SRI INDU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)

B.Tech. in COMPUTER SCIENCE AND ENGINEERING
COURSE STRUCTURE, III YEAR & IV YEAR SYLLABUS

(BR22 Regulations)

Applicable from Academic Year: 2022-23 BATCH

III Year I Semester

S. No.	Course Code	Course Title	L	T	P	Credits
1	CS501PC	Design and Analysis of Algorithms	3	1	0	4
2	CS502PC	Computer Networks	3	0	0	3
3	CS503PC	DevOps	3	0	0	3
4		Professional Elective-I	3	0	0	3
5		Professional Elective -II	3	0	0	3
6	CS521PC	Computer Networks Lab	0	0	2	1
7	CS522PC	DevOps Lab	0	0	2	1
8	EN501HS	Advanced English Communication Skills Lab	0	0	2	1
9	CS523PC	UI design- Flutter	0	0	2	1
10	*MC501	Intellectual Property Rights	3	0	0	0
		Total	18	1	8	20

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	CS601PC	Machine Learning	3	0	0	3
2	CS602PC	Formal Languages and Automata Theory	3	0	0	3
3	CS603PC	Artificial Intelligence	3	0	0	3
4		Professional Elective – III	3	0	0	3
5		Open Elective-I	3	0	0	3
6	CS621PC	Machine Learning Lab	0	0	2	1
7	CS622PC	Artificial Intelligence Laboratory	0	0	2	1
8		Professional Elective-III Lab	0	0	2	1
9	CS623PC	Industrial Oriented Mini Project/ Internship/ Skill Development Course (Big data-Spark)	0	0	4	2
10	*MC601	Environmental Science	3	0	0	0
		Total	18	0	8	20

Environmental Science in III Yr II Sem Should be Registered by Lateral Entry Students Only.

*MC – Satisfactory/Unsatisfactory

#Skill Course - 1 credit with 2 Practical Hours

Professional Elective - I

CS541PE	Quantum Computing
CS542PE	Advanced Computer Architecture
CS543PE	Data Analytics
EC523PE	Image Processing
CS545PE	Principles of Programming Languages

Professional Elective - II

CS546PE	Computer Graphics
EC522PE	Embedded Systems
CS548PE	Information Retrieval Systems
CS549PE	Distributed Databases
CS550PE	Natural Language Processing

Professional Elective - III

CS641PE	Full Stack Development
CS642PE	Internet of Things
CS643PE	Scripting Languages
CS644PE	Mobile Application Development
CS645PE	Software Testing Methodologies

Professional Elective - III LAB

CS661PE	Full Stack Development lab
CS662PE	Internet of Things lab
CS663PE	Scripting Languages lab
CS664PE	Mobile Application Development lab
CS665PE	Software Testing Methodologies lab

Courses in PE - III and PE - III Lab must be in 1-1 correspondence.

Open Elective -1:

CS681OE	Data Structures
CS682OE	Database Management Systems

DESIGN AND ANALYSIS OF ALGORITHMS
(Course Code: CS501PC)

B.Tech. III Year I Sem.   L T P C
                                                                         3 1 0 4

Prerequisites:

  1. A course on "Computer Programming and Data Structures".

  2. A course on "Advanced Data Structures".

Course Objectives:

  • Introduces the notations for analysis of the performance of algorithms and the data structure of disjoint sets.

  • Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate

  • Describes how to evaluate and compare different algorithms using worst-, average-, and best case analysis.

  • Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NP complete.

Course Outcomes:

  • Analyze the performance of algorithms

  • Choose appropriate data structures and algorithm design methods for a specified application

  • Understand the choice of data structures and the algorithm design methods

UNIT - I

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

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

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

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

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.

TEXT BOOK:

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

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.

COMPUTER NETWORKS
(Course Code: CS502PC)

B.Tech. III Year I Sem.   **L T P C**
                                                                                          3 0 0 3

Prerequisites

   1. A course on "Programming for problem solving"

   2. A course on "Data Structures"

Course Objectives

   • The objective of the course is to equip the students with a general overview of the concepts and fundamentals of computer networks.

   • Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

Course Outcomes

   • Gain the knowledge of the basic computer network technology.

   • Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.

   • Obtain the skills of subnetting and routing mechanisms.

   • Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

UNIT - I

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.

Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless Transmission.

Data link layer: Design issues, framing, Error detection and correction.

UNIT - II

Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.

Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.

Medium Access sublayer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT - III

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding,
Hierarchical routing,Broadcast, Multicast, distance vector routing, Congestion Control
Algorithms, Quality of Service,Internetworking, The Network layer in the internet.

UNIT - IV

Transport Layer: Transport Services, Elements of Transport protocols, Connection
management, TCP and UDP protocols.

UNIT - V

Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP,
Streaming audio and video.

TEXT BOOK:

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition.
   Pearson Education/PHI

REFERENCE BOOKS:

1. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson
   Education

2. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

DEVOPS

(Course Code: CS503PC)

B.Tech. III Year I Sem.   **L T P C**

                                                              3 0 0 3

Pre-Requisites:

    1. Software Engineering

    2. Software Project Management

Course Objectives:

- Understand the skill sets and high-functioning teams involved in Agile, DevOps and related methods to reach a continuous delivery capability.

- Implement automated system update and DevOps lifecycle.

Course Outcomes:

- Understand the various components of DevOps environment.

- Identify Software development models and architectures of DevOps

- Use different project management and integration tools.

- Select an appropriate testing tool and deployment model for project.

UNIT-I

Introduction to DevOps:

Introduction, Agile development model, DevOps and ITIL. DevOps process and Continuous Delivery, Release management, Scrum, Kanban, delivery pipeline, identifying bottlenecks.

UNIT-II

Software development models and DevOps:

DevOps Lifecycle for Business Agility, DevOps, and Continuous Testing. DevOps influence on Architecture: Introducing software architecture, The monolithic scenario, Architecture rules of thumb, The separation of concerns, Handling database migrations, Micro services and the data tier, DevOps, architecture, and resilience.

UNIT-III

Introduction to project management:

The need for source code control, the history of source code management, Roles and code, source code management system and migrations, shared authentication, Hosted Git servers, Different Git server implementations, Docker intermission, Gerrit, The pull request model, GitLab.

UNIT-IV

Integrating the system:

Build systems, Jenkins build server, Managing build dependencies, Jenkins plugins, and file system layout, The host server, Build slaves, Software on the host, Triggers, Job chaining and build pipelines, Build servers and infrastructure as code, Building by dependency order, Build phases, Alternative build servers, Collating quality measures.

UNIT-V

Testing Tools and Deployment:

Various types of testing, Automation of testing Pros and cons, Selenium - Introduction, Selenium features, JavaScript testing, Testing backend integration points, Test-driven development, REPL-driven development. Deployment of the system: Deployment systems, Virtualization stacks, code execution at the client, Puppet master and agents, Ansible, Deployment tools: Chef, Salt Stack and Docker.

TEXT BOOKS:

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

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

QUANTUM COMPUTING (Professional Elective – I)
(Course Code: CS541PE)

B.Tech. III Year I Sem.   L T P C

                                                                      3 0 0 3

Course Objectives

- To introduce the fundamentals of quantum computing
- The problem-solving approach using finite dimensional mathematics

Course Outcomes

- Understand basics of quantum computing
- Understand physical implementation of Qubit
- Understand Quantum algorithms and their implementation
- Understand The Impact of Quantum Computing on Cryptography

UNIT - I

History of Quantum Computing: Importance of Mathematics, Physics and Biology. Introduction to Quantum Computing: Bits Vs Qubits, Classical Vs Quantum logical operations

UNIT - II

Background Mathematics: Basics of Linear Algebra, Hilbert space, Probabilities and measurements. Background Physics: Paul's exclusion Principle, Superposition, Entanglement and super-symmetry, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. Background Biology: Basic concepts of Genomics and Proteomics (CentralDogma)

UNIT - III

Qubit: Physical implementations of Qubit. Qubit as a quantum unit of information. The Bloch sphere Quantum Circuits: single qubit gates, multiple qubit gates, designing the quantum circuits. Bell states.

UNIT - IV

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

UNIT - V

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

TEXT BOOKS:

    1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge

REFERENCE BOOKS:

1. Quantum Computing for Computer Scientists by Noson S. Yanofsky and Mirco A. Mannucci

2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II

3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms

ADVANCED COMPUTER ARCHITECTURE (Professional Elective – I)
(Course Code: CS542PE)

B.Tech. III Year I Sem. **L T P C**

3 0 0 3

Prerequisites: Computer Organization

Course Objectives:

- To impart the concepts and principles of parallel and advanced computer architectures.
- To develop the design techniques of Scalable and multithreaded Architectures.
- To Apply the concepts and techniques of parallel and advanced computer architectures to design modern computer systems

Course Outcomes:

- Computational models and Computer Architectures.
- Concepts of parallel computer models.
- Scalable Architectures, Pipelining, Superscalar processors

UNIT - I

Theory of Parallelism, Parallel computer models, The State of Computing, Multiprocessors and Multicomputers, Multivector and SIMD Computers, PRAM and VLSI models, Architectural development tracks, Program and network properties, Conditions of parallelism, Program partitioning and Scheduling, Program flow Mechanisms, System interconnect Architectures.

UNIT - II

Principles of Scalable performance, Performance metrics and measures, Parallel Processing applications, Speed up performance laws, Scalability Analysis and Approaches, Hardware Technologies, Processes and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors

UNIT - III

Shared-Memory Organizations, Sequential and weak consistency models, Pipelining and superscalar techniques, Linear Pipeline Processors, Non-Linear Pipeline Processors, Instruction Pipeline design, Arithmetic pipeline design, superscalar pipeline design.

UNIT - IV

Parallel and Scalable Architectures, Multiprocessors and Multicomputers, Multiprocessor system interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputers, Message-passing Mechanisms, Multivetor and SIMD computers.

UNIT - V

Vector Processing Principles, Multivector Multiprocessors, Compound Vector processing,

SIMD computer Organizations, The connection machine CM-5.

TEXT BOOK

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

REFERENCE BOOKS:

1. Computer Architecture, J.L. Hennessy and D.A. Patterson, 4th Edition, ELSEVIER.

2. Advanced Computer Architectures, S.G.Shiva, Special Indian edition, CRC, Taylor &Francis.

3. Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G.Wellein, CRC Press, Taylor & Francis Group.

3. Advanced Computer Architecture, D. Sima, T. Fountain, P. Kacsuk, Pearson education.

5. Computer Architecture, B. Parhami, Oxford Univ. Press.

DATA ANALYTICS (Professional Elective – I)
(Course Code: CS543PE)

B.Tech. III Year I Sem. L T P C

3 0 0 3

Prerequisites

1. A course on "Database Management Systems".

2. Knowledge of probability and statistics.

Course Objectives:

- To explore the fundamental concepts of data analytics.

- To learn the principles and methods of statistical analysis

- Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms.

- To understand the various search methods and visualization techniques.

Course Outcomes: After completion of this course students will be able to

- Understand the impact of data analytics for business decisions and strategy

- Carry out data analysis/statistical analysis

- To carry out standard data visualization and formal inference procedures

- Design Data Architecture

- Understand various Data Sources

UNIT - I

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

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

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

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

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

TEXT BOOKS:

    1. Student's Handbook for Associate Analytics – II, III.

    2. Data Mining Concepts and Techniques, Han, Kamber, 3rd Edition, Morgan Kaufmann Publishers.

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.

IMAGE PROCESSING (Professional Elective – I)
(Course Code: EC523PE)

B.Tech. III Year I Sem.   **L T P C**
                                                                3 0 0 3

Prerequisites

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"

Course Objectives

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

Course Outcomes

- Demonstrate the knowledge of the basic concepts of two-dimensional signal acquisition, sampling, and quantization.
- Demonstrate the knowledge of filtering techniques.
- Demonstrate the knowledge of 2D transformation techniques.
- Demonstrate the knowledge of image enhancement, segmentation, restoration and compression techniques.

UNIT - I

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

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

UNIT - III

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

UNIT - IV

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

UNIT - V

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

TEXT BOOK:

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

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.

PRINCIPLES OF PROGRAMMING LANGUAGES (Professional Elective – I)
(Course Code: CS545PE)

B.Tech. III Year I Sem.   **L T P C**
                                                              3 0 0 3

Prerequisites

    1. A course on "Mathematical Foundations of Computer Science".

    2. A course on "Computer Programming and Data Structures".

Course Objectives

- Introduce important paradigms of programming languages

- To provide conceptual understanding of high-level language design and implementation

- Topics include programming paradigms; syntax and semantics; data types, expressions and statements; subprograms and blocks; abstract data types; concurrency; functional and logic programming languages; and scripting languages

Course Outcomes

- Acquire the skills for expressing syntax and semantics in formal notation

- Identify and apply a suitable programming paradigm for a given computing application

- Gain knowledge of the features of various programming languages and their comparison

UNIT - I

Preliminary Concepts: 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

Names, Bindings, and Scopes: Introduction, Names, Variables, Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants

Data Types: Introduction, Primitive Data Types, Character String Types, User Defined 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 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

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

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

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)

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.

REFERENCE BOOKS:

      1. Programming Languages, 2nd Edition, A.B. Tucker, R. E. Noonan, TMH.

      2. Programming Languages, K. C. Louden, 2nd Edition, Thomson, 2003.

COMPUTER GRAPHICS (Professional Elective – II)
(Course Code: CS546PE)

B.Tech. III Year I Sem.	**L T P C**
	3 0 0 3

Prerequisites

1. Programming for problem solving and Data Structures

Course Objectives

- Provide the basics of graphics systems including Points and lines, line drawing algorithms, 2D, 3D objective transformations

Course Outcomes

- Explore applications of computer graphics

- Understand 2D, 3D geometric transformations and clipping algorithms

- Understand 3D object representations, curves, surfaces, polygon rendering methods, color models

- Analyze animation sequence and visible surface detection methods

UNIT - I

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

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

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

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

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, BSPtree method, area sub-division method and octree method.

TEXT BOOKS:

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

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.

EMBEDDED SYSTEMS (Professional Elective – II)
(Course Code: EC522PE)

B.Tech. III Year I Sem.   **L T P C**

                                                                   3 0 0 3

Pre-requisites:

1. A course on "Digital Logic Design and Microprocessors"

2. A course on "Computer Organization and Architecture"

Course Objectives:

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

Course Outcomes:

- Expected to understand the selection procedure of processors in the embedded domain.
- Design procedure of embedded firm ware.
- Expected to visualize the role of realtime operating systems in embedded systems.
- Expected to evaluate the correlation between task synchronization and latency issues

UNIT - I

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

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

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

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

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.

TEXT BOOK:

1. Embedded Systems, Raj Kamal, 2nd edition, Tata Mc Graw Hill

2. Shibu K V, "Introduction to Embedded Systems", Second Edition, Mc Graw Hill

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.

INFORMATION RETRIEVAL SYSTEMS (Professional Elective – II)
(Course Code: CS548PE)

B.Tech. III Year I Sem.   **L T P C**

                                                                     3 0 0 3

Prerequisites:

   1. Data Structures

Course Objectives:

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

Course Outcomes:

- Ability to apply IR principles to locate relevant information large collections of data
- Ability to design different document clustering algorithms
- Implement retrieval systems for web search tasks.
- Design an Information Retrieval System for web search tasks.

UNIT - I

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

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

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

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

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

TEXT BOOK:

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

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.

DISTRIBUTED DATABASES (Professional Elective – II)
(Course Code: CS549PE)

B.Tech. III Year I Sem.   **L T P C**

                                                                               3 0 0 3

Prerequisites:

   1. A course on "Database Management Systems"

Course Objectives:

- 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.

- Introduce basic principles and implementation techniques of distributed database systems.

- Equip students with principles and knowledge of parallel and object-oriented databases.

- Topics include distributed DBMS architecture and design; query processing and optimization; distributed transaction management and reliability; parallel and object database management systems.

Course Outcomes:

- Understand theoretical and practical aspects of distributed database systems.

- Study and identify various issues related to the development of distributed database systems.

- Understand the design aspects of object-oriented database systems and related developments.

UNIT - I

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

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

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

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

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

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.

REFERENCE BOOK:

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

NATURAL LANGUAGE PROCESSING (Professional Elective – II)
(Course Code: CS550PE)

B.Tech. III Year I Sem.   **L T P C**

                                                                               3 0 0 3

Prerequisites:

    1. Data structures and compiler design

Course Objectives:

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

Course Outcomes:

- Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.

- Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems

- Manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.

- Design, implement, and analyze NLP algorithms; and design different language modeling Techniques.

UNIT - I

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

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

UNIT – III

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

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

UNIT - IV

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

UNIT - V

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

TEXT BOOKS:

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

REFERENCE BOOK:

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.

COMPUTER NETWORKS LAB
(Course Code: CS521PC)

B.Tech. III Year I Sem.   **L T P C**

                                                              0 0 2 1

Course Objectives

- To understand the working principle of various communication protocols.
- To understand the network simulator environment and visualize a network topology and observe its performance
- To analyze the traffic flow and the contents of protocol frames

Course Outcomes

- Implement data link layer farming methods
- Analyze error detection and error correction codes.
- Implement and analyze routing and congestion issues in network design.
- Implement Encoding and Decoding techniques used in presentation layer
- To be able to work with different network tools

List of Experiments

1. Implement the data link layer framing methods such as character, character-stuffing and bitstuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijsktra's algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting techniques used in buffers.
10. **Wireshark**

    i. Packet Capture Using Wire shark

    ii. Starting Wire shark

    iii. Viewing Captured Traffic

    iv. Analysis and Statistics & Filters.

How to run Nmap scan

Operating System Detection using Nmap

Do the following using NS2 Simulator

      i. NS2 Simulator-Introduction

      ii. Simulate to Find the Number of Packets Dropped

      iii. Simulate to Find the Number of Packets Dropped by TCP/UDP

      iv. Simulate to Find the Number of Packets Dropped due to Congestion

      v. Simulate to Compare Data Rate & Throughput.

      vi. Simulate to Plot Congestion for Different Source/Destination

      vii. Simulate to Determine the Performance with respect to Transmission of

Packets

TEXT BOOK:

    1. Computer Networks, Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson
      Education/PHI.

REFERENCE BOOKS:

    1. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson
      Education.

    2. Data Communications and Networking – Behrouz A. Forouzan. 3rd Edition, TMH.

DEVOPS LAB

(Course Code: CS522PC)

B.Tech. III Year I Sem. **L T P C**

0 0 2 1

Course Objectives:

- 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.

Course Outcomes:

1. Understand the need of DevOps tools
2. Understand the environment for a software application development
3. Apply different project management, integration and development tools
4. Use Selenium tool for automated testing of application

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.

TEXT BOOKS:

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

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.

ADVANCED ENGLISH COMMUNICATION SKILLS LAB
(Course Code: EN501HS)

B.Tech. III Year I Sem. **L T P C**

0 0 2 1

1. Introduction

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 and 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. Objectives:

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

- To improve the students' fluency in English, with a focus on vocabulary

- To enable them to listen to English spoken at normal conversational speed by educated English speakers

- To respond appropriately in different socio-cultural and professional contexts

- To communicate their ideas relevantly and coherently in writing

- To prepare the students for placements.

3. 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: 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 – Subskills of reading - Reading for facts, negative facts and Specific Details- Guessing Meanings from Context, Inferring Meaning - Critical Reading — Reading Comprehension – Exercises for Practice.

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

3. 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 Clarification, Making a Request, Asking for and Refusing Permission, Participating in a Small Talk – Oral presentations (individual and group) through JAM sessions-PPTs – 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

4. 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.

5. 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.

4. 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

5. 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

6. 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 BlackSwan Pvt. Ltd.

3. Bailey, Stephen. (2018). Academic Writing: A Handbook for International Students. (5th Edition). 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 Edition) Oxford University Press.

6. Anderson, Paul V. (2007). Technical Communication. Cengage Learning Pvt. Ltd. New Delhi.

7. McCarthy, 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. BloomsburyPublishing.

UI DESIGN-FLUTTER
(Course Code: CS523PC)

B.Tech. III Year I Sem.   **L T P C**

                                                                0 0 2 1

Course Objectives:

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

Course Outcomes:

- Implements Flutter Widgets and Layouts
- Responsive UI Design and with Navigation in Flutter
- Create custom widgets for specific UI elements and also Apply styling using themes and custom styles.
- Design a form with various input fields, along with validation and error handling
- Fetches data and write code for unit Test for UI components and also animation

List of Experiments: Students need to implement the following 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.

TEXT BOOK:

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

INTELLECTUAL PROPERTY RIGHTS

(Course Code: *MC501)

B.Tech. III Year I Sem. **L T P C**

3 0 0 0

Course Objectives:

- Significance of intellectual property and its protection
- Introduce various forms of intellectual property

Course Outcomes:

- Distinguish and Explain various forms of IPRs.
- Identify criteria to fit one's own intellectual work in particular form of IPRs.
- Apply statutory provisions to protect particular form of IPRs.
- Appraise new developments in IPR laws at national and international level

UNIT – I

Introduction to Intellectual property: Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

UNIT – II

Trade Marks: Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

UNIT – III

Law of copyrights: Fundamental of copyright law, originality of material, rights of reproduction, rights to perform the work publicly, copyright ownership issues, copyright registration, notice of copyright, International copyright law.

Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

UNIT – IV

Trade Secrets: Trade secret law, determination of trade secret status, liability for misappropriations of trade secrets, protection for submission, trade secret litigation.

Unfair competition: Misappropriation right of publicity, false advertising.

UNIT – V

New development of intellectual property: new developments in trade mark law; copyright law, patent law, intellectual property audits.

International overview on intellectual property, international – trade mark law, copyright law, international patent law, and international development in trade secrets law.

TEXT BOOK:

    1. Intellectual property right, Deborah. E. Bouchoux, Cengage learning.

REFERENCE BOOK:

    1. Intellectual property right – Unleashing the knowledge economy, prabuddha ganguli, Tata McGraw Hill Publishing company ltd.

MACHINE LEARNING

(Course Code: CS601PC)

B.Tech. III Year II Sem.   **L T P C**

                                                                                      3 0 0 3

Course Objectives:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To have a thorough understanding of the Supervised and Unsupervised learning techniques
- To study the various probability-based learning techniques

Course Outcomes:

- Distinguish between, supervised, unsupervised and semi-supervised learning
- Understand algorithms for building classifiers applied on datasets of non-linearly separable classes
- Understand the principles of evolutionary computing algorithms
- Design an ensembler to increase the classification accuracy

UNIT - I

Learning – Types of Machine Learning – Supervised Learning – The Brain and the Neuron – Design a Learning System – Perspectives and Issues in Machine Learning – Concept Learning Task – Concept Learning as Search – Finding a Maximally Specific Hypothesis – Version Spaces and the Candidate Elimination Algorithm – Linear Discriminants: – Perceptron – Linear Separability – Linear Regression.

UNIT - II

Multi-layer Perceptron– Going Forwards – Going Backwards: Back Propagation Error – Multi-layer Perceptron in Practice – Examples of using the MLP – Overview – Deriving Back-Propagation – Radial Basis Functions and Splines – Concepts – RBF Network – Curse of Dimensionality – Interpolations and Basis Functions – Support Vector Machines

UNIT - III

Learning with Trees – Decision Trees – Constructing Decision Trees – Classification and Regression Trees – Ensemble Learning – Boosting – Bagging – Different ways to Combine Classifiers – Basic Statistics – Gaussian Mixture Models – Nearest Neighbor Methods – Unsupervised Learning – K means Algorithms

UNIT - IV

Dimensionality Reduction – Linear Discriminant Analysis – Principal Component Analysis –
Factor Analysis – Independent Component Analysis – Locally Linear Embedding – Isomap –
Least Squares Optimization

Evolutionary Learning – Genetic algorithms – Genetic Offspring: - Genetic Operators – Using
Genetic

Algorithms

UNIT - V

Reinforcement Learning – Overview – Getting Lost Example

Markov Chain Monte Carlo Methods – Sampling – Proposal Distribution – Markov Chain

Monte Carlo – Graphical Models – Bayesian Networks – Markov Random Fields – Hidden

Markov Models – Tracking Methods

TEXT BOOKS:

1. Stephen Marsland, ―Machine Learning – An Algorithmic Perspective, Second   3
   Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series,
   2014.

REFERENCE BOOKS:

1. Tom M Mitchell, ―Machine Learning, First Edition, McGraw Hill Education, 2013.

2. Peter Flach, ―Machine Learning: The Art and Science of Algorithms that Make
   Sense of Data‖, First Edition, Cambridge University Press, 2012.

3. Jason Bell, ―Machine learning – Hands on for Developers and Technical
   Professionals‖, First Edition, Wiley, 2014

4. Ethem Alpaydin, ―Introduction to Machine Learning 3e (Adaptive Computation and
   Machine Learning Series), Third Edition, MIT Press, 2014

FORMAL LANGUAGES AND AUTOMATA THEORY
(Course Code: CS602PC)

B.Tech. III Year II Sem.   **L T P C**
                                                              3 0 0 3

Course Objectives

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and undecidability.

Course Outcomes

- Understand the concept of abstract machines and their power to recognize the languages.
- Employ finite state machines for modeling and solving computing problems.
- Design context free grammars for formal languages.
- Distinguish between decidability and undecidability.

UNIT - I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA,Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA, Moore and Melay machines

UNIT - II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT - III

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

UNIT - IV

Normal Forms for Context- Free Grammars: Eliminating useless symbols, Eliminating €-Productions. Chomsky Normal form Greibach Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications

Closure Properties of Context-Free Languages: Closure properties of CFL's, Decision Properties of CFL's Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

UNIT - V

Types of Turing machine: Turing machines and halting

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd edition, PHI.

REFERENCE BOOKS:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Textbook on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

ARTIFICIAL INTELLIGENCE
(Course Code: CS603PC)

B.Tech. III Year II Sem.   **L T P C**
                                                                 3 0 0 3

Prerequisites:

   1. Programming for problem solving, Data Structures.

Course Objectives:

   • To learn the distinction between optimal reasoning Vs. human like reasoning

   • To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.

   • To learn different knowledge representation techniques.

   • To understand the applications of AI, namely game playing, theorem proving, and machine learning.

Course Outcomes:

   • Understand search strategies and intelligent agents

   • Understand different adversarial search techniques

   • Apply propositional logic, predicate logic for knowledge representation

   • Apply AI techniques to solve problems of game playing, and machine learning.

UNIT - I

Introduction to AI, Intelligent Agents, problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth-first search, Uniform cost search, Depth-first search, Iterative deepening Depth-first search, Bidirectional search, Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions, Beyond Classical Search: Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces

UNIT - II

Problem Solving by Search-II and Propositional Logic

Adversarial Search: Games, Optimal Decisions in Games, Alpha–Beta Pruning, Imperfect Real-Time Decisions. Constraint Satisfaction Problems: Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems. Propositional Logic: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic.

UNIT - III

Logic and Knowledge Representation

First-Order Logic: Representation, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

UNIT - IV

Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

Classical Planning: Definition of Classical Planning, Algorithms for Planning with State-Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.

UNIT - V

Uncertain knowledge and Learning Uncertainty: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use

Probabilistic Reasoning: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.

TEXT BOOK:

1. Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henry Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

FULL STACK DEVELOPMENT (Professional Elective – III)
(Course Code: CS641PE)

B.Tech. III Year II Sem. **L T P C**
3 0 0 3

Pre-Requisites:

    1. Object Oriented Programming

    2. Web Technologies

Course Objectives:

- Students will become familiar to implement fast, efficient, interactive and scalable web applications using run time environment provided by the full stack components.

Course Outcomes:

- Understand Full stack components for developing web application.

- Apply packages of NodeJS to work with Data, Files, Http Requests and Responses.

- Use MongoDB data base for storing and processing huge data and connects with NodeJS application.

- Design faster and effective single page applications using Express and Angular.

- Create interactive user interfaces with react components.

UNIT-I

Introduction to Full Stack Development:

Understanding the Basic Web Development Framework- User, Browser, Webserver, Backend Services, Full Stack Components - Node.js, MongoDB, Express, React, Angular. Java Script Fundamentals, NodeJS- Understanding Node.js, Installing Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks

UNIT-II

Node.js:

Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Accessing the File System from Node.js- Opening, Closing, Writing, Reading Files and other File System Tasks. Implementing HTTP Services in Node.js- Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects, Implementing HTTP Clients and Servers in Node.js, Implementing HTTPS Servers and Clients. Using Additional Node.js Modules-Using the os Module, Using the util Module, Using the dns Module, Using the crypto Module.

UNIT-III

MongoDB:

Need of NoSQL, Understanding MongoDB, MongoDB Data Types, Planning Your Data Model, Building the MongoDB Environment, Administering User Accounts, Configuring Access Control, Administering Databases, Managing Collections, Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Understanding the Objects Used in the MongoDB Node.js Driver, Accessing and Manipulating Databases, Accessing and Manipulating Collections

UNIT-IV

Express and Angular:

Getting Started with Express, Configuring Routes, Using Requests Objects, Using Response Objects. Angular: importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing Angular Services in Web Applications.

UNIT-V

React:

Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React, Rendering and Life Cycle Methods in React, Working with forms in React, integrating third party libraries, Routing in React.

TEXT BOOKS:

    1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.

    2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

    1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress, 2019.

    2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.

    3. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

INTERNET OF THINGS (Professional Elective – III)
(Course Code: CS642PE)

B.Tech. III Year II Sem. **L T P C**

3 0 0 3

Pre-Requisites: Computer organization, Computer Networks

Course Objectives:

- To introduce the terminology, technology and its applications
- To introduce the concept of M2M (machine to machine) with necessary protocols
- To introduce the Python Scripting Language which is used in many IoT devices
- To introduce the Raspberry PI platform, that is widely used in IoT applications
- To introduce the implementation of web-based services on IoT devices

Course Outcomes:

- Interpret the impact and challenges posed by IoT networks leading to new architectural models.
- Compare and contrast the deployment of smart objects and the technologies to connect them to network.
- Appraise the role of IoT protocols for efficient network communication.
- Identify the applications of IoT in Industry.

UNIT - I

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels and Deployment Templates Domain Specific IoTs – Home automation, Environment, Agriculture, Health and Lifestyle

UNIT - II

IoT and M2M – M2M, Difference between IoT and M2M, SDN and NFV for IoT,

IoT System Management with NETCOZF, YANG- Need for IoT system Management, Simple Network management protocol, Network operator requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG

UNIT - III

IoT Systems – Logical design using Python-Introduction to Python – Python Data types & Data structures, Control flow, Functions, Modules, Packaging, File handling, Data/Time operations, Classes, Exception, Python packages of Interest for IoT

UNIT - IV

IoT Physical Devices and Endpoints - Raspberry Pi, Linux on Raspberry Pi, Raspberry Pi Interfaces,Programming Raspberry PI with Python, Other IoT devices.

IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs, WAMP-AutoBahn for IoT, Xively Cloud for IoT, Python web application framework –Django, Designing a RESTful web API

UNIT V

Case studies- Home Automation, Environment-weather monitoring-weather reporting- air pollution monitoring, Agriculture.

TEXT BOOK:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547.

REFERENCE BOOK:

1. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759.

SCRIPTING LANGUAGES (Professional Elective – III)
(Course Code: CS643PE)

B.Tech. III Year II Sem.   **L T P C**

                                                                                3 0 0 3

Prerequisites:

1. A course on "Computer Programming and Data Structures".

2. A course on "Object Oriented Programming Concepts".

Course Objectives:

- This course introduces the script programming paradigm

- Introduces scripting languages such as Perl, Ruby and TCL.

- Learning TCL

Course Outcomes:

- Comprehend the differences between typical scripting languages and typical system and application programming languages.

- Gain knowledge of the strengths and weakness of Perl, TCL and Ruby; and select an appropriate language for solving a given problem.

- Acquire programming skills in scripting language

UNIT - I

Introduction: Ruby, Rails, The structure and Execution of Ruby Programs, Package Management with RUBYGEMS, Ruby and web: Writing CGI scripts, cookies, Choice of Webservers, SOAP and web services

RubyTk – Simple Tk Application, widgets, Binding events, Canvas, scrolling

UNIT - II

Extending Ruby: Ruby Objects in C, the Jukebox extension, Memory allocation, Ruby Type System, Embedding Ruby to Other Languages, Embedding a Ruby Interpreter

UNIT - III

Introduction to PERL and Scripting

Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines.

UNIT - IV

Advanced perl

Finer points of looping, pack and unpack, filesystem, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

UNIT - V

TCL

TCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, input/output, procedures, strings, patterns, files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, trapping errors, event driven programs, making applications internet aware, Nuts and Bolts Internet Programming, Security Issues, C Interface.

Tk

Tk-Visual Tool Kits, Fundamental Concepts of Tk, Tk by example, Events and Binding, Perl-Tk.

TEXT BOOKS:

1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly
3. "Programming Ruby" The Pramatic Progammers guide by Dabve Thomas Second edition

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B. Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O'Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
5. Perl Power, J. P. Flynt, Cengage Learning.

MOBILE APPLICATION DEVELOPMENT (Professional Elective – III)
(Course Code: CS644PE)

B.Tech. III Year II Sem.   **L T P C**

                                                               3 0 0 3

Prerequisites

   1. Acquaintance with JAVA programming

   2. A Course on DBMS

Course Objectives

   - To demonstrate their understanding of the fundamentals of Android operating systems

   - To improves their skills of using Android software development tools

   - To demonstrate their ability to develop software with reasonable complexity on mobile platform

   - To demonstrate their ability to deploy software to mobile devices

   - To demonstrate their ability to debug programs running on mobile devices

Course Outcomes

   - Understand the working of Android OS Practically.

   - Develop Android user interfaces

   - Develop, deploy and maintain the Android Applications.

UNIT - I

Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Android Studio, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes

Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes

UNIT - II

Android User Interface: Measurements – Device and pixel density independent measuring unit – s Layouts – Linear, Relative, Grid and Table Layouts

User Interface (UI) Components –Editable and non-editable TextViews, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers

Event Handling – Handling clicks or changes of various UI components

Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

UNIT - III

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS

Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity

Notifications – Creating and Displaying notifications, Displaying Toasts

UNIT - IV

Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference

UNIT - V

Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and etindelg data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)

TEXT BOOK:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012

REFERENCE BOOKS:

1. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning,2013
2. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013

SOFTWARE TESTING METHODOLOGIES (Professional Elective – III)
(Course Code: CS645PE)

B.Tech. III Year II Sem.   **L T P C**

                                                                          3 0 0 3

Prerequisites

    1. Software Engineering

Course Objectives

- To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
- To develop skills in software test automation and management using the latest tools.

Course Outcomes:

- Understand purpose of testing and path testing
- Understand strategies in data flow testing and domain testing
- Develop logic-based test strategies
- Understand graph matrices and its applications
- Implement test cases using any testing automation tool

UNIT - I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II

Transaction Flow Testing: transaction flows, transaction flow testing techniques.

Data Flow testing: Basics of data flow testing, strategies in data flow testing, application of data flow testing.

Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces

testing, domain and interface testing, domains and testability.

UNIT - III

Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV

State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V

Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like Jmeter/selenium/soapUI/Catalon).

TEXT BOOKS:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.

2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

REFERENCE BOOKS:

1. The craft of software testing - Brian Marick, Pearson Education.

2. Software Testing Techniques – SPD(Oreille)

3. Software Testing in the Real World – Edward Kit, Pearson.

4. Effective methods of Software Testing, Perry, John Wiley.

5. Art of Software Testing – Meyers, John Wiley.

DATA STRUCTURES (Open Elective – I)
(Course Code: CS681OE)

B.Tech. III Year II Sem.   **L T P C**

                                                                     3 0 0 3

Prerequisites

  1. A course on "Programming for Problem Solving

Course Objectives

- Exploring basic data structures such as stacks and queues.

- Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.

- Introduces sorting and pattern matching algorithms

Course Outcomes

  1. Ability to select the data structures that efficiently model the information in a problem.

  2. Ability to assess efficiency trade-offs among different data structure implementations or combinations.

  3. Implement and know the application of algorithms for sorting and pattern matching.

  4. Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees.

UNIT - I

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks- Operations, array and linked representations of stacks, stack applications, Queues- operations, array and linked representations.

UNIT - II

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.

Hash Table Representation: hash functions, collision resolution-separate chaining, open addressinglinear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

Search Trees: Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion, B- Trees, B+ Trees, AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching, Red –Black, Splay Trees.

UNIT - IV

Graphs: Graph Implementation Methods. Graph Traversal Methods.

Sorting: Quick Sort, Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

1. Fundamentals of Data Structures in C, 2 nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press.

2. Data Structures using C – A. S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

REFERENCE BOOK:

1. Data Structures: A Pseudocode Approach with C, 2 nd Edition, R. F. Gilberg and B.A.Forouzan, Cengage Learning.

DATABASE MANAGEMENT SYSTEMS (Open Elective – I)
(Course Code: CS682OE)

B.Tech. III Year II Sem.   **L T P C**

                                                                                             3 0 0 3

Prerequisites: A course on "Data Structures".

Course Objectives:

- To understand the basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

Course Outcomes:

- Gain knowledge of fundamentals of DBMS, database design and normal forms
- Master the basics of SQL for retrieval and management of data.
- Be acquainted with the basics of transaction processing and concurrency control.
- Familiarity with database storage structures and access techniques

UNIT - I

Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT - II

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views.

Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT - III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL,

triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies, FOURTH normal form, FIFTH normal form.

UNIT - IV

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions.

UNIT - V

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index Structure.

TEXT BOOKS:

1. Database System Concepts, Silberschatz, Korth, McGraw hill, V edition. 3rd Edition
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

MACHINE LEARNING LAB
(Course Code: CS621PC)

B.Tech. III Year II Sem. **L T P C**

0 0 2 1

Course Objective:

- The objective of this lab is to get an overview of the various machine learning techniques and can demonstrate them using python.

Course Outcomes:

- Understand modern notions in predictive data analysis
- Select data, model selection, model complexity and identify the trends
- Understand a range of machine learning algorithms along with their strengths and weaknesses
- Build predictive models from data and analyze their performance

List of Experiments

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode Measure of Dispersion: Variance, Standard Deviation
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy
3. Study of Python Libraries for ML application such as Pandas and Matplotlib
4. Write a Python program to implement Simple Linear Regression
5. Implementation of Multiple Linear Regression for House Price Prediction using sklearn
6. Implementation of Decision tree using sklearn and its parameter tuning
7. Implementation of KNN using sklearn
8. Implementation of Logistic Regression using sklearn
9. Implementation of K-Means Clustering
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

TEXT BOOK:

1. Machine Learning – Tom M. Mitchell, - MGH.

REFERENCE BOOK:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis.

ARTIFICIAL INTELLIGENCE LAB
(Course Code: CS622PC)

B.Tech. III Year II Sem. **L T P C**

0 0 2 1

Course Objectives:

- Become familiar with basic principles of AI toward problem solving, knowledge representation, and learning.

Course Outcomes:

- Apply basic principles of AI in solutions that require problem solving, knowledge representation, and learning.

LIST OF EXPERIMENTS

Write a Program to Implement the following using Python.

1. Breadth First Search

2. Depth First Search

3. Tic-Tac-Toe game

4. 8-Puzzle problem

5. Water-Jug problem

6. Travelling Salesman Problem

7. Tower of Hanoi

8. Monkey Banana Problem

9. Alpha-Beta Pruning

10. 8-Queens Problem

TEXT BOOK:

1. Artificial Intelligence a Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)

2. Artificial Intelligence, 3rd Edn., Patrick Henny Winston, Pearson Education.

3. Artificial Intelligence, Shivani Goel, Pearson Education.

FULL STACK DEVELOPMENT LAB (Professional Elective – III)
(Course Code: CS661PE)

B.Tech. III Year II Sem. **L T P C**

0 0 2 1

Pre-Requisites:

    1. Object Oriented Programming

    2. Web Technologies

Course Objectives:

- Introduce fast, efficient, interactive and scalable web applications using run time environment provided by the full stack components.

Course Outcomes:

- Design flexible and responsive Web applications using Node JS, React, Express and Angular.

- Perform CRUD operations with MongoDB on huge amount of data.

- Develop real time applications using react components.

- Use various full stack modules to handle http requests and responses.

List of Experiments

    1. Create an application to setup node JS environment and display "Hello World".

    2. Create a Node JS application for user login system.

    3. Write a Node JS program to perform read, write and other operations on a file.

    4. Write a Node JS program to read form data from query string and generate response Using NodeJS

    5. Create a food delivery website where users can order food from a particular restaurant listed in the website for handling http requests and responses using NodeJS.

    6. Implement a program with basic commands on databases and collections using MongoDB.

    7. Implement CRUD operations on the given dataset using MongoDB.

    8. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.

    9. Develop an angular JS form to apply CSS and Events.

    10. Develop a Job Registration form and validate it using angular JS.

    11. Write an angular JS application to access JSON file data of an employee from a server using $http service.

12. Develop a web application to manage student information using Express and Angular JS.

13. Write a program to create a simple calculator Application using React JS.

14. Write a program to create a voting application using React JS

15. Develop a leave management system for an organization where users can apply different types of leaves such as casual leave and medical leave. They also can view the available number of days using react application.

16. Build a music store application using react components and provide routing among the web pages.

17. Create a react application for an online store which consist of registration, login, product information pages and implement routing to navigate through these pages.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley,2019.

2. Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress,2019.

2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.

3. Brad Green& Seshadri. Angular JS. 1st Edition. O'Reilly Media, 2013.

4. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

INTERNET OF THINGS LAB (PROFESSIONAL ELECTIVE – III)
(Course Code: CS662PE)

B.Tech. III Year II Sem.   **L T P C**

                                                                          0 0 2 1

Course Objectives

- To introduce the raspberry PI platform, that is widely used in IoT applications
- To introduce the implementation of distance sensor on IoT devices

Course Outcomes

- Ability to introduce the concept of M2M (machine to machine) with necessary protocols and get awareness in implementation of distance sensor
- Get the skill to program using python scripting language which is used in many IoT devices

List of Experiments

1. Using Raspberry pi

    a. Calculate the distance using a distance sensor.

    b. Interface an LED and switch with Raspberry pi.

    c. Interface an LDR with Raspberrry Pi.

2. Using Arduino

    a. Calculate the distance using a distance sensor.

    b. Interface an LED and switch with Aurdino.

    c. Interface an LDR with Aurdino

    d. Calculate temperature using a temperature sensor.

3. Using Node MCU

    a. Calculate the distance using a distance sensor.

    b. Interface an LED and switch with Raspberry pi.

    c. Interface an LDR with Node MCU

    d. Calculate temperature using a temperature sensor.

4. Installing OS on Raspberry Pi

    a) Installation using PiImager

    b) Installation using image file

    - `Downloading an Image
    - Writing the image to an SD card
        - using Linux
        - using Windows

• Booting up Follow the instructions given in the URL

https://www.raspberrypi.com/documentation/computers/getting-started.html

5. Accessing GPIO pins using Python

    a) Installing GPIO Zero library.

      update your repositories list:

      install the package for Python 3:

    b) Blinking an LED connected to one of the GPIO pin

    c) Adjusting the brightness of an LED Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) using the in-built PWM wavelength.

6. Create a DJANGO project and an app.

7. Create a DJANGO view for weather station REST API

8. Create DJANGO template

9. Configure MYSQL with DJANGO framework

TEXT BOOKS:

    1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547.

    2. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759.

REFERENCE BOOKS:

    1. Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", ISBN 978-3- 642-19156-5 e-ISBN 978-3-642-19157-2, Springer, 2016

    2. N. Ida, Sensors, Actuators and Their Interfaces, Scitech Publishers, 2014.

SCRIPTING LANGUAGES LAB (Professional Elective – III)
(Course Code: CS663PE)

B.Tech. III Year II Sem. L T P C

0 0 2 1

Prerequisites: Any High level programming language (C, C++)

Course Objectives

- To Understand the concepts of scripting languages for developing web based projects
- To understand the applications the of Ruby, TCL, Perl scripting languages

Course Outcomes

- Ability to understand the differences between Scripting languages and programming languages
- Gain some fluency programming in Ruby, Perl, TCL

LIST OF EXPERIMENTS

1. Write a Ruby script to create a new string which is n copies of a given string where n is a non-negative integer

2. Write a Ruby script which accept the radius of a circle from the user and compute the parameter and area.

3. Write a Ruby script which accept the users first and last name and print them in reverse order with a space between them

4. Write a Ruby script to accept a filename from the user print the extension of that

5. Write a Ruby script to find the greatest of three numbers

6. Write a Ruby script to print odd numbers from 10 to 1

7. Write a Ruby script to check two integers and return true if one of them is 20 otherwise return their sum

8. Write a Ruby script to check two temperatures and return true if one is less than 0 and the other is greater than 100

9. Write a Ruby script to print the elements of a given array

10. Write a Ruby program to retrieve the total marks where subject name and marks of a student stored in a hash

11. Write a TCL script to find the factorial of a number

12. Write a TCL script that multiplies the numbers from 1 to 10

13. Write a TCL script for sorting a list using a comparison function

14. Write a TCL script to (i) create a list (ii) append elements to the list (iii) Traverse the list (iv) Concatenate the list

15. Write a TCL script to comparing the file modified times.

16. Write a TCL script to Copy a file and translate to native format.

17. a) Write a Perl script to find the largest number among three numbers.

   b) Write a Perl script to print the multiplication tables from 1-10 using subroutines.

18. Write a Perl program to implement the following list of manipulating functions

   a) Shift

   b) Unshift

   c) Push

19. a) Write a Perl script to substitute a word, with another word in a string.

    b) Write a Perl script to validate IP address and email address.

20. Write a Perl script to print the file in reverse order using command line arguments

TEXT BOOKS:

   1. The World of Scripting Languages, David Barron,Wiley Publications.

   2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly

   3. "Programming Ruby" The Pramatic Progammers guide by Dabve Thomas Second edition

REFERENCE BOOKS:

   1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B. Ware (Addison Wesley) Pearson Education.

   2. Perl by Example, E. Quigley, Pearson Education.

   3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O'Reilly, SPD.

   4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.

   5. Perl Power, J. P. Flynt, Cengage Learning.

MOBILE APPLICATION DEVELOPMENT LAB (Professional Elective – III)
(Course Code: CS664PE)

B.Tech. III Year II Sem.   **L T P C**

                                                                      0 0 2 1

Prerequisites: --- NIL---

Course Objectives:

- To learn how to develop Applications in an android environment.

- To learn how to develop user interface applications.

- To learn how to develop URL related applications.

Course Outcomes:

- Understand the working of Android OS Practically.

- Develop user interfaces.

- Develop, deploy and maintain the Android Applications.

LIST OF EXPERIMENTS:

1. Create an Android application that shows Hello + name of the user and run it on an emulator.

(b) Create an application that takes the name from a text box and shows hello message along with the name entered in the text box, when the user clicks the OK button.

2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Datepicker), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.

3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on the right fragment instead of the second screen with the back button. Use Fragment transactions and Rotation event listeners.

4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.

5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.

6. Create an application that uses a text file to store usernames and passwords (tab separated fields and one record per line). When the user submits a login name and password through a

screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with a Login Failed message.

7. Create a user registration application that stores the user details in a database table.

8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.

9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.

10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.

11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.

12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.

TEXT BOOKS:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012.
2. Android Application Development for Java Programmers, James C Sheusi, Cengage, 2013.

REFERENCE BOOK:

1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013.

SOFTWARE TESTING METHODOLOGIES LAB (Professional Elective – III)
(Course Code: CS665PE)

B.Tech. III Year II Sem.   **L T P C**

                                                                 0 0 2 1

Prerequisites

- A basic knowledge of programming.

Course Objectives

- To provide knowledge of software testing methods.
- To develop skills in automation of software testing and software test automation management using the latest tools.

Course Outcomes

- Design and develop the best test strategies in accordance with the development model.
- Design and develop GUI, Bitmap and database checkpoints
- Develop database checkpoints for different checks
- Perform batch testing with and without parameter passing

List of Experiments

1. Recording in context sensitive mode and analog mode

2. GUI checkpoint for single property

3. GUI checkpoint for single object/window

4. GUI checkpoint for multiple objects

5.

   a. Bitmap checkpoint for object/window

   b. Bitmap checkpoint for screen area

6. Database checkpoint for Default check

6. Database checkpoint for custom check

6. Database checkpoint for runtime record check

   a. Data driven test for dynamic test data submission

   b. Data driven test through flat files

   c. Data driven test through front grids

   d. Data driven test through excel test

   a. Batch testing without parameter passing

b. Batch testing with parameter passing

11. Data driven batch

11. Silent mode test execution without any interruption

12. Test case for calculator in windows application

TEXT BOOKS

1. Software Testing techniques, Baris Beizer, 2nd Edition, Dreamtech.

2. Software Testing Tools, Dr. K.V.K.K.Prasad, Dreamtech.

REFERENCE BOOKS

1. The craft of software testing, Brian Marick, Pearson Education.

2. Software Testing Techniques – SPD(Oreille)

3. Software Testing in the Real World, Edward Kit, Pearson.

4. Effective methods of Software Testing, Perry, John Wiley.

5. Art of Software Testing, Meyers, John Wiley.

BIG DATA-SPARK

(Course Code: CS623PC)

B.Tech. III Year II Sem. **L T P C**

0 0 4 2

Course Objectives:

- The main objective of the course is to process Big Data with advance architecture like spark and streaming data in Spark

Course Outcomes:

- Develop MapReduce Programs to analyze large dataset Using Hadoop and Spark
- Write Hive queries to analyze large dataset Outline the Spark Ecosystem and its components
- Perform the filter, count, distinct, map, flatMap RDD Operations in Spark.
- Build Queries using Spark SQL
- Apply Spark joins on Sample Data Sets
- Make use of sqoop to import and export data from hadoop to database and vice-versa

List of Experiments:

1. To Study of Big Data Analytics and Hadoop Architecture

    (i) know the concept of big data architecture

    (ii) know the concept of Hadoop architecture

2. Loading DataSet in to HDFS for Spark Analysis

    Installation of Hadoop and cluster management

    (i) Installing Hadoop single node cluster in ubuntu environment

    (ii) Knowing the differencing between single node clusters and multi-node clusters

    (iii) Accessing WEB-UI and the port number

    (iv) Installing and accessing the environments such as hive and sqoop

3. File management tasks & Basic linux commands

    (i) Creating a directory in HDFS

    (ii) Moving forth and back to directories

    (iii) Listing directory contents

    (iv) Uploading and downloading a file in HDFS

    (v) Checking the contents of the file

    (vi) Copying and moving files

    (vii) Copying and moving files between local to HDFS environment

    (viii) Removing files and paths

(ix) Displaying few lines of a file

(x) Display the aggregate length of a file

(xi) Checking the permissions of a file

(xii) Zipping and unzipping the files with & without permission pasting it to a location

(xiii) Copy, Paste commands

4. Map-reducing

(i) Definition of Map-reduce

(ii) Its stages and terminologies

(iii) Word-count program to understand map-reduce (Mapper phase, Reducer phase, Driver code)

5. Implementing Matrix-Multiplication with Hadoop Map-reduce

6. Compute Average Salary and Total Salary by Gender for an Enterprise.

7. (i) Creating hive tables (External and internal)

(ii) Loading data to external hive tables from sql tables(or)Structured c.s.v using scoop

(iii) Performing operations like filterations and updations

(iv) Performing Join (inner, outer etc)

(v) Writing User defined function on hive tables

8. Create a sql table of employees Employee table with id,designation Salary table (salary ,dept id) Create external table in hive with similar schema of above tables,Move data to hive using scoop and load the contents into tables,filter a new table and write a UDF to encrypt the table with AES-algorithm, Decrypt it with key to show contents

9. (i) Pyspark Definition(Apache Pyspark) and difference between Pyspark, Scala, pandas

(ii) Pyspark files and class methods

(iii) get(file name)

(iv) get root directory()

10. Pyspark -RDD'S

(i) what is RDD's?

(ii) ways to Create RDD

(iii) parallelized collections

(iv) external dataset

(v) existing RDD's

(vi) Spark RDD's operations (Count, foreach(), Collect, join,Cache())

11. Perform pyspark transformations

(i) map and flatMap

(ii) to remove the words, which are not necessary to analyze this text.

(iii) groupBy

(iv) What if we want to calculate how many times each word is coming in corpus ?

(v) How do I perform a task (say count the words 'spark' and 'apache' in rdd3) separatly

on each partition and get the output of the task performed in these partition ?

(vi) unions of RDD

(vii) join two pairs of RDD Based upon their key

12. Pyspark sparkconf-Attributes and applications

(i) What is Pyspark spark conf ()

(ii) Using spark conf create a spark session to write a dataframe to read details in a c.s.v

and later move that c.s.v to another location

TEXT BOOKS:

1. Spark in Action, Marko Bonaci and Petar Zecevic, Manning.

2. PySpark SQL Recipes: With HiveQL, Dataframe and Graphframes, Raju Kumar

Mishra and Sundar Rajan Raman, Apress Media.

WEB LINKS:

1. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01330150584451891225 1

8 2_shared/overview

2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01258388119638835242_

s hared/overview

3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01260526842300825616 9

2_shared/overview

ENVIRONMENTAL SCIENCE
(Course Code: *MC601)

B.Tech. III Year II Sem.   **L T P C**

                                                                3 0 0 0

Course Objectives:

- Understanding the importance of ecological balance for sustainable development.
- Understanding the impacts of developmental activities and mitigation measures.
- Understanding the environmental policies and regulations

Course Outcomes:

- Based on this course, the Engineering graduate will understand /evaluate / develop technologies on the basis of ecological principles and environmental regulations which in turn helps in sustainable development

UNIT - I

Ecosystems: Definition, Scope, and Importance of ecosystem. Classification, structure, and function of an ecosystem, Food chains, food webs, and ecological pyramids. Flow of energy, Biogeochemical cycles, Bioaccumulation, Biomagnification, ecosystem value, services and carrying capacity, Field visits.

UNIT - II

Natural Resources: Classification of Resources: Living and Non-Living resources, water resources: use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. Mineral resources: use and exploitation, environmental effects of extracting and using mineral resources, Land resources: Forest resources, Energy resources: growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

UNIT - III

Biodiversity and Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Ex-situ conservation. National Biodiversity act.

UNIT - IV

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, Air Pollution: Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. Water pollution: Sources and types of pollution, drinking water quality standards.

Soil Pollution: Sources and types, Impacts of modern agriculture, degradation of soil. Noise Pollution: Sources and Health hazards, standards, **Solid waste:** Municipal Solid Waste management, composition and characteristics of e-Waste and its management.

Pollution control technologies: Wastewater Treatment methods: Primary, secondary and Tertiary. Overview of air pollution control technologies, Concepts of bioremediation. Global Environmental

Issues and Global Efforts: Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol. NAPCC-GoI Initiatives.

UNIT - V

Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socioeconomical aspects. Strategies for risk assessment, Concepts of Environmental Management Plan (EMP). Towards Sustainable Future: Concept of Sustainable Development Goals, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

TEXT BOOKS:

1 Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.

2 Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:

1. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHL Learning Private Ltd. New Delhi.

2. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHI Learning Pvt. Ltd.

3. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.

4. Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.

5. Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.

6. Introduction to Environmental Science by Y. Anjaneyulu, BS. Publications.