GLOBAL INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
COURSE CATALOGUE
REGULATIONS B.TECH – GR - 24
COMPUTER SCIENCE AND ENGINEERING
III YEAR I SEMESTER

Course Code	Course Name	Subject Area	Category	Periods Per Week			Credits	Scheme of Examination Max Marks		
				L	T	P		CIA	SEE	Total
THEORY										
CS501PC	Design and Analysis of Algorithms	PCC	Core	3	1	0	4	40	60	100
CS502PC	Cryptography and Network Security	PCC	Core	3	0	0	3	40	60	100
CS503PC	DevOps	PCC	Core	3	0	0	3	40	60	100
	Professional Elective-I	PEC	Elective	3	0	0	3	40	60	100
	Professional Elective -II	PEC	Elective	3	0	0	3	40	60	100
PRACTICAL										
CS504PC	DevOps Lab	PCC	Core	0	0	2	1	40	60	100
CS505PC	Cryptography and Network Security Lab	PCC	Core	0	0	2	1	40	60	100
EN506HS	Advanced English Communication Skills Laboratory	HSMC	Foundation	0	0	2	1	40	60	100
CS507PC	UI design- Flutter	PCC	Core	0	0	2	1	40	60	100
			Total Credits	**15**	**1**	**8**	**20**			

Professional Elective – I

CS511PE	Quantum Computing
CS512PE	Advanced Computer Architecture
CS513PE	Data Analytics
CS514PE	Image Processing
CS515PE	Principles of Programming Languages

Professional Elective – II

CS521PE	Computer Graphics
CS522PE	Embedded Systems
CS523PE	Information Retrieval Systems
CS524PE	Distributed Databases
CS525PE	Natural Language Processing

COURSE CONTENT

DESIGN AND ANALYSIS OF ALGORITHMS								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS501PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**1**	**-**	**4**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: 16	Practical Classes: Nil				**Total Classes: 64**		
Prerequisite: 1. A course on "Computer Programming and Data Structures". 2. A course on "Advanced Data Structures".								

1. COURSE OVERVIEW

The Design and Analysis of Algorithms course focuses on the fundamental principles of algorithm design and performance evaluation. It introduces students to mathematical techniques for analyzing algorithms and explores various algorithmic paradigms such as divide and conquer, greedy methods, dynamic programming, backtracking, and branch and bound. The course also provides insights into computational complexity, including tractable and intractable problems, and introduces NP-Hard and NP-Complete problems. By the end of the course, students will be equipped to design efficient algorithms and analyze their performance for real-world applications.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Introduces the notations for analysis of the performance of algorithms and the data structure of disjoint sets.
2) Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate.
3) Describes how to evaluate and compare different algorithms using worst-, average-, and best case analysis.
4) Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NP complete.

3. COURSE OUTCOMES

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

CO 1	Analyze algorithmic problems using divide-and-conquer techniques and evaluate their time and space complexities using asymptotic notations.
CO 2	Apply data structures and backtracking techniques to solve combinatorial and graph-related problems such as disjoint sets, priority queues, N-Queens, and graph coloring.
CO 3	Formulate and solve optimization problems using dynamic programming techniques for applications such as shortest paths, knapsack, traveling salesperson, and reliability design.
CO 4	Apply greedy strategies and traversal techniques to design efficient algorithms for scheduling, spanning trees, shortest paths, and graph connectivity problems.
CO 5	Analyze complex optimization problems using branch and bound techniques and classify computational problems based on NP-Hard and NP-Complete complexity classes.

4. COURSE CONTENT

UNIT – I **(12L)**

Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

UNIT - II **(14L)**

Disjoint Sets: Disjoint set operations, union and find algorithms, Priority Queue- Heaps, Heapsort

Backtracking: General method, applications, n-queen's problem, sum of subsets problem, graph Coloring, hamitonian cycles.

UNIT - III **(14L)**

Dynamic Programming: General method, applications- Optimal binary search tree, 0/1 knapsack problem, All pairs shortest path problem, Traveling salesperson problem, Reliability design.

UNIT - IV **(12L)**

Greedy method: General method, applications-Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected components, Biconnected components.

UNIT - V **(12L)**

Branch and Bound: General method, applications - Traveling salesperson problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP-Hard and NP-Complete classes, Cook's theorem.

5. TEXT BOOKS

1) Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharan, University press, 1998.

6. REFERENCE BOOKS

1) Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education.
2) Introduction to Algorithms, second edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, PHI Pvt. Ltd./ Pearson Education.
3) Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R. Tamassia, John Wiley and sons.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**3**	**2**	**2**	**1**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**–**
CO 2	**3**	**3**	**3**	**2**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**–**
CO 3	**3**	**3**	**3**	**3**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**–**
CO 4	**3**	**3**	**3**	**2**	**2**	**1**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 5	**3**	**3**	**2**	**3**	**1**	**–**	**–**	**–**	**–**	**–**	**–**	**3**	**3**	**–**

COURSE CONTENT

CRYPTOGRAPHY AND NETWORK SECURITY								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS502PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: Nil								

1. COURSE OVERVIEW

The *Cryptography and Network Security* course provides a comprehensive understanding of the principles and practices used to secure computer systems and networks. It introduces fundamental security concepts such as confidentiality, integrity, authentication, and availability, along with cryptographic techniques for secure communication. The course covers symmetric and asymmetric encryption algorithms, message authentication, digital signatures, key management, network security protocols, and real-world security case studies. By the end of the course, students will be equipped to analyze security threats and apply cryptographic mechanisms to protect information systems.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the fundamental concepts and principles of information and network security.
2) To study symmetric and asymmetric cryptographic algorithms and their applications.
3) To analyze security threats, attacks, and vulnerabilities in computer networks.
4) To understand authentication mechanisms, digital signatures, and key management techniques.
5) To learn security protocols and technologies for secure communication and data protection.

3. COURSE OUTCOMES

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

CO 1	Explain the principles of confidentiality, integrity, authentication, and availability in secure systems.
CO 2	Apply symmetric and asymmetric cryptographic algorithms for secure data communication.
CO 3	Implement message authentication, cryptographic hash functions, and digital signatures.
CO 4	Analyze network security mechanisms including IPSec, SSL/TLS, and wireless security protocols.
CO 5	Identify security threats, intrusion techniques, and legal and ethical issues related to information security.

4. COURSE CONTENT

UNIT – I **(10L)**

Security Concepts: Introduction, need for security, security approaches, principles of security, types of security attacks, security services, security mechanisms, model for network security.

Cryptography Concepts and Techniques: Plain text and cipher text, substitution and transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, types of attacks.

UNIT – II **(10L)**

Symmetric Key Ciphers: Block cipher principles, DES, AES, Blowfish, RC5, IDEA, block cipher modes of operation, stream ciphers, RC4.

Asymmetric Key Ciphers: Principles of public-key cryptosystems, RSA algorithm, ElGamal cryptography, Diffie–Hellman key exchange, Knapsack algorithm.

UNIT – III **(10L)**

Cryptographic Hash Functions: Message authentication, Secure Hash Algorithm (SHA-512).

Message Authentication Codes: Authentication requirements, HMAC, CMAC.

Digital Signatures: ElGamal digital signature scheme.

Key Management and Distribution: Symmetric key distribution using symmetric and asymmetric encryption, public key distribution, Kerberos, X.509 authentication service, Public Key Infrastructure (PKI).

UNIT – IV **(9L)**

Transport-Level Security: Web security considerations, Secure Socket Layer (SSL), Transport Layer Security (TLS), HTTPS, Secure Shell (SSH).

Wireless Network Security: Wireless security issues, mobile device security, IEEE 802.11 WLAN, IEEE 802.11i WLAN security.

UNIT – V **(9L)**

E-Mail Security: Pretty Good Privacy (PGP), S/MIME.

IP Security: IPSec overview, IPSec architecture, Authentication Header (AH), Encapsulating Security Payload (ESP), security associations, Internet Key Exchange (IKE).

Case Studies: Secure multiparty computation, virtual elections, single sign-on, secure inter-branch payment transactions, cross-site scripting (XSS) vulnerability.

5. TEXT BOOKS

1) William Stallings, *Cryptography and Network Security: Principles and Practice*, 6th Edition, Pearson Education.
2) Atul Kahate, *Cryptography and Network Security*, 3rd Edition, McGraw Hill.

6. REFERENCE BOOKS

1) C. K. Shyamala, N. Harini, T. R. Padmanabhan, *Cryptography and Network Security*, Wiley India.
2) Forouzan and Mukhopadhyay, *Cryptography and Network Security*, McGraw Hill.
3) Mark Stamp, *Information Security: Principles and Practice*, Wiley India.
4) W. Arthur Conklin, Greg White, *Principles of Computer Security*, TMH.
5) Neal Krawetz, *Introduction to Network Security*, Cengage Learning.
6) Bernard Menezes, *Network Security and Cryptography*, Cengage Learning.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**3**	**1**
CO 2	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**3**	**1**
CO 3	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**3**	**1**
CO 4	**3**	**3**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**3**	**1**
CO 5	**3**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**2**	**1**

COURSE CONTENT

DEVOPS								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS503PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. Software Engineering 2. Software Project Management								

1. COURSE OVERVIEW

The *DevOps* course introduces students to modern software development practices that integrate development and operations to enable continuous delivery and deployment. It focuses on Agile methodologies, DevOps culture, automation, continuous integration, continuous testing, and deployment strategies. The course also emphasizes architectural considerations, project management tools, build automation, testing frameworks, and deployment technologies used in real-world DevOps environments.

2. COURSE OBJECTIVE

The students will try to Learn:

1. To understand Agile, DevOps, and related methodologies for achieving continuous delivery.
2. To understand the DevOps lifecycle and its role in business agility.
3. To learn software development models and architectural patterns supporting DevOps.
4. To use version control, project management, integration, and automation tools.
5. To implement automated testing and deployment strategies in DevOps environments.

3. COURSE OUTCOMES

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

CO 1	Explain the fundamentals of DevOps, Agile methodologies, ITIL processes, delivery pipelines, Scrum, Kanban, and continuous delivery practices.
CO 2	Analyze software development models and their influence on DevOps, including architecture considerations for monolithic and microservices systems, database migrations, and resilient design.
CO 3	Apply source code management and project control techniques using Git, Gerrit, GitLab, and containerization tools like Docker for efficient collaboration and version control.
CO 4	Implement continuous integration and build automation using Jenkins, managing build dependencies, pipelines, infrastructure-as-code, and monitoring quality measures.
CO 5	Apply automated testing strategies, test-driven and REPL-driven development, and deploy systems effectively using virtualization and deployment tools.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction to DevOps: Introduction, Agile development model, DevOps and ITIL, DevOps processes and continuous delivery, release management, Scrum, Kanban, delivery pipeline, identifying bottlenecks.

UNIT – II **(10L)**

Software Development Models and DevOps: DevOps lifecycle for business agility, DevOps and continuous testing.

DevOps Influence on Architecture: Introduction to software architecture, monolithic architecture, architecture rules of thumb, separation of concerns, handling database migrations, microservices and data tier, DevOps, architecture, and resilience.

UNIT – III **(10L)**

Project Management and Source Code Control: Need for source code control, history of source code management, roles and code, source code management systems and migrations, shared authentication, hosted Git servers, Git server implementations, Docker intermission, Gerrit, pull request model, GitLab.

UNIT – IV **(9L)**

System Integration: Build systems, Jenkins build server, managing build dependencies, Jenkins plugins and file system layout, host server and build slaves, triggers, job chaining and build pipelines, infrastructure as code, build phases, alternative build servers, collating quality measures.

UNIT – V **(9L)**

Testing Tools and Deployment: Types of testing, automation of testing – pros and cons, Selenium (introduction and features), JavaScript testing, backend integration testing, test-driven development, REPL-driven development.

Deployment of Systems: Deployment systems, virtualization stacks, client-side code execution, Puppet master and agents, Ansible, deployment tools – Chef, SaltStack, Docker.

5. TEXT BOOKS

1) Joakim Verona, *Practical DevOps*, Packt Publishing, 2016.

6. REFERENCE BOOKS

1) Deepak Gaikwad, Viral Thakkar, *DevOps Tools from Practitioner's Viewpoint*, Wiley Publications.
2) Len Bass, Ingo Weber, Liming Zhu, *DevOps: A Software Architect's Perspective*, Addison-Wesley.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**	**1**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**0**
CO 2	**3**	**3**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**0**
CO 3	**3**	**3**	**3**	**2**	**3**	**0**	**0**	**0**	**2**	**0**	**0**	**2**	**3**	**0**
CO 4	**3**	**3**	**3**	**3**	**3**	**0**	**0**	**0**	**2**	**0**	**0**	**2**	**3**	**1**
CO 5	**3**	**3**	**3**	**2**	**3**	**1**	**0**	**1**	**2**	**2**	**0**	**2**	**3**	**1**

COURSE CONTENT

QUANTUM COMPUTING (Professional Elective – I)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS511PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: Nil								

1. COURSE OVERVIEW

The *Quantum Computing* course introduces the fundamental principles of quantum computation and information processing. It explores the mathematical, physical, and biological foundations required to understand quantum systems, qubits, and quantum algorithms. The course covers physical implementations of qubits, quantum circuits, major quantum algorithms, error correction techniques, and the impact of quantum computing on cryptography. By the end of the course, students will gain insight into how quantum computation differs from classical computation and its implications for future computing technologies.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To introduce the fundamentals and evolution of quantum computing.
2) To develop a problem-solving approach using finite-dimensional mathematics.
3) To understand the physical principles underlying quantum computation.
4) To study quantum algorithms and their applications.
5) To analyze the impact of quantum computing on information security and cryptography.

3. COURSE OUTCOMES

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

CO 1	Explain the basic concepts and principles of quantum computing and quantum information.
CO 2	Describe the physical implementation and behavior of qubits.
CO 3	Design and analyze quantum circuits using single and multi-qubit gates.
CO 4	Apply quantum algorithms such as Deutsch–Jozsa, Shor's, and Grover's algorithms.
CO 5	Analyze quantum error correction techniques and the impact of quantum computing on cryptography.

4. COURSE CONTENT

UNIT – I **(10L)**

History of Quantum Computing: Importance of mathematics, physics, and biology.
Introduction to Quantum Computing: Bits vs qubits, classical vs quantum logical operations.

UNIT – II **(10L)**

Background Mathematics: Basics of linear algebra, Hilbert space, probabilities and measurements.

Background Physics: Pauli's exclusion principle, superposition, entanglement and supersymmetry, density operators and correlation, basics of quantum mechanics, measurements in bases other than computational basis.

Background Biology: Basic concepts of genomics and proteomics (Central Dogma).

UNIT – III **(10L)**

Qubit: Physical implementations of qubits, qubit as a quantum unit of information.

Quantum Circuits: Single-qubit gates, multiple-qubit gates, designing quantum circuits, Bell states, Bloch sphere representation.

UNIT – IV **(9L)**

Quantum Algorithms: Classical computation on quantum computers, relationship between quantum and classical complexity classes, Deutsch's algorithm, Deutsch–Jozsa algorithm, Shor's factorization algorithm, Grover's search algorithm.

UNIT – V **(9L)**

Noise and Error Correction: Graph states and codes, quantum error correction, fault-tolerant computation.

Quantum Information and Cryptography: Comparison between classical and quantum information theory, quantum cryptography, quantum teleportation.

5. TEXT BOOKS

1) Nielsen, M. A., *Quantum Computation and Quantum Information*, Cambridge University Press.

6. REFERENCE BOOKS

1) Noson S. Yanofsky and Mirco A. Mannucci, *Quantum Computing for Computer Scientists*.
2) Benenti, G., Casati, G., and Strini, G., *Principles of Quantum Computation and Information*, Vol. I: Basic Concepts, Vol. II: Basic Tools and Special Topics, World Scientific.
3) Pittenger, A. O., *An Introduction to Quantum Computing Algorithms*.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**1**
CO 2	**3**	**3**	**2**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**1**
CO 3	**3**	**3**	**3**	**2**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**1**
CO 4	**3**	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**1**
CO 5	**3**	**3**	**2**	**2**	**3**	**1**	**1**	**2**	**0**	**0**	**0**	**3**	**2**	**1**

COURSE CONTENT

ADVANCED COMPUTER ARCHITECTURE (Professional Elective – I)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS512PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: Computer Organization								

1. COURSE OVERVIEW

The *Advanced Computer Architecture* course provides an in-depth understanding of modern parallel and scalable computer architectures. It focuses on the principles of parallelism, performance optimization, scalable multiprocessor systems, pipelining, superscalar processors, and vector processing techniques. The course emphasizes architectural models, interconnection networks, memory organization, and performance evaluation methods required to design high-performance computing systems. By the end of the course, students will be able to analyze and apply advanced architectural concepts to modern computing platforms.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To impart the concepts and principles of parallel and advanced computer architectures.
2) To understand computational models and parallel processing techniques.
3) To develop design techniques for scalable and multithreaded architectures.
4) To analyze performance metrics and scalability issues in parallel systems.
5) To apply advanced architectural concepts in the design of modern computer systems.

3. COURSE OUTCOMES

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

CO 1	Explain parallel computer models, multiprocessor and multicomputer architectures, program partitioning and scheduling, and conditions of parallelism.
CO 2	Analyze scalable performance, speedup laws, processor technologies, and apply parallel processing concepts to performance evaluation.
CO 3	Explain memory organizations and consistency models, and analyze pipelining techniques including instruction, arithmetic, and superscalar pipelines.
CO 4	Analyze interconnection and synchronization mechanisms in multiprocessor and multicomputer systems, and explain message-passing architectures.
CO 5	Explain vector processing principles, SIMD architectures, and multivector and compound vector processing for high-performance computing.

4. COURSE CONTENT

UNIT – I **(10L)**

Theory of Parallelism: Parallel computer models, state of computing.

Parallel Architectures: Multiprocessors and multicomputers, multivector and SIMD computers, PRAM and VLSI models.

Program Characteristics: Architectural development tracks, program and network properties, conditions of parallelism, program partitioning and scheduling, program flow mechanisms, system interconnect architectures.

UNIT – II **(10L)**

Scalable Performance: Principles of scalable performance, performance metrics and measures.

Parallel Processing Applications: Speedup performance laws, scalability analysis and approaches.

Processor Technologies: Hardware technologies, processes and memory hierarchy, advanced processor technology, superscalar and vector processors.

UNIT – III **(10L)**

Memory Organizations: Shared-memory organizations, sequential and weak consistency models.

Pipelining Techniques: Linear and non-linear pipeline processors, instruction pipeline design, arithmetic pipeline design, superscalar pipeline design.

UNIT – IV **(9L)**

Parallel and Scalable Architectures: Multiprocessors and multicomputers.

Interconnection and Synchronization: Multiprocessor system interconnects, cache coherence and synchronization mechanisms.

Message Passing Systems: Three generations of multicomputers, message-passing mechanisms, multivector and SIMD computers.

UNIT – V **(9L)**

Vector Processing: Vector processing principles, multivector multiprocessors, compound vector processing.

SIMD Architectures: SIMD computer organizations, connection machine CM–5.

5. TEXT BOOKS

1) Kai Hwang, *Advanced Computer Architecture*, 2nd Edition, Tata McGraw Hill Publishers.

6. REFERENCE BOOKS

1) J. L. Hennessy and D. A. Patterson, *Computer Architecture*, 4th Edition, Elsevier.
2) S. G. Shiva, *Advanced Computer Architectures*, Special Indian Edition, CRC, Taylor & Francis.
3) G. Hager and G. Wellein, *Introduction to High Performance Computing for Scientists and Engineers*, CRC Press.
4) D. Sima, T. Fountain, and P. Kacsuk, *Advanced Computer Architecture*, Pearson Education.
5) B. Parhami, *Computer Architecture*, Oxford University Press.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**3**	**2**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**0**
CO 2	**3**	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**0**
CO 3	**3**	**3**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**0**
CO 4	**3**	**3**	**3**	**3**	**2**	**1**	**0**	**0**	**2**	**0**	**0**	**2**	**3**	**1**
CO 5	**3**	**3**	**3**	**2**	**2**	**1**	**0**	**0**	**2**	**0**	**0**	**2**	**3**	**1**

COURSE CONTENT

DATA ANALYTICS (Professional Elective – I)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS513PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on "Database Management Systems". 2. Knowledge of probability and statistics.								

2. COURSE OVERVIEW

The *Data Analytics* course provides a comprehensive understanding of techniques used to collect, manage, analyze, and visualize data for informed decision-making. It introduces statistical analysis, regression models, supervised and unsupervised learning techniques, time-series analysis, and data visualization methods. The course emphasizes real-world applications of analytics in business and engineering domains, focusing on data architecture design, data quality management, and extracting meaningful insights from diverse data sources.

3. COURSE OBJECTIVE

The students will try to Learn:

1) To explore the fundamental concepts of data analytics.
2) To understand the principles and methods of statistical analysis.
3) To discover patterns using supervised and unsupervised learning models and evaluate algorithm performance.
4) To understand various search, modeling, and visualization techniques.
5) To design data architectures and manage data from multiple sources.

4. COURSE OUTCOMES

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

CO 1	Design and manage data architectures by identifying diverse data sources and applying appropriate data quality, preprocessing, and data management techniques for analytics.
CO 2	Apply data analytics tools and business modeling techniques to analyze different types of data and variables, including handling missing data through suitable imputation methods.
CO 3	Develop and evaluate regression and logistic regression models by applying least squares estimation, validating model assumptions, and interpreting results for business decision-making.
CO 4	Construct and analyze segmentation, decision tree, and time series models by selecting suitable supervised or unsupervised learning techniques and assessing model accuracy for prediction.
CO 5	Visualize complex datasets and relationships using appropriate visualization techniques to effectively communicate analytical insights and support data-driven decisions.

5. COURSE CONTENT

UNIT – I **(10L)**

Data Management: Design Data Architecture and manage the data for analysis, understand various sources of Data like Sensors/Signals/GPS etc. Data Management, Data Quality(noise, outliers, missing values, duplicate data) and Data Processing & Processing.

UNIT - II **(10L)**

Data Analytics: Introduction to Analytics, Introduction to Tools and Environment, Application of Modeling in Business, Databases & Types of Data and Variables, Data Modeling Techniques, Missing Imputations etc. Need for Business Modeling.

UNIT - III **(10L)**

Regression – Concepts, Blue property assumptions, Least Square Estimation, Variable Rationalization, and Model Building etc.

Logistic Regression: Model Theory, Model fit Statistics, Model Construction, Analytics applications to various Business Domains etc.

UNIT – IV **(9L)**

Object Segmentation: Regression Vs Segmentation – Supervised and Unsupervised Learning, Tree Building – Regression, Classification, Overfitting, Pruning and Complexity, Multiple Decision Trees etc.

Time Series Methods: Arima, Measures of Forecast Accuracy, STL approach, Extract features from generated model as Height, Average Energy etc and Analyze for prediction.

UNIT – V **(9L)**

Data Visualization: Pixel-Oriented Visualization Techniques, Geometric Projection Visualization Techniques, Icon-Based Visualization Techniques, Hierarchical Visualization Techniques, Visualizing Complex Data and Relations.

6. TEXT BOOKS

1) Student's Handbook for Associate Analytics – II, III.
2) Data Mining Concepts and Techniques, Han, Kamber, 3rd Edition, Morgan Kaufmann Publishers.

7. REFERENCE BOOKS

1) Introduction to Data Mining, Tan, Steinbach and Kumar, Addision Wisley, 2006.
2) Data Mining Analysis and Concepts, M. Zaki and W. Meira
3) Mining of Massive Datasets, Jure Leskovec Stanford Univ. Anand Rajaraman Milliway Labs Jeffrey D Ullman Stanford Univ.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**1**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**1**
CO 2	**3**	**3**	**2**	**2**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**1**
CO 3	**3**	**3**	**3**	**2**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**1**
CO 4	**3**	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**3**	**1**
CO 5	**3**	**2**	**2**	**1**	**2**	**1**	**1**	**2**	**0**	**0**	**0**	**3**	**2**	**1**

COURSE CONTENT

IMAGE PROCESSING (Professional Elective – I)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS514PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		

Prerequisite:

1. Students are expected to have knowledge in linear signals and systems, Fourier Transform, basic linear algebra, basic probability theory and basic programming techniques; knowledge of digital signal processing is desirable.
2. A course on "Computational Mathematics"
3. A course on "Computer Oriented Statistical Methods"

1. COURSE OVERVIEW

The **Image Processing** course provides a strong theoretical and mathematical foundation for understanding and manipulating digital images. The course covers the complete image processing pipeline starting from image acquisition to advanced techniques such as image enhancement, restoration, segmentation, transformation, and compression. Emphasis is placed on two-dimensional signal processing concepts and practical techniques used in engineering, medical imaging, remote sensing, and computer vision applications.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Provide a theoretical and mathematical foundation of fundamental Digital Image Processing concepts.
2) The topics include image acquisition; sampling and quantization; preprocessing; enhancement; restoration; segmentation; and compression.

3. COURSE OUTCOMES

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

CO 1	Explain the fundamentals of digital images, including sampling, quantization, pixel relationships, imaging geometry, and 2D transformations such as DFT, DCT, KLT, and SVD.
CO 2	Apply spatial and frequency domain techniques for image enhancement, including point processing, histogram processing, and filtering methods for smoothing and sharpening.
CO 3	Analyze and implement image restoration techniques using degradation models, inverse filtering, least mean square filters, and constrained least squares approaches.
CO 4	Apply image segmentation methods including edge detection, edge linking, thresholding, and region-based segmentation for effective object extraction.
CO 5	Explain image compression concepts, including redundancy removal, fidelity criteria, source coding models, error-free and lossy compression techniques.

4. COURSE CONTENT

UNIT – I **(10L)**

Digital Image Fundamentals: Digital Image through Scanner, Digital Camera. Concept of Gray Levels. Gray Level to Binary Image Conversion. Sampling and Quantization. Relationship between Pixels. Imaging Geometry. 2D Transformations-DFT, DCT, KLT and SVD.

UNIT - II **(9L)**

Image Enhancement in Spatial Domain Point Processing, Histogram Processing, Spatial Filtering, Enhancement in Frequency Domain, Image Smoothing, Image Sharpening.

UNIT - III **(10L)**

Image Restoration Degradation Model, Algebraic Approach to Restoration, Inverse Filtering, Least Mean Square Filters, Constrained Least Squares Restoration, Interactive Restoration.

UNIT - IV **(10L)**

Image Segmentation Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region Oriented Segmentation.

UNIT - V **(9L)**

Image Compression Redundancies and their Removal Methods, Fidelity Criteria, Image Compression Models, Source Encoder and Decoder, Error Free Compression, Lossy Compression.

5. TEXT BOOKS

1) Digital Image Processing: R.C. Gonzalez & R. E. Woods, Addison Wesley/ Pearson Education, 2nd Ed, 2004.

6. REFERENCE BOOKS

1) Fundamentals of Digital Image Processing: A. K. Jain, PHI.
2) Digital Image Processing using MAT LAB: Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins: Pearson Education India, 2004.
3) Digital Image Processing: William K. Pratt, John Wiley, 3rd Edition, 2004.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**3**	**2**	**2**	**3**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**–**
CO 2	**3**	**3**	**3**	**2**	**3**	**–**	**–**	**–**	**2**	**–**	**–**	**2**	**3**	**–**
CO 3	**3**	**3**	**3**	**3**	**2**	**–**	**–**	**–**	**2**	**–**	**–**	**2**	**3**	**1**
CO 4	**3**	**3**	**3**	**3**	**2**	**1**	**–**	**–**	**2**	**–**	**–**	**2**	**3**	**1**
CO 5	**3**	**3**	**3**	**2**	**3**	**1**	**–**	**1**	**2**	**2**	**–**	**2**	**3**	**1**

COURSE CONTENT

PRINCIPLES OF PROGRAMMING LANGUAGES (Professional Elective – I)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS515PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on "Mathematical Foundations of Computer Science". 2. A course on "Computer Programming and Data Structures".								

1. COURSE OVERVIEW

The *Principles of Programming Languages* course provides a comprehensive understanding of the fundamental concepts, design principles, and paradigms underlying modern programming languages. The course explores syntax and semantics, data types, control structures, abstraction mechanisms, programming paradigms, and language implementation issues. By studying different language features and paradigms, students gain the ability to analyze, compare, and design programming languages and apply appropriate programming concepts in real-world software development.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the fundamental concepts and principles of programming languages.
2) To study syntax, semantics, and language design issues.
3) To explore various programming paradigms such as procedural, object-oriented, functional, and logic programming.
4) To analyze data types, control structures, and abstraction mechanisms.
5) To understand runtime environments, memory management, and language implementation concepts.

3. COURSE OUTCOMES

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

CO 1	Explain the fundamental concepts, syntax, and semantics of programming languages.
CO 2	Analyze and compare different programming paradigms and language features.
CO 3	Apply data types, control structures, and abstraction mechanisms effectively.
CO 4	Understand language implementation issues including memory management and runtime environments.
CO 5	Evaluate programming languages based on design principles and application requirements.

4. COURSE CONTENT

UNIT - I

Preliminary Concepts: **(10L)**

Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments Syntax **and Semantics:** General Problem of Describing Syntax and Semantics, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meanings of Programs

UNIT - II **(10L)**

Names, Bindings, and Scopes:

Introduction, Names, Variables, Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants.

Data Types:
Ordinal Types, Array, Associative Arrays, Record, Union, Tuple Types, List Types, Pointer and Reference Types, Type Checking, Strong Typing, Type Equivalence Expressions and Statements, Arithmetic Expressions, Overloaded Operators, Type Introduction, Primitive Data Types, Character String Types, User Defined Conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment Control Structures – Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

UNIT - III **(10L)**

Subprograms and Blocks:
Fundamentals of Sub-Programs, Design Issues for Subprograms, Local Referencing Environments, Parameter Passing Methods, Parameters that Are Subprograms, Calling Subprograms Indirectly, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User Defined Overloaded Operators, Closures, Coroutines Implementing Subprograms:
General Semantics of Calls and Returns, Implementing Simple Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping.

Abstract Data Types:
The Concept of Abstraction, Introductions to Data Abstraction, Design Issues, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulations

UNIT - IV **(9L)**

Concurrency:
Introduction, Introduction to Subprogram Level Concurrency, Semaphores, Monitors, Message Passing, Java Threads, Concurrency in Function Languages, Statement Level Concurrency. Exception Handling and Event Handling: Introduction, Exception Handling in Ada, C++, Java, Introduction to Event Handling, Event Handling with Java and C#.

UNIT - V **(9L)**

Functional Programming Languages:
Introduction, Mathematical Functions, Fundamentals of Functional Programming Language, LISP, Support for Functional Programming in Primarily Imperative Languages, Comparison of Functional and Imperative Languages.

Logic Programming Language:
Introduction, an Overview of Logic Programming, Basic Elements of Prolog, Applications of Logic Programming.

Scripting Language:
Pragmatics, Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library. (Text Book 2).

5. TEXT BOOKS

1) Concepts of Programming Languages Robert. W. Sebesta 10/E, Pearson Education.
2) Programming Language Design Concepts, D. A. Watt, Wiley Dreamtech, 2007.

6. REFERENCE BOOKS

1) Programming Languages, 2nd Edition, A.B. Tucker, R. E. Noonan, TMH.
2) Programming Languages, K. C. Louden, 2nd Edition, Thomson, 2003.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**1**									**2**	**3**	**1**
CO 2	**3**	**3**	**2**	**1**	**1**							**2**	**3**	**1**
CO 3	**3**	**3**	**3**	**2**	**2**							**2**	**3**	**1**
CO 4	**3**	**3**	**2**	**2**	**2**							**2**	**3**	**1**
CO 5	**3**	**2**	**2**	**2**	**3**	**1**	**1**	**2**				**3**	**2**	**1**

COURSE CONTENT

COMPUTER GRAPHICS (Professional Elective – II)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS521PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on "Mathematical Foundations of Computer Science". 2. A course on "Computer Programming and Data Structures".								

1. COURSE OVERVIEW

The **Computer Graphics** course introduces the fundamental concepts and techniques used in the creation, manipulation, and rendering of graphical images. The course covers the basics of graphics systems, geometric modeling, transformations, viewing, rendering, animation, and visible surface detection. Emphasis is placed on both theoretical foundations and practical algorithms used in modern graphics applications.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the fundamentals of computer graphics systems.
2) To study points, lines, and line drawing algorithms.
3) To understand two-dimensional and three-dimensional geometric transformations.
4) To analyze clipping techniques and viewing transformations.
5) To introduce rendering, animation, and visible surface detection techniques

3. COURSE OUTCOMES

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

CO 1	Explain the architecture of computer graphics systems, display devices, and input/output primitives, and implement fundamental line, circle, ellipse, and polygon filling algorithms.
CO 2	Apply 2-D geometric transformations and viewing operations using matrix and homogeneous coordinate representations, including clipping algorithms for graphical objects.
CO 3	Model and represent 3-D objects using polygonal, quadric, and spline-based techniques, and apply appropriate color models and rendering methods.
CO 4	Perform 3-D geometric transformations and viewing operations by applying projection techniques, view volumes, and clipping for realistic scene representation.
CO 5	Design basic computer animations and apply visible surface detection algorithms to determine object visibility in complex graphical scenes.

4. COURSE CONTENT

UNIT - I **(10L)**

Introduction: Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random-scan systems, graphics monitors and work stations and input devices

Output primitives: Points and lines, line drawing algorithms (DDA and Bresenham's Algorithm) circle- generating algorithms and ellipse - generating algorithms

Polygon Filling: Scan-line algorithm, boundary-fill and flood-fill algorithms

UNIT - II **(10L)**

2-D geometric transformations: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, clipping operations, point clipping, Line clipping-Cohen Sutherland algorithms, Polygon clipping-Sutherland Hodgeman polygon clipping algorithm.

UNIT - III **(10L)**

3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces, Polygon rendering methods, color models and color applications.

UNIT - IV **(9L)**

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D viewing: Viewing pipeline, viewing coordinates, projections, view volume and general projection transforms and clipping.

UNIT - V **(9L)**

Computer animation: Design of animation sequence, general computer animation functions, raster animations, computer animation languages, key frame systems, motion specifications.

Visible surface detection methods: Classification, back-face detection, depth-buffer method, BSP-tree method, area sub-division method and octree method.

5. TEXT BOOKS

1) "Computer Graphics C version", Donald Hearn and M. Pauline Baker, Pearson Education.

6. REFERENCE BOOKS

1) Procedural elements for Computer Graphics, David F Rogers, Tata Mc Graw hill, 2nd edition.
2) Principles of Interactive Computer Graphics", Neuman and Sproul, TMH.
3) Principles of Computer Graphics, Shalini Govil, Pai, 2005, Springer.
4) "Computer Graphics Principles & practice", second edition in C, Foley, Van Dam, Feiner and Hughes, Pearson Education.
5) Computer Graphics, Steven Harrington, TMH.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	–	–	–	–	–	–	–	–	–	**2**	**3**	**1**
CO 2	**3**	**3**	**2**	–	–	–	–	–	–	–	–	**2**	**3**	**1**
CO 3	**3**	**3**	**2**	**2**	–	–	–	–	–	–	–	**2**	**3**	**1**
CO 4	**3**	**3**	–	**2**	–	–	–	–	–	–	–	**2**	**3**	**1**
CO 5	**3**	**3**	**3**	**2**	**3**	–	–	–	–	–	–	**3**	**2**	**1**

COURSE CONTENT

EMBEDDED SYSTEMS (Professional Elective – II)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS522PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on "Digital Logic Design and Microprocessors" 2. A course on "Computer Organization and Architecture"								

1. COURSE OVERVIEW

The **Embedded Systems** course introduces the design and development of dedicated computer systems built to perform specific functions within larger systems. The course covers the fundamentals of embedded hardware and software, including microcontrollers, processors, interfacing techniques, real-time constraints, and embedded programming. Students gain an understanding of how embedded systems are designed, programmed, and integrated into real-world applications such as consumer electronics, automotive systems, industrial automation, and IoT devices.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To provide an overview of principles of Embedded System.
2) To provide a clear understanding of role of firmware, operating systems in correlation with hardware systems.

3. COURSE OUTCOMES

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

CO 1	Explain the fundamentals, classification, design process, and quality attributes of embedded systems, including embedded hardware and software components.
CO 2	Describe processor and microcontroller architectures, memory organization, and interrupt handling mechanisms, and apply them to real-world interfacing scenarios.
CO 3	Analyze on-board communication mechanisms by comparing serial, parallel, and wireless communication devices and implementing standard bus protocols such as I2C and SPI.
CO 4	Develop embedded firmware using assembly language and C by applying programming constructs, data types, data structures, and processor directives.
CO 5	Explain the role of operating systems in embedded systems by applying OS services, task management, inter-process communication, and synchronization mechanisms.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction to Embedded Systems: Processor embedded into a system, Embedded Hardware units and devices in a system, Embedded software in a system, Design process of an embedded system, classification of embedded systems, characteristics and quality attributes of an embedded systems.

UNIT - II **(10L)**

Introduction to processor/microcontroller architecture, Real world interfacing, processor and memory organization, memory types, memory maps and addresses, interrupt sources and interrupt service mechanism.

UNIT - III **(10L)**

On board Communication Basics: serial; communication devices, Parallel devices, Wireless devices, Real time clock, Serial bus communication Protocols - I2C, SPI; Parallel buss communication - ISA, PCI.

UNIT - IV **(9L)**

Embedded Firmware Development: Overview of programming concepts - in assembly language and in high level language 'C', C Program elements- Heads, Source files, Processor Directives, Macros, Functions, Data types and Data Structures

UNIT - V **(9L)**

OS Based Embedded Systems: OS services - Process/Task Management, Memory Management, I/O subsystem manager, Inter Process/Task communications - Tasks, Task states, Shared data, Signals, Message Queues, Mailbox, Pipes and concepts of Semaphores.

5. TEXT BOOKS

1) Embedded Systems, Raj Kamal, 2nd edition, Tata Mc Graw Hill.
2) Shibu K V, "Introduction to Embedded Systems", Second Edition, Mc Graw Hill.

6. REFERENCE BOOKS

1) Rajkamal, Embedded Systems Architecture, Programming and Design, Tata McGraw-Hill.
2) Frank Vahid and Tony Givargis, "Embedded Systems Design" - A Unified Hardware/Software Introduction, John Wiley
3) Lyla, "Embedded Systems" –Pearson
4) David E. Simon, An Embedded Software Primer, Pearson Education Asia, First Indian Reprint 2000.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**1**	**1**	**–**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 2	**3**	**3**	**2**	**1**	**1**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 3	**3**	**2**	**2**	**1**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 4	**3**	**2**	**3**	**2**	**3**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 5	**3**	**3**	**2**	**2**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**3**	**2**	**1**

COURSE CONTENT

INFORMATION RETRIEVAL SYSTEMS (Professional Elective – II)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
CS523PE	**Elective**	**3**	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. Data Structures								

1. COURSE OVERVIEW

This course introduces the principles and techniques used to search, retrieve, and rank relevant information from large data collections. It covers indexing, query processing, retrieval models, and evaluation methods, with applications in search engines and digital libraries.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To learn the concepts and algorithms in Information Retrieval Systems.
2) To understand the data/file structures that are necessary to design, and implement information retrieval (IR) systems.

3. COURSE OUTCOMES

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

CO 1	Apply Information Retrieval (IR) principles to locate and retrieve relevant information from large data collections.
CO 2	Design and analyze different document clustering and indexing techniques for efficient information organization.
CO 3	Implement retrieval models and similarity-based ranking mechanisms for web search tasks.
CO 4	Design and develop a complete Information Retrieval System for web-based search applications.
CO 5	Analyze and apply text and multimedia information retrieval techniques for handling diverse data formats such as text, audio, image, and video. Bottom of Form

4. COURSE CONTENT

UNIT - I **(10L)**

Introduction to Information Retrieval Systems: Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses Information Retrieval System Capabilities: Search Capabilities, Browse Capabilities, Miscellaneous Capabilities.

UNIT - II **(10L)**

Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models.

UNIT - III **(10L)**

Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages

Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters

UNIT - IV **(9L)**

User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems, Searching the INTERNET and Hypertext

Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies

UNIT - V **(9L)**

Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems

Multimedia Information Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval

5. TEXT BOOKS

1) Information Storage and Retrieval Systems – Theory and Implementation, Second Edition, Gerald J. Kowalski, Mark T. Maybury, Springer.

6. REFERENCE BOOKS

1) Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
2) Information Storage & Retrieval by Robert Korfhage – John Wiley & Sons.
3) Modern Information Retrieval by Yates and Neto Pearson Education.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**	**1**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 2	**3**	**3**	**3**	**2**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 3	**3**	**3**	**3**	**2**	**3**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 4	**3**	**3**	**3**	**3**	**3**	**–**	**–**	**–**	**2**	**1**	**1**	**2**	**3**	**1**
CO 5	**3**	**3**	**2**	**2**	**3**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**2**	**1**

COURSE CONTENT

DISTRIBUTED DATABASES (Professional Elective – II)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS524PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on "Database Management Systems"								

1. COURSE OVERVIEW

The Distributed Databases course introduces the concepts, architectures, and design principles of databases that are distributed across multiple networked sites. The course covers data distribution strategies, fragmentation, replication, distributed query processing, transaction management, and concurrency control. Students gain an understanding of reliability, fault tolerance, and consistency issues in distributed environments, enabling them to design and manage scalable and efficient distributed database systems for modern applications.

2. COURSE OBJECTIVE

The students will try to Learn:

1) The purpose of the course is to enrich the previous knowledge of database systems and expose the need for distributed database technology to confront the deficiencies of the centralized database systems.
2) Introduce basic principles and implementation techniques of distributed database systems.
3) Equip students with principles and knowledge of parallel and object-oriented databases.
4) Topics include distributed DBMS architecture and design; query processing and optimization; distributed transaction management and reliability; parallel and object database management systems.

3. COURSE OUTCOMES

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

CO 1	Explain the fundamentals of distributed databases, architecture models of DDBMS, and design issues including fragmentation and data allocation strategies.
CO 2	Analyze query processing and decomposition techniques, and apply distributed query optimization algorithms for efficient data retrieval.
CO 3	Apply transaction management concepts in distributed environments, including concurrency control, serializability, deadlock handling, and transaction types.
CO 4	Analyze reliability and fault-tolerance in distributed DBMS, and evaluate parallel database architectures, data placement, and load balancing techniques.
CO 5	Explain object-oriented distributed database systems, including object modeling, distributed object management, object query processing, and comparison with ORDBMS.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction: Distributed Data Processing, Distributed Database System, Promises of DDBSs, Problem areas.

Distributed DBMS Architecture: Architectural Models for Distributed DBMS, DDMBS Architecture.

Distributed Database Design: Alternative Design Strategies, Distribution Design issues, Fragmentation, Allocation.

UNIT - II **(10L)**

Query processing and decomposition: Query processing objectives, characterization of query processors, layers of query processing, query decomposition, localization of distributed data.

Distributed query Optimization: Query optimization, centralized query optimization, distributed query optimization algorithms.

UNIT - III **(10L)**

Transaction Management: Definition, properties of transaction, types of transactions, distributed concurrency control: serializability, concurrency control mechanisms & algorithms, time - stamped & optimistic concurrency control Algorithms, deadlock Management.

UNIT - IV **(9L)**

Distributed DBMS Reliability: Reliability concepts and measures, fault-tolerance in distributed systems, failures in Distributed DBMS, local & distributed reliability protocols, site failures and network partitioning.

Parallel Database Systems: Parallel database system architectures, parallel data placement, parallel query processing, load balancing, database clusters.

UNIT - V **(9L)**

Distributed object Database Management Systems: Fundamental object concepts and models, object distributed design, architectural issues, object management, distributed object storage, object query Processing.

Object Oriented Data Model: Inheritance, object identity, persistent programming languages, persistence of objects, comparison OODBMS and ORDBMS

5. TEXT BOOKS

1) M. Tamer OZSU and Patuck Valduriez: Principles of Distributed Database Systems, Pearson Edn. Asia, 2001.
2) Stefano Ceri and Giuseppe Pelagatti: Distributed Databases, McGraw Hill.

6. REFERENCE BOOKS

1) Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: "Database Systems: The Complete Book", Second Edition, Pearson International Edition.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**3**	**3**	**2**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**–**
CO 2	**3**	**3**	**3**	**3**	**3**	**–**	**–**	**–**	**2**	**–**	**–**	**2**	**3**	**–**
CO 3	**3**	**3**	**3**	**3**	**2**	**–**	**–**	**–**	**2**	**–**	**–**	**2**	**3**	**1**
CO 4	**3**	**3**	**3**	**3**	**2**	**1**	**–**	**1**	**2**	**–**	**–**	**2**	**3**	**1**
CO 5	**3**	**3**	**3**	**2**	**2**	**1**	**–**	**1**	**2**	**–**	**–**	**2**	**3**	**1**

COURSE CONTENT

NATURAL LANGUAGE PROCESSING (Professional Elective – II)								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS525PE	**Elective**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. Data structures and compiler design								

1. COURSE OVERVIEW

The Natural Language Processing (NLP) course introduces the principles and techniques that enable computers to understand, interpret, and generate human language. The course covers text processing, syntactic and semantic analysis, language modeling, machine learning and deep learning approaches for NLP, and applications such as information extraction, sentiment analysis, machine translation, and question answering. Students will gain the skills needed to design and implement NLP solutions for real-world.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Introduction to some of the problems and solutions of NLP and their relation to linguistics and statistics.

3. COURSE OUTCOMES

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

CO 1	Understand and model the internal structure of words and documents using morphological analysis and document structure identification techniques.
CO 2	Apply syntactic parsing techniques and parsing algorithms to analyze and represent the grammatical structure of natural language sentences.
CO 3	Analyze ambiguity in natural language and implement semantic parsing techniques to resolve word sense and syntactic ambiguities in multilingual contexts.
CO 4	Design and implement semantic representation models using predicate–argument structures and formal meaning representation systems.
CO 5	Construct, evaluate, and adapt statistical language models using n-grams, Bayesian estimation, and multilingual language modeling techniques.

4. COURSE CONTENT

UNIT – I **(10L)**

Finding the Structure of Words: Words and Their Components, Issues and Challenges, Morphological Models

Finding the Structure of Documents: Introduction, Methods, Complexity of the Approaches, Performances of the Approaches, Features

UNIT - II **(10L)**

Syntax I: Parsing Natural Language, Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms

UNIT – III **(10L)**

Syntax II: Models for Ambiguity Resolution in Parsing, Multilingual Issues

Semantic Parsing I: Introduction, Semantic Interpretation, System Paradigms, Word Sense

UNIT - IV **(9L)**

Semantic Parsing II: Predicate-Argument Structure, Meaning Representation Systems

UNIT - V **(9L)**

Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Bayesian parameter estimation, Language Model Adaptation, Language Models- class based, variable length, Bayesian topic based, Multilingual and Cross Lingual Language Modeling.

5. TEXT BOOKS

1) Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication.

6. REFERENCE BOOKS

1) Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications.
2) Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**–**	**–**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 2	**3**	**3**	**2**	**2**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 3	**3**	**3**	**2**	**2**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 4	**3**	**3**	**3**	**2**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 5	**3**	**3**	**3**	**3**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**3**	**2**	**1**

COURSE CONTENT

DEVOPS LAB								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS504PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

The DevOps Laboratory provides hands-on experience in implementing DevOps practices and tools used for modern software development and deployment. The course focuses on continuous integration and continuous delivery (CI/CD), version control, build automation, configuration management, containerization, and cloud-based deployment. Through practical exercises, students learn to collaborate effectively, automate software workflows, monitor applications, and manage scalable infrastructure. By the end of the lab, students will be able to apply DevOps tools and methodologies to build, test, deploy, and maintain reliable software systems.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Develop a sustainable infrastructure for applications and ensure high scalability. DevOps aims to shorten the software development lifecycle to provide continuous delivery with high-quality.

3. COURSE OUTCOMES

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

CO 1	Understand the need for DevOps tools and practices in modern softwares Develop basic web applications and manage source code using Git and GitHub.
CO 2	Apply continuous integration concepts using Jenkins for automated build and deployment.
CO 3	Build and manage containerized applications using Docker.
CO 4	Deploy and automate containerized applications using Kubernetes.
CO 5	Perform automated testing of web and containerized applications using Selenium.

4. LIST OF EXPERIMENTS:

1. Write code for a simple user registration form for an event.
2. Explore Git and GitHub commands.
3. Practice Source code management on GitHub. Experiment with the source code in exercise 1.
4. Jenkins installation and setup, explore the environment.
5. Demonstrate continuous integration and development using Jenkins.
6. Explore Docker commands for content management.
7. Develop a simple containerized application using Docker.
8. Integrate Kubernetes and Docker
9. Automate the process of running containerized application for exercise 7 using Kubernetes.
10. Install and Explore Selenium for automated testing.
11. Write a simple program in JavaScript and perform testing using Selenium.
12. Develop test cases for the above containerized application using selenium.

5. TEXT BOOKS

1) Joakim Verona., Practical DevOps, Packt Publishing, 2016.

6. REFERENCE BOOKS

1) Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications.
2) Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**–**	**–**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 2	**3**	**3**	**2**	**–**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 3	**3**	**3**	**3**	**2**	**3**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 4	**2**	**3**	**2**	**3**	**2**	**–**	**–**	**–**	**–**	**–**	**–**	**2**	**3**	**1**
CO 5	**3**	**3**	**3**	**3**	**3**	**–**	**–**	**–**	**–**	**–**	**–**	**3**	**2**	**1**

COURSE CONTENT

CRYPTOGRAPHY AND NETWORK SECURITY LAB								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS505PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

The Cryptography and Network Security Laboratory provides practical exposure to the fundamental techniques used to secure data and communication networks. The course focuses on implementing classical and modern cryptographic algorithms, secure key management techniques, authentication mechanisms, and network security protocols. Through hands-on experiments, students learn to analyze security threats, implement encryption and decryption methods, and evaluate the effectiveness of security mechanisms in protecting information systems and networked applications.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Explain the objectives of information security
2) Explain the importance and application of each of confidentiality, integrity, authentication and availability
3) Understand various cryptographic algorithms.

3. COURSE OUTCOMES

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

CO 1	Understand basic cryptographic algorithms, message authentication, web authentication mechanisms, and common security issues.
CO 2	Identify information system security requirements for both client-side and server-side environments.
CO 3	Understand current legal, ethical, and regulatory issues related to information and network security.
CO 4	Implement symmetric and asymmetric cryptographic algorithms using programming languages such as C and Java.
CO 5	Apply cryptographic hash functions and key exchange mechanisms to ensure data integrity, confidentiality, and secure communication.

4. COURSE CONTENT

LIST OF EXPERIMENTS

1. Write a C program that contains a string (char pointer) with a value ‘Hello world’. The program should XOR each character in this string with 0 and display the result.
2. Write a C program that contains a string (char pointer) with a value ‘Hello world’. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a Java program to perform encryption and decryption using the following algorithms
   a. Ceaser cipher b. Substitution cipher c. Hill Cipher

4. Write a C/JAVA program to implement the DES algorithm logic.
5. Write a C/JAVA program to implement the Blowfish algorithm logic.
6. Write a C/JAVA program to implement the Rijndael algorithm logic.
7. Write the RC4 logic in Java Using Java cryptography; encrypt the text "Hello world" using Blowfish. Create your own key using Java key tool.
8. Write a Java program to implement the RSA algorithm.
9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
11. Calculate the message digest of a text using the MD5 algorithm in JAVA.

5. TEXT BOOK

1) Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition
2) Cryptography and Network Security: Atul Kahate, McGraw Hill, 3rd Edition

6. REFERENCE BOOKS

1) Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition.
2) Cryptography and Network Security: Forouzan Mukhopadhyay, McGraw Hill, 3rd Edition
3) Information Security, Principles, and Practice: Mark Stamp, Wiley India.
4) Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH
5) Introduction to Network Security: Neal Krawetz, CENGAGE Learning
6) Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	–	–	**2**	–	–	**2**	–	–	–	**2**	**3**	**1**
CO 2	**3**	**3**	**2**	**2**	**2**	–	–	–	–	–	–	**2**	**3**	**1**
CO 3	**2**	**2**	–	–	–	**2**	–	**3**	–	–	–	**2**	**3**	**1**
CO 4	**3**	**3**	**3**	**2**	**3**	–	–	–	–	–	–	**2**	**3**	**1**
CO 5	**3**	**3**	**2**	**3**	**3**	–	–	–	–	–	–	**3**	**2**	**1**

COURSE CONTENT

ADVANCED ENGLISH COMMUNICATION SKILLS LAB								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
EN506HS	**Foundation**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Computer with Headphones, OALD & Interactive Communication Skills Lab								

1. COURSE OVERVIEW

The introduction of the Advanced English Communication Skills Lab is considered essential at the B. Tech 3rd year level. At this stage, the students need to prepare themselves for their career which may require them to listen to, read, speak and write in English both for their professional & interpersonal communication in the globalised context.

The proposed course should be a laboratory course to enable students to use appropriate English and perform the following:

1. Gathering ideas and information to organise ideas relevantly and coherently.
2. Making oral presentations.
3. Writing formal letters.
4. Transferring information from non-verbal to verbal texts and vice-versa.
5. Writing project/research reports/technical reports.
6. Participating in group discussions.
7. Engaging in debates.
8. Facing interviews.
9. Taking part in social and professional communication.

2. COURSE OBJECTIVE

This Lab focuses on using multi-media instruction for language development to meet the following targets:

1. To improve the students' fluency in English, with a focus on vocabulary
2. To enable them to listen to English spoken at normal conversational speed by educated English speakers
3. To respond appropriately in different socio-cultural and professional contexts
4. To communicate their ideas relevantly and coherently in writing
5. To prepare the students for placements.

3. COURSE OUTCOMES

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

CO 1	Demonstrate the fundamentals of writing, Grammar & Vocabulary.
CO 2	Develop different types of writing.
CO 3	Analyse writing tasks and adapt style, tone, and format
CO 4	Produce academic & Professional documents like Letters, Reports, emails etc.
CO 5	Use the language in real life situations

Syllabus:

The following course content to conduct the activities is prescribed for the Advanced English Communication Skills (AECS) Lab:

1. **Activities on Listening and Reading Comprehension**

2. **For Practice:** Active Listening – Development of Listening Skills Through Audio clips – Benefits of Reading – Methods and Techniques of Reading.

   Basic steps to Effective Reading – Common Obstacles – Discourse Markers or Linkers – Sub- skills of reading - Reading for facts, negative facts and Specific Details – Guessing Meanings from Context, inferring meaning – Critical Reading – Reading Comprehension – Exercises.

3. **Activities on Writing Skills:** Vocabulary for Competitive Examinations – Planning for Writing – Improving Writing Skills- Structure and presentation of different types of writing – Structured writing – Letter writing – Writing a letter of Application – Resume vs. Curriculum Vitae – Writing a Resume – Styles of Resume – e-Correspondence – Emails – Blog Writing –(N) etiquette – Report Writing – Importance of Reports – Types of Reports – Technical Report Writing – Exercises for practice.

4. **Activities on Presentation Skills:** Starting a conversation – responding appropriately and relevantly – Using the right language and body language – Role Play in different situations including Seeking Clarifications, Making a Request, Asking for and Refusing Permission, Participating in a Small Talk – Oral Presentations (individual and group) through JAM sessions – PPT's – Importance of Presentation Skills – Planning, Preparing, Rehearsing and Making a Presentation – Dealing with Glossophobia or Stage Fear – Understanding Nuances of Delivery – Presentations through Posters/Projects/ Reports – Checklist for making a Presentation and Rubrics of Evaluation.

5. **Activities on Group Discussion (GD):** Types of GD and GD as a part of a Selection Procedure – Dynamics of Group Discussion – Myths of GD – Intervention, Summarizing – Modulation of voice, Body Language, Relevance, Fluency and Organization of ideas – Do's and Don'ts – GD Strategies – Exercises for Practice.

6. **Interview Skills:** Concept and Process – Interview Preparation Techniques – Types of Interview Questions – Pre-interview Planning, Opening Strategies, Answering Strategies – Interview Through Tele -conference & Video-conference – Mock Interviews.

5. MINIMUM REQUIREMENT:

The Advanced English Communication Skills (AECS) Laboratory shall have the following infrastructural facilities to accommodate at least 35 students in the lab:

- Spacious room with appropriate acoustics
- Round Tables with movable chairs
- Audio – visual aids
- LCD Projector
- Public Address System
- One PC with latest configuration for the teacher
- T.V. a digital stereo & Camcorder
- Headphones of High quality

6. SUGGESTED SOFTWARE:

The software consisting of the prescribed topics elaborated above should be procured and used.

- TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
- Oxford Advanced Learner's Dictionary, 10th Edition.
- Cambridge Advanced Learner's Dictionary
- DELTA'S key to the next Generation TOEFL Test: Advanced Skill Practice.
- Lingua TOEFL CBT Insider, by Dreamtech

7. BOOKS RECOMMENDED:

1. Rizvi, M. Ashraf (2018). Effective Technical Communication. (2nd ed) McGraw Hill Education (India) Pvt. Ltd.
2. Suresh Kumar, E. (2015). Engineering English. Orient Black Swan Pvt. Ltd.
3. Bailey, Stephen. (2018) Academic Writing: A Handbook for International Students. (5th ed). Routledge.
4. Koneru, Aruna. (2016). Professional Communication. McGraw Hill Education (India) Pvt. Ltd.
5. Raman, Meenakshi & Sharma, Sangeeta. (2022). Technical Communication, Principles and Practice. (4th ed) Oxford University Press.
6. Anderson, Paul V. (2007) Technical Communication. Cengage Learning Pvt. Ltd. New Delhi.
7. Mc Carthy, Michael; O'Dell, Felicity & Redman, Stuart. (2017). English Vocabulary in use series. Cambridge University Press.
8. Sen, Leela. (2009). Communication Skills. PHI Learning Pvt. Ltd., New Delhi.
9. Elbow, Peter. (1998). Writing with Power. Oxford University Press.
10. Goleman, Daniel. (2013). Emotional Intelligence: Why it can matter more than IQ. Bloomsbury Publishing.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1									**2**	**3**		**3**		**2**
CO 2									**2**	**3**		**3**		**2**
CO 3									**2**	**3**		**3**		**2**
CO 4									**2**	**3**		**3**		**2**
CO 5									**2**	**3**		**3**		**2**

COURSE CONTENT

UI DESIGN-FLUTTER								
III Year - I Semester: CSE								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS507PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course introduces Flutter and Dart for cross-platform mobile application development. It covers Dart language fundamentals, Flutter widgets, layouts, and responsive UI design. Students learn navigation, state management, form handling, animations, and REST API integration. The course also emphasizes testing, debugging, and best practices for building robust and scalable Flutter applications.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Learns to Implement Flutter Widgets and Layouts
2) Understands Responsive UI Design and with Navigation in Flutter
3) Knowledge on Widges and customize widgets for specific UI elements, Themes
4) Understand to include animation apart from fetching data

3. COURSE OUTCOMES

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

CO 1	Understand and apply Dart programming fundamentals and set up the Flutter development environment for building mobile applications.
CO 2	Design and implement user interfaces using Flutter widgets, layouts, custom widgets, and theming to create interactive applications.
CO 3	Develop responsive and adaptive Flutter applications that work seamlessly across different screen sizes using media queries and layout techniques.
CO 4	Implement navigation, state management, and form handling in Flutter applications using Navigator, routes, stateful/stateless widgets, and validation techniques.
CO 5	Integrate advanced features such as animations, REST API data fetching, debugging, and testing to create robust and user-friendly Flutter applications.

4. LIST OF EXPERIMENTS

1. a) Install Flutter and Dart SDK.

b) Write a simple Dart program to understand the language basics.

2. a) Explore various Flutter widgets (Text, Image, Container, etc.).

b) Implement different layout structures using Row, Column, and Stack widgets.

3. a) Design a responsive UI that adapts to different screen sizes.

b) Implement media queries and breakpoints for responsiveness.

4. a) Set up navigation between different screens using Navigator.

b) Implement navigation with named routes.

5. a) Learn about stateful and stateless widgets.

b) Implement state management using set State and Provider.

6. a) Create custom widgets for specific UI elements.

b) Apply styling using themes and custom styles.

7. a) Design a form with various input fields.

b) Implement form validation and error handling.

8. a) Add animations to UI elements using Flutter's animation framework.

b) Experiment with different types of animations (fade, slide, etc.).

9. a) Fetch data from a REST API.

b) Display the fetched data in a meaningful way in the UI.

10. a) Write unit tests for UI components.

b) Use Flutter's debugging tools to identify and fix issues.

5. TEXT BOOKS

1) Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**		**3**							**2**	**3**	**2**
CO 2	**3**	**2**	**3**		**3**				**1**			**2**	**3**	**2**
CO 3	**3**	**2**	**3**	**1**	**3**				**1**			**2**	**3**	**2**
CO 4	**2**	**3**	**3**		**3**				**1**			**2**	**2**	**3**
CO 5	**3**	**3**	**3**		**3**				**1**			**2**	**3**	**3**