain, “Modern digital Electronics”, Tata McGraw Hill, 4th edition, 2009.

2.S.Salivahanan and S.Arivazhagan, DIGITAL CIRCUITS AND DESIGN, 5th edition,2018

3. Fundamental of digital circuits by A. ANAND KUMAR, PHI Publication.

Reference Books

1.D. P. Kothari and J. S Dhillon, ―Digital Circuits and Design‖, Pearson, 2016,

2. Morris Mano, ―Digital Design‖, Prentice Hall of India, Third Edition.

3..Charles H Roth Jr., Larry L. Kinney ―Fundamentals of Logic Design, Cengage Learning, 7th Edition

4.Digital Fundamentals by FLOYD & JAIN, Pearsons Publication

5.Fundamentals of Logic Design by Charles H. Roth Thomson

Internal Member/BoS External Member/BoS

BCH23112T	ENVIRONMENTAL SCIENCE	L	T	P	C
		2	0	0	0

Pre-requisite: Biology, Sociology, Chemistry

Course Objectives: The objectives of this course are to:

1. To examine the concept of environment and ecosystem.
2. To understand the different types of natural resources and the concept of biodiversity and its importance for the environment.
3. To examine the concept of different types of environmental problems such as pollution, climate change, population growth and its causes, effects on environment and to find out the solution to control the environmental degradation.

Course Outcome: After successful completion of this course, the students will be able to

CO1 : Understand the basic concept of environment and ecosystem.
CO2: Value the natural resources, conservation of biodiversity and its importance.
CO3: Evaluate the problems of environmental issues such as pollution, population growth, climate change and its impact on human and environment and the control measures.

Module1: Concepts of Environmental Science	**3 hours**

Definition of environment, scope and importance of environmental studies; Need for public awareness; Structure and functions in an ecosystem.

Module 2: Natural Resources	**6 hours**

Renewable and Non-renewable Resources;
Forest, water, minerals, food and land resources (with example of one case study); Energy, growing energy needs, energy sources (conventional and alternative).

Module 3: Biodiversity And Its Conservation	**5 hours**

Biodiversity at global, national and local levels; India as a mega diversity nation;
Threats to biodiversity (biotic, abiotic stresses), and strategies for conservation.

Module 4: Environmental Pollution	**8 hours**

Types of pollution-Air, water (including urban, rural, marine), soil, noise, thermal, nuclear; Pollution prevention; Management of pollution –Rural /Urban/Industrial waste management [with case study of any one type, e.g., power (thermal/nuclear), fertilizer, tannin, leather, chemical, sugar], Solid/Liquid waste management, disaster management.

Module 5: Social Issues and Environment	**8 hours**

From unsustainable to sustainable development; Problems relating to urban environment-Population pressure, water scarcity, industrialization, remedial measures;
Climate change-Reasons, effects (global warming, ozone layer depletion, acid rain) with one case study; Legal Issues-Environmental legislation (Acts and issues involved), Environmental ethics.

Total Lecture hours	**30 hours**

Text Book(s)

1	S.C. Santra: Environmental Science, New Central Book Agency
2	S.E. Manahan: Environmental Chemistry
3	K.V. Krishnamurthy: Textbook of Biodiversity

Reference Book(s)

1	Agarwal, K.C., Environmental Biology, Nidi Publication Ltd., Bikaner, 2001.
2	Bharucha Erach, Biodiversity of India, Mapin Publishing Pvt. Ltd., Ahmadabad, 2002
3	Dr R J Ranjit Daniels and Dr Jagadish Krishnaswamy, Environmental studies-2010-Willey India