VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

B. Tech (IT) -R22 SYLLABUS

III B.Tech (Semester-I)

COURSE STRUCTURE

III YEAR I SEMESTER

S. No.	Course Code	Course Title	Category	L	T	P	Credits
1	22IT3111	Software Engineering	PC	3	0	0	3
2	22CS3111	Design and Analysis of Algorithms	PC	3	0	0	3
3	22CS3113	Web Technologies	PC	3	0	0	3
4	**Professional Elective – I**		PE	3	0	0	3
	22IT3171	Principles of Programming Languages					
	22CS3171	Computer Graphics					
	22CS3172	Advanced Computer Architecture					
	22IT3172	Data Analytics					
5	**Professional Elective – II**		PE	3	0	0	3
	22IT3173	Distributed Systems					
	22DS3174	Artificial Intelligence					
	22IT3175	Distributed Databases					
	22IT3176	Data Warehousing and Data Mining					
6	22IT3151	Software Engineering and R Programming Lab	PC	0	0	3	1.5
7	22IT3152	Web Technologies Lab	PC	0	0	3	1.5
8	22IT3153	Skill Development Course-UI Design Flutter	PC	0	0	2	1
9	22HS3151	Advanced English Communication Skills Lab	HS	0	0	2	1
10	22MC0005	Intellectual Property Rights	MC	3	0	0	0
Total				**18**	**0**	**10**	**20**

III YEAR II SEMESTER

S. No.	Course Code	Course Title	Category	L	T	P	Credits
1	22IT3211	Automata Theory and Compiler Design	PC	3	0	0	3
2	22IT3212	Data Communications and Computer Networks	PC	3	0	0	3
3	22CS3211	Machine Learning	PC	3	0	0	3
4	**Professional Elective –III**		PE	3	0	0	3
	22IT3271	BlockChain Technology					
	22AM3271	Cloud Computing					
	22IT3272	Computer Vision					
	22IT3273	Embedded Systems					
5	**Open Elective-I**		OE	3	0	0	3
6	22IT3251	Compiler Design Lab	PC	0	0	2	1
7	22IT3252	Data Communications and Computer Networks Lab	PC	0	0	2	1
8	22IT3253	Machine Learning Lab	PC	0	0	2	1
9	22IT3281	Industry Oriented Mini Project	PW	0	0	4	2
10	22MC0002	Environmental Science	MC	3	0	0	0
Total				**18**	**0**	**10**	**20**

Environmental Science in III Yr II Semester Should be registered by Lateral Entry Students Only.

22IT3111: SOFTWARE ENGINEERING

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

Prerequisites:

- C Programming

Course Objectives:

1. To understand fundamental principles of Software engineering, and their application in the development of software products.
2. To understand and create the software requirements specifications document.
3. To understand and use unified modeling language for specifying, analysis and designing.
4. To understand testing strategies for testing software applications
5. To understand Software metrics, Risk Management strategies and software process improvement.

Course Outcomes

1. Able to apply the software engineering lifecycle phases communication, planning, analysis, design, construction, and deployment.
2. Ability to translate end-user requirements into system and software requirements into Software Requirements specification Document (SRS).
3. Able to apply UML in object-oriented software modeling to develop computer software.
4. Able to identify problems in software and will be able to develop a simple testing report.
5. Able to apply Software Metrics to mitigate risks thereby improving software process.

UNIT I

Introduction: The evolving role of software, Changing Nature of Software, legacy software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models,

Process Models: The waterfall model, Incremental process models, Evolutionary process models, specialized process models, The Unified process,

Agile development- Agile process, Extreme Programming, scrum, dynamic systems development method, agile model, Agile Unified Process

UNIT II

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management, Software requirements documents.

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

UNIT III

Modeling Techniques using UML: The Unified Approach to Modeling, Structural and Behavioral Diagrams.

Design Engineering: Data Flow Diagrams, Design process and Design quality, Design concepts, the design model, pattern-based software design.

Creating an architectural design: Architectural styles and patterns, Architectural Design, assessing alternative architectural designs.

UNIT IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Regression Testing, Unit Testing, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Quality Management: Quality concepts, software quality assurance, software reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

UNIT V

Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Text Books:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 8th edition, Mc Graw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

References:

1. "Software Engineering", Ian Sommerville, Addison-Wesley, 9th Edition, 2010, ISBN- 13:978-0137035151.
2. Richard Fairley, "Software Engineering Concepts", Tata McGraw Hill.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.
4. Software testing techniques by Boris Beizer, dreamtech.
5. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
6. Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies.

22CS3111: DESIGN AND ANALYSIS OF ALGORITHMS

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

Prerequisites:

1. A course on "C Programming"
2. A course on "Data Structures"

Course Objectives:

1. To analyse performance of algorithms
2. To understand and choose the appropriate algorithm design technique for a specified application
3. To solve problems using algorithm design techniques such as the greedy method, divide and conquer, dynamic programming, backtracking and branch and bound.
4. To analyse the impact of algorithm design techniques on each application solved
5. To introduce and understand P and NP classes.

Course Outcomes: After the completion of the course student will be able to

1. Analyse the different algorithm design techniques for a given problem.
2. Design algorithms for various computing problems.
3. Argue the correctness of algorithms using inductive proofs and invariants.
4. Analyze the limitations of algorithms.
5. Explain about coping with the limitations of algorithms.

UNIT - I

Notation of an Algorithm: Fundamentals of Algorithmic Problem Solving, Fundamentals of the Analysis of Algorithm Efficiency–Order Notations and its properties, Mathematical analysis for Recursive -Towers of Hanoi and Non-recursive algorithms

Divide and conquer- General method-Control abstraction, Solving Recurrence Relation using Substitution method and Master's Theorem, applications - Binary search, Merge sort, Quick sort, Strassen's Matrix Multiplication, Finding Maximum and Minimum element.

UNIT - II

Greedy Method- General method-Control abstraction, applications- Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Single source shortest path problem.

UNIT - III

Dynamic Programming: General Method, applications-Multi Stage Graphs, Chained matrix multiplication, All pairs shortest path problem, Optimal binary search trees, 0/1 knapsack problem, Reliability design, Traveling sales person problem.

UNIT - IV

Backtracking: General method-Control abstraction, applications-The 8-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT-V

Branch and Bound: General Method-Control abstraction, applications-15-Puzzle Problem - LC search,0/1 Knapsack problem-LC Branch and Bound solution, FIFO Branch and Bound solution, Travelling sales person problem.

NP-Hard and NP-Complete problems: Basic concepts, Non-deterministic algorithms, NP – Hard and NP- Complete classes, Cook's theorem- proof of reduction.

TEXT BOOKS:

1. Ellis Horowitz, SatrajSahni and S Rajasekharam, Fundamentals of Computer Algorithms, Galgotia publishers
2. M.T. Goodrich, Robert Tamassia, Algorithm design: Foundations, Analysis and Internet examples,Wiley student Edn, John Wiley &sons.
3. Parag Himanshu Dave,Himanshu Bhalchandra Dave, Design and Analysis algorithms PearsonPublication.

REFERENCES:

1. Allen Weiss, Data structures and Algorithm Analysis in C++, 2nd Edn, Pearson Education
2. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI Learning Private Limited.
3. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education.

22CS3113: WEB TECHNOLOGIES

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

Prerequisites :

- A Course on " Object oriented programming through Java"

Course Objectives:

1. To introduce Client-side scripting with HTML and java script
2. To introduce Server-side programming with java servlets
3. To introduce Server-side programming with JSP
4. To introduce Struts framework
5. To introduce PHP language for Server side scripting

Course Outcomes: After completion of the course, the student will be able to:

1. Understand basics of HTML, CSS, Design and Development of Web pages (using Validations) with Java Script
2. Develop Server-side Applications with Servlets (Sessions and Cookies).
3. Create JSP pages with Database Server.
4. Create application development using struts.
5. Understand Server-side Scripting with PHP language.

UNIT -I

HTML Common tags- List, Tables, images, frames, divisions, forms; Cascading Style sheets;
XML: Introduction to XML, Defining XML tags, their attributes and values, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

UNIT –II

Client-side Scripting: Introduction to JavaScript, Javascript language – declaring variables, scope of variables, functions, Event handlers (onclick, onsubmit etc.), Document Object Model, Form validation.

UNIT -III

Introduction to Servlets: Common Gateway Interface (CGI), Lifecycle of a Servlets, deploying a Servlets, The Servlets API, Reading Servlets parameters, Reading initialization parameters, Handling Http Request & Responses, Using Cookies and sessions, connecting to a database using JDBC.

UNIT -IV

Introduction to JSP:

The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Session tracking in JSP, connecting to database in JSP. Struts framework, application development using struts.

UNIT-V

Introduction to PHP:

Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies

File Handling in PHP:

File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

TEXT BOOKS

1. Web Technologies, Uttam K Roy, Oxford University Press
2. Web Programming, building internet applications, Chris Bates 2nd edition,WILEY Dreamtech
3. The complete Reference Java 2 Fifth Edition by Patrick Naughton and Herbert Schildt. TMH
4. Java Server Pages –Hans Bergsten, SPD O'Reilly

REFERENCES:

1. Web Programming, building internet applications, Chris Bates 2nd edition, Wiley DreamTech.
2. Java Server Pages –Hans Bergsten, SPD O'Reilly
3. Java Script, D. Flanagan, O'Reilly, SPD.
4. Beginning Web Programming-Jon Duckett WROX.
5. Programming World Wide Web, R. W. Sebesta, Fourth Edition, Pearson.
6. Internet and World Wide Web – How to program, Dietel and Nieto, Pearson

22IT3171: PRINCIPLES OF PROGRAMMING LANGUAGES
(Professional Elective - I)

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

1. Discuss the background for choosing appropriate programming languages for certain Classes of programming problems
2. Explain how to solve the principle to program in an imperative (or procedural), an Object-oriented, a functional, and a logical programming language
3. Recognize Increase the capacity to express programming concepts and choose among alternative ways to express things.
4. Discuss principle to design a new programming language.
5. Explain the use of debuggers and related tools.

Course Outcomes

1. Acquire the skills for expressing syntax and semantics in formal notation
2. Identify and apply a suitable programming paradigm for a given computing application
3. Gain knowledge of and able to compare the features of various programming languages
4. Illustrate with language abstraction constructs of classes, interfaces, packages, and procedures.
5. Demonstrate how to design and construct with using functional languages, be exposed to using logic languages.

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 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

22CS3171: COMPUTER GRAPHICS
(Professional Elective-I)

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

Prerequisites:

1. A Course on "C Programming"
2. A Course on "Data Structures "

Course Objectives

1. To make students understand about the fundamental concepts of computer.
    a. graphics and explore line , circle drawing algorithms.
2. To make students understand about 2D, 3D geometrical transformations.
3. To make students understand about line & polygon clipping.
4. To make students understand about polygon rendering methods.
5. To make students understand about computer animation.

Course Outcomes

1. Be able to implement techniques for efficiently rendering straight line , circle & ellipse
2. Acquire familiarity with the relevant mathematics of computer graphics : translation,
    a. scaling, rotation.
3. Be able to design applications that display graphic images to given specifications.
4. Implement 3-D geometric transformation and 3-D viewing.
5. Apply Computer animation.

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 (Bresenham's and DDA Algorithm), midpoint circle and ellipse algorithms.

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

UNIT-II

2-D geometrical transforms: 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, Cohen-Sutherland algorithms, 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, Basic illumination models, polygon rendering methods.

UNIT-IV

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

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

UNIT-V

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

Visible surface detection methods:

Classification, back-face detection, depth-buffer, BSP-tree methods and area sub-division methods.

TEXT BOOKS:

1. "Computer Graphics C version", Donald Hearn and M. Pauline Baker, Pearson Education.
2. "Computer Graphics Principles & practice", second edition in C, Foley, Van Dam, Feiner and Hughes, Pearson Education.
3. Computer Graphics, Steven Harrington, TMH.

REFERENCES:

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.

22CS3172: ADVANCED COMPUTER ARCHITECTURE
(Professional Elective-I)

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

Prerequisites:
A course on Computer Organization

Course Objectives
1. To impart the concepts and principles of parallel and advanced computer architectures.
2. To develop the design techniques of Scalable and multithreaded Architectures.
3. To apply the concepts and techniques of parallel and advanced computer architectures to
    a. design modern computer systems.
4. To understand the functions of Bus Cache.
5. To study and analyze fundamental issues in architecture design and their impact on performance.

Course Outcomes: Gain knowledge of
1. Computational models and Computer Architectures.
2. Concepts of parallel computer models.
3. Scalable Architectures, Pipelining, Superscalar processors, multiprocessors.
4. Bus Cache functions and Shared Memory functions.
5. Various principles of Scalability Performance.

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, Memory Hierarchy Technology, Virtual Memory Technology.

UNIT - III
Bus Cache and Shared memory: Backplane bus systems, Cache Memory organizations, 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 Multicomputer, Multiprocessor system interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputer, Message-passing Mechanisms, Multivector and SIMD computers, Vector Processing Principals, Multivector Multiprocessors, Compound Vector processing, SIMD computer Organizations, The connection machine CM-5.

UNIT - V
Scalable: Multithreaded and Dataflow Architectures, Latency-hiding techniques, Principles of Multithreading, Fine-Grain Multicomputer, Scalable and multithreaded Architectures, Dataflow and hybrid Architectures.

TEXT BOOK:

1. Advanced Computer Architecture Second Edition, Kai Hwang, Tata McGraw Hill Publishers.
2. John L. Hennessy & David A. Patterson Morgan Kufmann, "Computer Architecture A Quantitative Approach", 3rd Edition, An Imprint of Elsevier, 2011

REFERENCE BOOKS:

1. Computer Architecture, Fourth edition, J. L. Hennessy and D.A. Patterson. ELSEVIER.
2. Andrew S. Tanenbaum, Structured Computer Organization, Prentice Hall, 6th edition, 2012, ISBN: 978-0132916523.
3. C. Hamacher, Z. Vranesic and S. Zaky, Computer Organization, McGraw-Hill, 5th edition,2002, ISBN: 0072320869.

22IT3172: DATA ANALYTICS
(Professional Elective-I)

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

Prerequisites

- Data Base Management Systems, Computer Oriented Statistical Methods

Course Objectives

1. To explore the fundamental concepts of data analytics and understand various Data Sources
2. Understand several key big data technologies used for storage, analysis and manipulation of data.
3. To learn the principles and methods of statistical analysis.
4. Understand Regression, supervised algorithms to perform data analytics.
5. Understand various visualization techniques.

Course Outcomes

1. Identify the various sources of Big Data.
2. Understand big data technologies and the impact of data analytics for business decisions and strategy.
3. Apply and analyze various regression techniques.
4. Outline various Time series methods to discover interesting patterns
5. To carry out standard data visualization and formal inference procedures

UNIT - 1

Data Management: Design Data Architecture and manage the data for analysis, understand various sources of Data like Sensors/Signals/GPS etc. Data Management, Exploratory data analysis, Data pre-processing, Missing Values - Outlier Detection and Treatment.

UNIT- 2

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. Introduction to HADOOP: Big Data, HDFS, Apache Hadoop, MapReduce.

UNIT - 3

Regression – Concepts, Blue property assumptions, Least Square Estimation, Variable Rationalization, Modeling Process – Training model – Validating model – Predicting new observations. Logistic Regression: Model Theory, Model fit Statistics, Model Construction, Analytics applications to various Business Domains etc.

UNIT - 4

Object Segmentation: Regression Vs Segmentation-Supervised and Unsupervised Learning, Tree Building – Regression, Classification, over fitting, Pruning and Complexity.

Time Series Methods: Arima, Measures of Forecast Accuracy, STL approach, Data Serialization, Data Extraction and Analyze for prediction

UNIT - 5

Data Visualization: Introduction to Data Visualization, Data visualization options, Data visualization Techniques, Visualizing Complex Data and Relations, Filters – Dashboard development tools.

TEXT BOOKS:

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

REFERENCES:

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.
4. Michael Minelli, Michele Chambers, Ambiga Dhiraj ,—Big Data, Big Analytics: EmergingBusiness Intelligence and Analytic Trends‖, John Wiley & Sons, 2013.
5. Bart Baesens, "Analytics in a Big Data World: The Essential Guide to Data Science and itsApplications", John Wiley & Sons, 2014

22IT3173: DISTRIBUTED SYSTEMS
(Professional Elective-II)

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

Prerequisites:

1. A course on "Operating Systems".
2. A course on "Computer Organization & Architecture".

Course Objectives:

1. Understand the basic concepts of Distributed system and sharing of resources in a distributed manner.
2. Familiarize the basics of Distributed systems.
3. Demonstrate the concepts of IPC, group communication and RPC.
4. Describe the theoretical concepts, namely, virtual time, agreement and consensus protocols.
5. Understand the concepts of Transaction in Distributed Environment, Concurrency control, Deadlocks and Error recovery.

Course Outcomes: After the completion of the course student should be able to

1. Characterize the Distributed Systems.
2. Know the support of Operating System like Operating system architecture, Protection, Communication and Invocation and architecture of file service.
3. Understand peer to peer systems and applications with case studies.
4. Understand Transactions and Concurrency control.
5. Understand Security issues like Transactions with replicated data.

UNIT - I

Characterization of Distributed Systems-Introduction, Examples of Distributed systems, Resource sharing and web, challenges, System models - Introduction, Architectural and Fundamental models, Networking and Internetworking, Inter - process Communication, Distributed objects and Remote Invocation - Introduction, Communication between distributed objects, RPC, Events and notifications, Case study - Java RMI.

UNIT - II

Operating System Support - Introduction, OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture, Distributed File Systems - Introduction, File Service architecture.

UNIT - III

Peer to Peer Systems - Introduction, Napster and its legacy, Peer to Peer middleware, Routing overlays, Overlay case studies-Pastry, Tapestry, Application case studies - Squirrel, OceanStore.

Time and Global States - Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging.

Coordination and Agreement - Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

UNIT - IV

Transactions and Concurrency control-Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering.

Distributed Transactions- Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery.

UNIT - V

Replication-Introduction, System model and group communication, Fault tolerant services, Transactions with replicated data. Distributed shared memory, Design and Implementation issues, and Consistency models.

TEXT BOOKS:

1. Distributed Systems Concepts and Design, G Coulouris, J Dollimore and T Kindberg, 4th Edition, Pearson Education.
2. Distributed Systems, S.Ghosh, Chapman & Hall/ CRC, Taylor & Francis Group, 2010.

REFERENCE BOOKS:

1. Distributed Systems - Principles and Paradigms, A.S. Tannenbaum and M.V. Steen, Pearson Education.
2. Distributed Computing, Principles, Algorithms and Systems, Ajay D. Kshemakalyani and Mukesh Singhal, Cambridge, rp 2010.

22DS3174: ARTIFICIAL INTELLIGENCE
(Professional Elective-II)

B.Tech. III Year I Sem

L	T	P	C
3	-	-	3

Course Objectives:

1. Understand the difference between various intelligent agents and environments including solving problems by searching the solution space.
2. Understand adversarial search and propositional logic to find the solutions of constraint satisfaction problems.
3. Reference using first order logic and describe knowledge representation.
4. Design solutions to a problem in the real world environment
5. Learn to infer in uncertain domains using probabilistic learning models.

Course Outcomes: Differentiate various intelligent agents and environments.

1. Also solve problems by searching the solution space.
2. Use adversarial search and propositional logic to solve constraint satisfaction problems
3. Use first order logic to infer and describe knowledge representation
4. Plan solutions for problems in the real world environment.
5. Infer in uncertain domains using probabilistic learning models

UNIT - I:

Problem Solving by Search-I & II 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.

UNIT – II

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. **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, Back- ward Chaining, Resolution.

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

UNIT - IV:

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

Planning and Acting in the Real World: Time, Schedules, and Resources, Hierarchical Planning, Planning and Acting in Nondeterministic Domains, Multi agent planning.

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, Stuart Russell and Peter Norvig, 4th Edition, Pearson Education, 2020.

REFERENCE BOOKS:

1. Artificial Intelligence, E.Rich and K.Knight, , 3rd Edition, TMH, 2009.
2. Artificial Intelligence, Patrick Henny Winston, 3rd Edition, Pearson Education, 2015.
3. Artificial Intelligence, ShivaniGoel, Pearson Education, 2013. .
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education, 2005

22IT3175: DISTRIBUTED DATABASES
(PROFESSIONAL ELECTIVE-II)

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

Prerequisites

- A course on "Database Management Systems"

Course Objectives:

- The purpose of the course is to enrich the previous knowledge of database systems and
- Exposing the need for distributed database technology to confront with 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 system.
- Understand the design aspects of object-oriented database system and relate development.
- Able to Practice Parallel distributed databases.
- Identify the differences between OODBMS and ORDBMS.

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.

References:

1. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: "Database Systems: The Complete Book", Second Edition, Pearson International Edition
2. Chanda Ray (2012), Distributed Database Systems, 1st Edition, Pearson Education India.

(Professional Elective-II)

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

Prerequisites:

- Should have knowledge of database systems

Course Objectives:

1. To describe the concepts related to data warehousing, On-Line Analytical Processing (OLAP).
2. To gain knowledge about the steps involved in Knowledge Data Discovery.
3. To study the performance of algorithms for Association Rules.
4. To examine classification algorithms and assess prediction techniques.
5. To describe methods for data-clustering approaches.

Course Outcomes:

1. Construct Multidimensional data models to represent data cubes and perform characterization and generalization tasks on data cubes.
2. Ability to identify the need and importance of preprocessing techniques.
3. Compute associations and correlations among items by mining frequent patterns from transactional databases.
4. Build model to classify unknown data objects.
5. Build clusters using various clustering techniques and assess clusters formed.

Unit 1:

Data Warehouse and OLAP

Introduction to Data Warehouse, Data Warehouse Architecture, OLAP, OLTP, OLAP Servers-(ROLAP, MOLAP, HOLAP)

Multidimensional Data Model: Data cube, Efficient methods for Data cube computation, schemas, OLAP Operations.

Unit 2:

Introduction to data mining and its issues

Data, Types of data, Need for data mining, Transactional databases, Data Mining functionalities, Applications, Classification of data mining systems, Data Mining Task Primitives, Major issues in Data Mining, KDD process.

Data Preprocessing: Data Cleaning, Data Integration and Transformation Data Reduction, Discretization and Concept Hierarchy Generation, measures of similarity and dissimilarity-basics.

Unit 3 :

Mining Association rules in large databases:

Mining Frequent Patterns, Associations and Correlations: Market Basket Analysis, Association rule mining, Mining Frequent Item sets-Apriori algorithm, compact representation of frequent item set-maximal frequent item set, closed frequent item set, FP-growth algorithms,

Unit 4

Classification and Prediction: Basic concepts, Decision tree induction, Bayesian classification, Naive Bayes Classification, classification by Back propagation,Lazy learners, other classification methods, Prediction.

Unit 5

Clustering: Types of Data in Cluster Analysis, Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, DBSCAN - Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses, Outlier Analysis.

Text Books:

[1] Data Mining – Concepts and Techniques - Jiawei Han & Micheline Kamber, Third Edition, Elsevier.
[2] Pang-Ning Tan & Michael Steinbach, ―Introduction to Data Mining‖,Vipin Kumar, Pearson
[3] Data Warehousing, Data Mining &OLAP- Alex Berson and Stephen J. Smith- Tata McGraw-Hill Edition, Tenth reprint 2007

Reference Books:

[1] Data Mining Introductory and Advanced topics–Margaret H Dunham, Pearson Education.
[2] Arun K Pujari, Data Mining Techniques, (2017) ,University Press.
[3] Mohammed J. Zaki, Wagner Meira, Jr ,‖Data Mining and Analysis - Fundamental Concepts and Algorithms‖, Oxford.

22IT3151: SOFTWARE ENGINEERING AND R PROGRAMMING LAB

III B. Tech I Sem.

L	T	P	C
0	0	3	1.5

Prerequisites:

- Basic Knowledge on C programming

Course Objectives:

1. To understand fundamental principles of Software engineering, and their application in the development of software products.
2. To understand and create the software requirements specifications document.
3. To understand and use unified modelling language for specifying, analysis and designing.
4. To understand testing strategies for testing software applications
5. To understand Software metrics and Risk Management strategies to identify potential problems before they occur.
6. Effective use of Business Intelligence (BI) technology (Tableau) to apply data visualization
7. To discern patterns and relationships in the data

Course Outcomes:

1. Able to recognize the software engineering lifecycle phases.
2. Ability to determine end-user requirements and software requirements and translate them into Software Requirements specification Document (SRS)
3. Able to select an appropriate architectural model with design engineering.
4. Able to assess problems in software and to write a simple testing report.
5. Able to determine Software Metrics, potential risk and how to manage them though RMMM plan.
6. Understand Tableau concepts of Dimensions and Measures.
7. Develop Programs and understand how to map Visual Layouts and Graphical Properties.

List of Experiments for Software Engineering:

Do the following 6 exercises for any two projects given in the list of sample projects or any other projects using smart draw, Rational Rose or Star UML for UML diagrams:

1) Development of problem statement.
2) Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3) Preparation of Software Configuration Management and Risk Management related documents.
4) Draw level 0, level 1, and level 2 dataflow diagrams
5) Study and usage of any Design phase CASE tool .
6) Performing the Design by using any Design phase CASE tools.
7) Develop test cases for unit testing and integration testing.
8) Develop test cases for various white box and black box testing techniques.

Sample Projects:

1) Hospital management system
2) Online mobile recharge portal
3) Online Exam Registration
4) Stock Maintenance System
5) Online course reservation system
6) E-ticketing

List of Experiments for R Programming:

1. Write an R-Program to take input from user.
2. Write an R Program to Find the Fibonacci sequence Using Recursive Function
3. Write an R-Program to demonstrate working with operators.
4. Write an R Program to Check if a Number is Odd or Even
5. Write an R Program to check if the given Number is a Prime Number
6. Write an R Program to Find the Factorial of a Number
7. Write an R Program to Find L.C.M of two numbers
8. Write an R Program to create a Vector and to access elements in a Vector
9. Write an R Program to create a Matrix and access rows and columns using functions colnames() and rownames()
10. Write an R Program to create a List and modify its components.
11. Write an R Program to create a Data Frame.
12. Write an R Program to access a Data Frame like
    i) List ii) Matrix

Text Books:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, and Pearson Education.
4. Microsoft Power BI cookbook, Brett Powell, 2nd edition.
5. R Programming for Data Science by Roger D. Peng.

References:

1 "Software Engineering", Ian Sommerville, Addison-Wesley, 9th Edition, 2010, ISBN- 13: 978-0137035151.
2 Richard Fairley, "Software Engineering Concepts", Tata McGraw Hill.
3. The Art of R Programming by Norman Matloff Cengage Learning India

22IT3152: WEB TECHNOLOGIES LAB

III B. Tech I Sem.

L	T	P	C
0	0	3	1.5

Course Objectives:

1. To develop an ability to design and implement static and dynamic website.
2. Use JavaScript for dynamic effects.
3. Understand, analyze and create XML documents and XML Schema.
4. Understand, analyze and build web applications using PHP.
5. Handling Cookies and Sessions using PHP, SERVLETS and JSP.

Course Outcomes:

1. Simple Applications with Technologies like HTML, JavaScript .
2. Parse XML Files using JAVA (DOM AND SAX Parsers).
3. Use Tomcat Server to Develop Servlet Applications and connect to Database.
4. Develop JSP Applications using Tomcat Server and connect to Database.
5. Design web application using PHP.

EXPERIMENTS:

List of Experiments

1. Write an HTML code to demonstrate

   a) Lists b) Tables (row span and col span) c) Cascading Style Sheets

2. Design a web page to demonstrate

   a) Divisions b) Frames c) Embedding Images

3. Develop static pages (use Only HTML) of an online book store. The pages should resemble: www.amazon.com. The website should consist the following pages.

   a) Home page
   b) Registration and user Login
   c) User Profile Page
   d) Books catalog
   e) Shopping Cart
   f) Payment By credit card
   g) Order Conformation.

4. Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).

5. Write a JavaScript program to validate the registration form contents with the following Rules (Use RegExp Object)

a) Username Must starts with Uppercase followed by set of lowercase letters or digits.
b) Password must contain only uppercase letters and length must be in between 8 to12.
c) Phone number contains 10 digits.
d) E-mail must follow some predefined format (example@domain.com)

6. Build a simple application on A) Angular JS B) Node.js

7. Create an XML document that contains 10 user's information. Write a Java Program, which takes User Id as input and returns the user details by taking the user information from XML document using DOM parser

8. Install the following on the local machine

a) Apache Tomcat Web Server

b) Install MySQL/Oracle (if not installed)

C) Install PHP and configure it to work with Apache web server and MySQL

9. a) Write a Servlet program to read the parameters from user interface and display Welcome message.

b) Write a Servlet program to read initialization parameters using ServletConfig and Servlet Context object.

10. Write Servlet programs to work with the following session tracking technique.
a) Http Session

11. Develop a dynamic web page which contains Registration and Login Forms using servlet with Oracle database. Validate the login page.

12. Write a JSP Program to handle the exceptions.

13. Develop a dynamic web page which contains Registration and Login Forms using JSP with Oracle database. Validate the login page.

14. Write a PHP script that reads data from one file and write into another file.

15. Develop a dynamic web page which contains Registration and Login Forms in PHP with MySQL database. Validate the login page.

Text Books:

1. WEB TECHNOLOGIES: A Computer Science Perspective, Jeffrey C. Jackson, Pearson Education

References:

1. Deitel H.M. and Deitel P.J., "Internet and World Wide Web How to program", Pearson International, 2012, 4th Edition.

2. J2EE: The complete Reference By James Keogh,McGraw-Hill

3. Bai and Ekedhi, The Web Warrior Guide to Web Programming,Thomson

4. Paul Dietel and Harvey Deitel," Java How to Program", Prentice Hall of India, 8thEdition

5. Web technologies, Black Book, Dreamtech press

22IT3153: Skill Development Course (UI design- Flutter)

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

0 0 2 1

Prerequisites:

- Basic programming experience (e.g., Java, Python).
- Familiarity with object-oriented programming concepts.

Course Objectives:

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

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

WEEK 1.
a) Install Flutter and Dart SDK.
b) Write a simple Dart program to understand the language basics.

WEEK 2.
a) Explore various Flutter widgets (Text, Image, Container, etc.).
b) Implement different layout structures using Row, Column, and Stack widgets.

WEEK 3.
a) Design a responsive UI that adapts to different screen sizes.
b) Implement media queries and breakpoints for responsiveness.

WEEK 4.
a) Set up navigation between different screens using Navigator.
b) Implement navigation with named routes.

WEEK 5
a) Learn about stateful and stateless widgets.
b) Implement state management using set State and Provider.

WEEK 6.
a) Create custom widgets for specific UI elements.
b) Apply styling using themes and custom styles.

WEEK 7.
a) Design a form with various input fields.
b) Implement form validation and error handling.

WEEK 8

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

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

WEEK 9.

a) Fetch data from a REST API.

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

WEEK 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

22HS3151: ADVANCED ENGLISH COMMUNICATIONS SKILLS LAB

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

0 0 2 1

Course Objectives

This lab focuses on using Multi-media instruction as well as stimulating peer group activities for language development to meet the following targets:

- To improve students fluency in spoken English.
- To enable them to listen to English spoken at normal conversational speed.
- To help students develop their vocabulary.
- To read and comprehend texts in different contexts.
- To communicate their ideas relevantly and coherently in writing.

Course Outcomes: After the completion of the course student will be able to

- Acquire vocabulary and Grammar and use them contextually.
- Listen and speak effectively, and present themselves effectively.
- Develop proficiency in academic reading and writing.
- Communicate confidently in formal and informal contexts.
- Increase their job opportunities.

Syllabus

The following course activities will be conducted as part of the Advanced English Communication Skills (AECS) Lab:

Unit I

Vocabulary and Grammar: Vocabulary Building – Word Formation: Prefixes and Suffixes - Synonyms, and Antonyms, One-word Substitutes, Idioms, Phrases, Collocations, and Compound Words.

Grammar – Articles, Prepositions, Tenses, Subject-Verb Agreement, Voice and Speech-Spotting Errors - Correction of Sentences,

Unit II

Advanced Reading Comprehension: Argumentative Analysis of (with reference to) GRE, TOEFL, IELTS – Jumbled Sentences and Sentence Completion.

Unit III

Writing Skills– Structure and Different Types of Writings – Argumentative Writing – Letter Writing - Resume Writing - Technical Report Writing

Creating and Using LinkedIn Profile - Netiquette - Statement of Purpose (SOP) - Letter of Recommendation

Unit IV

Presentation Skills -_Oral Presentations (Group/Individual) and Written Presentations – PPTs/ Posters (Virtual/Offline) – Projects, Reports and Assignments - Introducing Oneself Virtually (Making a Video on Oneself and Analyzing it critically).

Unit V

Group Dynamics & Interviews: Group Discussion - Dos and Don'ts - Intervention, Summarizing, Modulation of Voice, Body Language, Relevance, Fluency and Organization of Ideas – Debate: Concept and Process - Difference between Group Discussions and Debates-Rubrics of Evaluation - Interviews and Types of Interviews - Pre-interview Planning, Opening Strategies, Answering Strategies - Introducing Self - Oral Interviews (face-to-face) –Virtual Interviews - Mock Interviews - Handling Technical Glitches.

REFERENCES

1. Kumar, Sanjay and Pushp Lata. English for Effective Communication, Oxford University Press, 2015.

2. Konal, Nira. English Language Laboratories- A Comprehensive Manual, PHI Learning Pvt. Ltd. 2011.

3. The Official Guide to the GRE General Test. Tamil Nadu: McGraw Hills Education (India) 3rd Edition, 2017.

22MC0005: INTELLECTUAL PROPERTY RIGHTS

III B.Tech I Semester

L	T	P	C
3	0	0	0

Course Objectives:

- To know the concept of intellectual property
- To study about trade marks
- To study about law of copyrights and law of patents.
- To impart the knowledge on trade secrets
- To know new developments in IPR laws at national and international level.

Course Outcomes: At the end of this course, students will demonstrate the ability to

- 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.
- Explain about trade secrets
- 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 transfe

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

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

B. Tech (IT) -R22 SYLLABUS

III B.Tech (Semester-II)

22IT3211: AUTOMATA THEORY AND COMPILER DESIGN

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

Prerequisites:
1. Digital logic design
2. Computer Organization

Course Objectives
1. To introduce the fundamental concepts of formal languages, grammars and automata theory.
2. To understand deterministic and non-deterministic machines and the differences between decidability and undecidability.
3. Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
4. Emphasize the concept of phases of compiler, parsing, syntax directed translation, type checking use of symbol tables.
5. To learn about Intermediate code generation.

Course Outcomes
1. Able to employ finite state machines for modeling and solving computing problems.
2. Able to design context free grammars for formal languages.
3. Able to distinguish between decidability and undecidability.
4. Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis and acquire skills in using lex tool and design LR parsers.
5. Determine the different Syntax directed translations and code generation techniques.

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

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

UNIT - III
Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state
Turing Machines:
Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine
Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines

UNIT - IV

Introduction: The structure of a compiler

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical- Analyzer Generator Lex

Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom- Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers

UNIT - V

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Syntax Directed Translation Schemes, Implementing L-Attributed SDD's.

Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code

Run-Time Environments: Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management

TEXT BOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3rd 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. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
2. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
3. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
4. Lex & YACC – John R. Levine, Tony Mason, Doug Brown, O'reilly
5. Compiler Construction, Kenneth C. Louden, Thomson. Course Technology.

22IT3212: DATA COMMUNICATION AND COMPUTER NETWORKS

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

Course Objectives:

1. To introduce the fundamental various types of computer networks.
2. To introduce the TCP/IP and OSI models with merits and demerits.
3. To explore the various layers of OSI Model.
4. To introduce UDP and TCP Models.
5. Understand the basics of Cryptography and Network security.

Course Outcomes:

1. To explain the OSI Reference Model and TCP/IP Models and in particular have a good knowledge of Layers.
2. To apply error correction and detection techniques of Data Link Layer.
3. To identify the best routing techniques by applying algorithms of Network Layer.
4. To explain the Transport Layer Protocols.
5. To explain the Application Layer Protocols, Cryptography and Network security.

UNIT - I

Data Communications: Components, Direction of Data flow, Networks, Components and Categories, Types of Connections, Topologies, Protocols and Standards, ISO /OSI model.
Physical Layer: Transmission modes, Multiplexing, Transmission Media, Switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks.

UNIT - II

Data Link Layer: Introduction, Framing, Error Detection and Correction, Parity, LRC, CRC Hamming code, Flo7w and Error Control, Noiseless Channels, Noisy Channels, HDLC, Point to Point Protocols.
Medium Access sub Layer: ALOHA, CSMA/CD,CSMA/CA, LAN – Ethernet IEEE 802.3, IEEE 802.5 – IEEE 802.11, Random access, Controlled access, Channelization.

UNIT - III

Network Layer: Logical Addressing, Inter-networking, Tunneling, Address mapping, ICMP, IGMP, Forwarding, Uni-Cast Routing Protocols, Multicast Routing Protocols.

UNIT - IV

Transport Layer: Process to Process Delivery, UDP and TCP protocols, Data Traffic, Congestion, Congestion Control, QoS, Integrated Services, Differentiated Services, and QoS in Switched Networks.

UNIT – V

Application Layer: Domain name space, DNS in internet, electronic mail, SMTP, FTP, WWW, HTTP, SNMP.

TEXT BOOKS:

1. Data Communications and Networking, Behrouz A. Forouzan , Fourth Edition TMH, 2006.
2. Computer Networks, Andrew S Tanenbaum, 4th Edition. Pearson Education, PHI.

REFERENCE BOOKS:

1. Data communications and Computer Networks, P.C .Gupta, PHI.
2. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education.
3. Understanding communications and Networks, 3rd Edition, W.A. Shay, Cengage Learning.
4. Computer Networking: A Top-Down Approach Featuring the Internet. James F. Kurose & Keith W. Ross, 3 rd Edition, Pearson Education.
5. Data and Computer Communication, William Stallings, Sixth Edition, Pearson Education, 2000.

22CS3211: MACHINE LEARNING

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

Prerequisites:

- A Course on "Data Analytics"
- A Course on "Computer Oriented Statistical methods"

Course Objectives

1. Define Machine Learning and understand the basic theory underlying machine learning.
2. Understand the basic concepts of learning and decision trees.
3. Understand neural networks and Bayesian techniques for problems appear in machine learning
4. Understand the instance based learning and reinforced learning
5. Perform statistical analysis of machine learning techniques

Course Outcomes: After the completion of the course student will be able to

1. Illustrate the learning techniques and investigate concept learning
2. Apply the characteristics of decision tree to solve associated problems
3. Use and Apply Ensemble and Un-Supervised Learning Techniques.
4. Apply effectively neural networks for appropriate applications
5. Evaluate hypothesis and investigate instant based learning and reinforced learning

UNIT-I

Introduction - Well-posed learning problems, designing a learning system, Perspectives and issues in machine learning Concept learning and the general to specific ordering – introduction, a concept learning task, concept learning as search, find-S: finding a maximally specific hypothesis, version spaces and the candidate elimination algorithm, remarks on version spaces and candidate elimination, inductive bias, Gradient Descent Algorithm and its variants.

UNIT-II

Supervised Learning- Regression: Linear-Simple, Multiple, Logistic Regression. Classification- Naive Bayes Classifier, k-NN classifier. Support Vector Machines -Linear, Non Linear Ensemble Techniques I-Decision Trees-ID3(Iterative Dichotomiser3), CART(Classification and Regression Tree)

UNIT-III

Ensemble Techniques II- C4.5, CHAID (Chi-Square Automatic Interaction Detection), Random Forest Algorithm Unsupervised Learning-Clustering: Measures of distance, k-means, Gaussian Mixture Model Clustering, Hierarchical Learning- Divisive, Agglomerative Clustering

UNIT-IV

Artificial Neural Networks-1– Introduction, neural network representation, appropriate problems for neural network learning, perceptions, multi layer networks and the back-propagation algorithm.

Artificial Neural Networks-2- Remarks on the Back-Propagation algorithm, An illustrative example: face recognition, advanced topics in artificial neural networks.

UNIT - V

Genetic Algorithms – Motivation, Genetic algorithms, an illustrative example, hypothesis space search, genetic programming, models of evolution and learning, parallelizing genetic algorithms. **Reinforcement Learning** – Introduction, the learning task, Q–learning, non-deterministic, rewards and actions, temporal difference learning, generalizing from examples, relationship to dynamic programming.

TEXT BOOK:

1. Machine Learning – Tom M. Mitchell, - MGH.
2. Introduction to Machine Learning with Python, Author – Andreas C. Müller, Sara h Guido, Edition – First Edition, Publisher – O'Reilly Media, Inc.

REFERENCE BOOK:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis.
2. Mathematics for Machine learning, Author – Marc Peter Deisenroth, Edition – First Edition, Publisher – Cambridge University Press.

22IT3271: BLOCKCHAIN TECHNOLOGY
(Professional Elective-III)

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

Pre-requisites:

- Knowledge in Computer Networks
- Knowledge in Distributed Databases.

.

Course Objectives:

1. Impart strong technical understanding of Blockchain technologies.
2. Gain knowledge about applications of cryptography in Blockchain.
3. Learn about the concepts of various implementations of Blockchain technology such as Bit coin, Ethereum and Hyper ledger.
4. Understand the modern currencies and their market usage.
5. Introduce application areas, current practices and research activity.

Course Outcomes: After the completion of the course student should be able to

1. Learn fundamentals of Blockchain techniques.
2. Analyze various consensus problems.
3. Adapt Bitcoin technology to improve usage.
4. Make use of Ethereum frameworks to write smart contract.
5. Interpret Blockchain technology in real time applications.

UNIT I

Introduction: What is Blockchain, The history of block chain, Benefits and limitations of Blockchain, Distributed systems, Decentralization using block chain, CAP theorem and block chain, Crowd funding.

UNIT II

Cryptography in Blockchain: Cryptocurrency, How a Cryptocurrency works, cryptographic primitives, Asymmetric cryptography, public and private keys, line interface, Bitcoin improvement proposals (BIPs) , Consensus Algorithms, Digital Identity verification, Blockchain Neutrality, Digital art.

UNIT III

Bitcoin:- The Bitcoin network, Wallets and its types, Bitcoin payments, Bitcoin investment and buying and selling bitcoins, Bitcoin installation, Bitcoin programming and the command line interface, Bitcoin improvement proposals (BIPs).

Blockchain Science: Grid coin, Folding coin, Blockchain Genomics

UNIT IV

Ethereum:- Ethereum Virtual Machine (EVM),Wallets for Ethereum, Solidity, Smart Contracts, Some Attacks on Smart Contracts, The Ethereum network, Applications developed on Ethereum , Scalability and security issues.

UNIT V

Issues in Blockchain: - Technical challenges, Business model challenges, Government Regulations, Zero Knowledge proofs and protocols in Blockchain

Introduction to Hyperledger: - Hyperledger as a protocol, Fabric, Hyper ledger Fabric, Saw tooth Lake, Corda Architecture.

Text Books:

1. Blockchain Blue print for Economy by Melanie Swan.
2. I. Bashir, Mastering Block chain: Distributed ledger technology, decentralization, and smart

contracts explained, 2nd Edition, 2nd revised edition. Birmingham: Packt Publishing, 2018.

References:

1. Vigna, Paul, and Michael J. Casey. The Truth Machine: The Block chain and the Future of Everything. Picador, 2019.

2. Gerard, David. Attack of the 50 foot block chain: Bitcoin, block chain, Ethereum & smart contracts. David Gerard, 2017.

3. Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Block chain Technology: Architecture, Consensus, and Future Trends," in 2017 IEEE International Congress on Big Data (Big Data Congress), 2017, pp.557–564.

22AM3271: CLOUD COMPUTING
(Professional Elective-III)

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

Pre-requisites:

- A course on "Computer Networks"
- A course on "Operating Systems"

Course Objectives:

1. To explain the evolving computer model called cloud computing.
2. To understand the current trend and basics of cloud computing.
3. To introduce the various levels of services that can be achieved by cloud.
4. To describe the security aspects in cloud.
5. To learn cloud enabling technologies and its applications.

Course Outcomes: After the completion of the course student should be able to

1. Understand various service delivery models of a cloud computing architecture.
2. Understand the virtualization and cloud computing concepts.
3. Understand cloud computing architecture and managing cloud infrastructure and its applications.
4. Acquire knowledge on cloud service models.
5. Acquire knowledge on cloud service providers.

UNIT-I

Computing Paradigms: High-Performance Computing, Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing, Cloud Computing, Bio computing, Mobile Computing, Quantum Computing, Optical Computing, Nano Computing.

UNIT-II

Cloud Computing Fundamentals: Motivation for Cloud Computing, The Need for Cloud Computing, Defining Cloud Computing, Definition of Cloud Computing, Cloud Computing Is a Service, Cloud Computing Is a Platform, Principles of Cloud computing, Five Essential Characteristics, and Four Cloud Deployment Models.

UNIT-III

Cloud Computing Architecture and Management: Cloud Architecture, Layer, Anatomy of the Cloud, Network Connectivity in Cloud Computing, Applications, on the Cloud, Managing the Cloud, Managing the Cloud Infrastructure Managing the Cloud Application, Migrating Application to Cloud, Phases of Cloud Migration Approaches for Cloud Migration.

UNIT-IV

Cloud Service Models: Infrastructure as a Service, Characteristics of IaaS. Suitability of IaaS, Pros and Cons of IaaS, Summary of IaaS Providers, Platform as a Service, Characteristics of PaaS,

Suitability of PaaS, Pros and Cons of PaaS, Summary of PaaS Providers, Software as a Service, Characteristics of SaaS, Suitability of SaaS, Pros and Cons of SaaS, Summary of SaaS Providers, Other Cloud Service Models

UNIT-V

Cloud Service Providers: EMC, EMCIT, Captiva Cloud Toolkit, Google, Cloud Platform, Cloud Storage, Google Cloud Connect, Google Cloud Print, Google App Engine, Amazon Web Services, Amazon Elastic Compute Cloud, Amazon Simple Storage Service, Amazon Simple Queue, Service,

Microsoft, Windows Azure, Microsoft Assessment and Planning Toolkit, SharePoint, IBM, Cloud Models, IBM Smart Cloud, SAP Labs, SAP HANA Cloud Platform, Virtualization Services Provided by SAP, Salesforce, Sales Cloud, Service Cloud: Knowledge as a Service, Rack space, VMware, Manjrasoft, Aneka Platform.

TEXT BOOKS:

1. Essentials of cloud Computing: K. Chandrasekhran, CRC press, 2014.

REFERENCES:

1. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Brobergand Andrzej, M. Goscinski, Wiley, 2011.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C.Fox, Jack J. Dongarra, Elsevier, 2012
3. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumara swamy, Shahed Latif, O'Reilly, SPD, rp201

22IT3272: COMPUTER VISION
(Professional Elective-III)

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

Prerequisite:
- Programming and Mathematics course.

Course Objectives:
1. Recognize and describe both the theoretical and practical aspects of computing with images. Connect issues from Computer Vision to Human Vision.
2. Describe the foundation of image formation and image analysis. Understand the basics of 2D and 3D Computer Vision.
3. Become familiar with the major technical approaches involved in computer vision. Describe various methods used for registration, alignment, and matching in images.
4. Get an exposure to advanced concepts leading to object categorization and segmentation in images.
5. Build computer vision applications.

Course Outcomes: After the completion of the course student should be able to
1. Implement fundamental image processing techniques required for computer vision.
2. Understand Image formation process.
3. Extract features from Images and do analysis of Images.
4. Generate 3D model from images and to develop applications using computer vision techniques.
5. Understand video processing, motion computation and 3D vision and geometry.

UNIT - I

Introduction: Image Processing, Computer Vision and Computer Graphics, What is Computer Vision - Low - level, Mid-level, High-level.

Overview of Diverse Computer Vision Applications: Document Image Analysis, Biometrics, Object Recognition, Tracking, Medical Image Analysis, Content-Based Image Retrieval, Video Data Processing, Multimedia, Virtual Reality and Augmented Reality

UNIT - II

Image Formation Models: Monocular imaging system, Radiosity: The 'Physics' of Image Formation, Radiance, Irradiance, BRDF, color etc., Orthographic & Perspective Projection, Camera model and Camera calibration, Binocular imaging systems, Multiple views geometry, Structure determination, shape from shading, Construction of 3D model from image.

Image Processing and Feature Extraction: Image preprocessing, Image representations (continuous and discrete), Edge detection.

UNIT - III

Motion Estimation: Regularization theory, Optical computation, Stereo Vision, Motion estimation, Structure from motion.

Shape Representation and Segmentation: Contour based representation, Region based representation, Deformable curves and surfaces, Snakes and active contours, Level set representations, Fourier and wavelet descriptors, Medial representations, Multi resolution analysis.

UNIT – IV

Object recognition: Hough transforms and other simple object recognition methods, Shape correspondence and shape matching, Principal Component analysis, Shape priors for recognition.

Image Understanding: Pattern recognition methods, HMM, GMM and EM.

UNIT – V

Applications: Photo album - Face detection – Face recognition - Eigen faces - Active appearance and 3D shape models of faces Application: Surveillance - foreground-background separation - particle filters - Chamfer matching, tracking, and occlusion - combining views from multiple cameras - human gait analysis Application: In-vehicle vision system: locating roadway - road markings - identifying road signs - locating pedestrians.

TEXT BOOKS:

1. Computer Vision - A modern approach, by D. Forsyth and J. Ponce, Prentice Hall Robot Vision, by B. K. P. Horn, McGraw-Hill.
2. Introductory Techniques for 3D Computer Vision, by E. Trucco and A. Verri, Publisher: Prentice Hall.
3. Multiple View Geometry in Computer Vision Second Edition, Richard Hartley and Andrew Zisserman, Cambridge University Press, March 2004.

REFERENCE BOOKS:

1. R. C. Gonzalez, R. E. Woods. Digital Image Processing. Addison Wesley Longman, Inc., 1992. Wiley Dreamtech.
2. D. H. Ballard, C. M. Brown. Computer Vision. Prentice-Hall, Englewood Cliffs, 1982. Java Script, D. Flanagan, O'Reilly, SPD.
3. Richard Szeliski, Computer Vision: Algorithms and Applications (CVAA). Springer, 2010.
4. Mark Nixon and Alberto S. Aquado, Feature Extraction & Image Processing for Computer Vision, Third Edition, Academic Press, 2012.
5. Image Processing, Analysis, and Machine Vision. Sonka, Hlavac, and Boyle. Thomson.

22IT3273: EMBEDDED SYSTEMS
(Professional Elective-III)

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

Prerequisites
- A course on "Digital Logic Design".
- A course on "Computer Organization".

Course Objectives:
1. Introduce the basic concepts of an embedded system.
2. Explain various elements of embedded hardware and their design principles
3. Elaborate different steps involved in the design and development of firmware for embedded systems.
4. Discuss Internals of Real-Time operating system, the fundamentals of RTOS based embedded firmware design and fundamental issues in hardware software co-design.
5. Familiarize with different embedded system implementation and testing tools.

Course Outcomes:
1. Explain the basic concepts and the embedded system design approach to perform a specific function.
2. Analyze the hardware components required for an embedded system and the design approach of an embedded hardware.
3. Analyze various embedded firmware design approaches on embedded environment.
4. Evaluate the issues in hardware software co-design.
5. Integrate hardware and firmware of an embedded system using real time operating systems.

UNIT I
Embedded system-Definition, History of embedded systems, classification of embedded systems, major application areas of embedded systems, purpose of embedded systems, the typical embedded system-core of the embedded system, Memory, Sensors and Actuators, Communication Interface, Embedded firmware, Characteristics of an embedded system, Quality attributes of embedded systems, Application-specific and Domain-Specific examples of an embedded system.

UNIT II
Embedded hardware design: Analog and digital electronic components, I/O types and examples, Serial communication devices, Parallel device ports, Wireless devices, Timer and counting devices, Watchdog timer, Real time clock.

UNIT III
Embedded firmware design: Embedded Firmware design approaches, Embedded Firmware development languages, ISR concept, Interrupt sources, Interrupt servicing mechanism, Multiple interrupts, DMA, Device driver programming, Concepts of C versus Embedded C and Compiler versus Cross-compiler.

UNIT IV
Real time operating system: Operating system basics, Types of operating systems, Tasks, Process and Threads, Multiprocessing and Multitasking, Task Scheduling, Threads, processes and Scheduling, Task communication, Task synchronization.
Hardware software co-design: Fundamental Issues in Hardware Software Co-Design, Computational models in embedded design, Hardware software Trade-offs, Integration of Hardware and Firmware.

UNIT V

Embedded system development, implementation and testing: The integrated development environment, Types of files generated on cross-compilation, Dissembler/DE compiler, Simulators, Emulators and Debugging, Target hardware debugging, Embedded Software development process and tools, Interpreters, Compilers and Linkers, Debugging tools, Quality assurance and testing of the design, Testing on host machine, Simulators, Laboratory Tools.

Text Books:

1. Embedded Systems Architecture-By Tammy Noergaard, Elsevier Publications, 2013.
2. Embedded Systems-By Shibu. K.V-Tata McGraw Hill Education Private Limited, 2013.

References:

1. Embedded System Design, Frank Vahid, Tony Givargis, John Wiley Publications, 2013.
2. Embedded Systems-Lyla B.Das-Pearson Publications, 2013.

22IT3251: Compiler Design Lab

B.Tech III Year II Sem **L T P C**
0 0 2 1

Prerequisites:

- Knowledge on C programming

Course Objectives

1. To understand the various phases in the design of a compiler.
2. To understand syntax directed translation schemes
3. To introduce Lex and YACC tools.
4. To implement Top-down parsing technique.
5. To implement Bottom-up parsing technique.

Course Outcomes:

1. Ability to design, develop, and implement a compiler for any language.
2. Ability to use different tools in construction of the phases of a compiler for the mini language.
3. Ability to implement Lexical Analyzer for given language using C and Lex tool.
4. Ability to use YACC tools for developing a parser.
5. Able to design and implement LL and LR parsers.

List of experiments:

1. Design a lexical analyser for given language and the lexical analyser should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C language.
2. a. Write a C program to identify whether a given line is a comment or not
   b.Write a C program to test whether a given identifier is valid or not.
3. Write a C program to simulate lexical analyzer for validating operators
4. To Study about Lexical Analyzer Generator (LEX) and Flex (Fast Lexical Analyzer)
5. Implement following programs using Lex.
   a. Create a Lexer to take input from text file and count no of characters, no. of lines & no. of words.
   b. Write a Lex program to count number of vowels and consonants in a given input string.
6. Implement following programs using Lex.
   a. Write a Lex program to print out all numbers from the given file.
   b. Write a Lex program to printout all HTML tags in file.c.
7. Write a Lex program which adds line numbers to the given file and display the same onto the standard output.
8. Write a C program for constructing of LL (1) parsing.
9. Write a C program for constructing recursive descent parsing
10. Write a C program to implement LALR parsing.

Text Books:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman.

References:

1. Lex&Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction, Louden, Thomson.

22IT3252:DATA COMMUNICATION AND COMPUTER NETWORKS LAB

	L	T	P	C
B.Tech. III Year II Sem.	**0**	**0**	**2**	**1**

Prerequisites:

- Basic Knowledge of C and networking concepts.

Course Objectives:

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

Course Outcomes:

1. To implement data link layer farming methods.
2. To analyze error detection and error correction codes.
3. To implement and analyze routing and congestion issues in network design.
4. To implement Encoding and Decoding techniques used in presentation layer.
5. To be able to work with different network tools.

List of Experiments:

Part - A

1. Identifying various Network Devices & Demonstration of Assigning MAC address.
2. Write a program to implement data link layer framing method bit stuffing.
3. Write a program to implement data link layer framing method character stuffing.
4. Write a program to implement data link layer framing method character count.
5. Write a program to implement Cyclic Redundancy Check(CRC 12 ,CRC 16 and CRC CCIR) on a data set of characters.
6. Implement Dijsktra's algorithm to compute the shortest path through a network.
7. Implement distance vector routing algorithm for obtaining routing tables at each node.Write a program to implement encryption and decryption.

Part - B.

8. All the Experiments may be Conducted using Network Simulation software like NS-2, NSG-2.1 and Wire SHARK/equivalent software.

Note: Experiments Performance may be evaluated through simulation by using the parameters Throughput, Packet Delivery Ratio, Delay etc.

9. Evaluate the performance of various LAN Topologies.
10. Evaluate the performance of TCP and UDP Protocols.
11. Evaluate the performance of IEEE 802.11 and IEEE 802.15.4.
12. Capturing and Analysis of TCP and IP Packets.
13. Simulation and Analysis of ICMP and IGMP Packets.
14. Analysis of HTTP, DNS and DHCP Protocols.

TEXT BOOKS:

1. Computer Science: A Structured Programming Approach Using C, B. A. Forouzan and R. F. Gilberg, Third Edition, Cengage Learning.
2. Data Communications and Networking - Behrouz A. Forouzan, Fifth Edition TMH, 2013.

REFERENCE BOOK:

1. The C Programming Language, B.W. Kernighan and Dennis M. Ritchie, Second Edition, Pearson education.

22IT3253:MACHINE LEARNING LAB

B.Tech III Year II Sem **L T P C**
0 0 2 1

Prerequisites:

- Knowledge of Java Programming.
- Data mining concepts.

Course Objectives:

1) Understand basics and functions using Python programming language.
2) Understand all principal elements of Computational Learning Theory.
3) Gain the knowledge of decision tree and decision tree learning algorithms.
4) Make use of Data sets in implementing the machine learning algorithms.
5) Implement the machine learning concepts and algorithms and to understand the high-performance programs designed to strengthen practical expertise.

Course Outcomes:

At the end of the course, students would be able to

1) Understand the basic concepts of scripting and to explore Python especially the object-oriented concepts, and the built-in objects of Python.
2) Observe the concepts of computational intelligence like machine learning and Design an exemplarily learning system.
3) Apply the algorithms (Decision Tree techniques) to a real-world problem, optimize the models learned andreport on the expected accuracy.
4) Analyze the Neural Networks and its usage in machine learning applications.
5) Apply Bayesian reasoning and also target based learning techniques to develop a machine learning application and analyze the different search methods.

List of Programs:

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample
4. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

9. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

Text Books:

1. Machine Learning, Tom M Michel, McGraw Hill, 1997.

References:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis
2. https://towardsdatascience.com/tagged/model - evaluation
3. https://github.com/topics/handwriting-recognition?l=python

22IT3281 : INDUSTRIAL ORIENTED MINI PROJECT

III Year B.Tech. II- Sem

L	T	P	C
0	0	4	2

22MC0002:ENVIRONMENTAL SCIENCE

III Year B.Tech. II Sem

L	T	P	C
3	0	0	0

Course Objectives

Develop ability to

1. Identify the importance of ecosystem and its functions.
2. Understand the natural resources and their usage in day to day life.
3. Understand the concept of bio-diversity, its values and conservation.
4. Be aware of the causes of different types of pollution and its control.
5. Understand various environmental impacts, requirement of various policiesand legislations towards environmental sustainability.

Course Outcomes: After the completion of the course, the student would be able to –

1. Explain ecosystem and its functions namely, food chain, ecological pyramids etc.
2. Acquire knowledge about different types of natural resources such as land, water, minerals, non-renewable energy and their excessive usage leading to detrimental effects on environment.
3. Comprehend ecosystem diversity, its values and importance of hot spots to preserve the same.
4. Explain different types of pollution, its control and impact on global environment.
5. Recognize various environmental impacts and the importance of various acts and policies towards environmental sustainability.

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, Bio magnifications, 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. Environmental effects of extracting and using mineral resources, Land resources: Forest resources, Energy Resources-renewable and non-renewable.

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. Hot spots of biodiversity. 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. Global Environmental Issues and Global Efforts: Green House Gases And its effect, Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol.

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, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio-economic 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. Erach Bharucha, Textbook of Environmental Studies for Undergraduate Courses, University
2. Grants Commission.
3. R. Rajagopalan, Environmental Studies, Oxford University Press.

REFERENCES:

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