

**B. Tech -Information Technology****III B.Tech I SEMESTER**

S.No	Course Code	Course Title	Category	L	T	P	Credits
1	19IT3111	Design and Analysis of Algorithms	PC	3	-	-	3
2	19IT3112	Web Technologies	PC	3	1	-	4
3	19CS3113	Software Engineering	PC	3	-	-	3
4	<b>Open Elective – I</b>		OE	3	-	-	3
5	<b>Professional Elective – I</b>		PE	3	-	-	3
	19IT3171	Artificial Intelligence					
	19CS3172	Principles of Programming Languages					
	19IT3172	Computer Graphics					
	19IT3173	Mobile Computing					
6	19MC0005	Professional Ethics	MC	2	-	-	-
7	19IT3151	Design and Analysis of Algorithms Lab	PC	-	-	3	1.5
8	19IT3152	Web Technologies and Unified Modeling Language Lab	PC	-	-	3	1.5
9	19HS3151	Advanced English Communication Skills Lab	HS	-	-	2	1
10	19IT3181	Summer Internship*	PW	-	-	2	1
<b>Total</b>				17	1	10	<b>21</b>

\*Note: Summer Internship to be carried out during summer break after II-year II semester.

**III B. Tech II SEMESTER**

S.No	Course Code	Course Title	Category	L	T	P	Credits
1	19IT3211	Information Security	PC	3	1	-	4
2	19IT3212	Data Warehousing and Data Mining	PC	3	-	-	3
3	19IT3213	Automata & Compiler Design	PC	3	1	-	4
4	<b>Open Elective – II</b>		OE	3	-	-	3
5	<b>Professional Elective – II</b>		PE	3	-	-	3
	19IT3271	Cloud Computing					
	19IT3272	Embedded Systems					
	19CS3273	Information Retrieval System					
	19CS3274	Software Testing Methodologies					
6	19IT3251	Information Security Lab	PC	-	-	3	1.5
7	19IT3252	Data Warehousing and Data Mining Lab	PC	-	-	3	1.5
8	19IT3291	Technical Paper Presentation	PW	-	-	2	1
<b>Total</b>				15	2	8	<b>21</b>

**Course Code: 19IT3111**

**DESIGN AND ANALYSIS OF ALGORITHMS**

**B.Tech. III Year I Sem.**

**L T P C**

**3 - - 3**

**Course Objectives:**

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

**Course Outcomes:**

- Able to analyze the different algorithm design techniques for a given problem.
- Able to design algorithms for various computing problems.
- Able to argue the correctness of algorithms using inductive proofs and invariants.
- Able to synthesize set operations
- Able to 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, Randomized Algorithms-Monte Carlo and Las Vegas, Amortized analysis Examples.

**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**

**Disjoint Set Operations:** Union and find algorithms, AND/OR graphs, Graph traversals-Breadth first search, Depth First search, Connected Components and Spanning trees, Bi-connected components.

**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.

**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, Traveling sales person problem.

## UNIT –V

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

**Approximation Algorithms for NP – Hard Problems:** Traveling Salesman problem, Knapsack problem.

### 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 Pearson Publication.

### REFERENCE BOOKS

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

**Prerequisite:**

- Object oriented concepts

**Course Objectives:**

- To introduce HTML, CSS and to introduce Client-Side Scripting with JavaScript.
- To introduce XML and processing of XML Data with Java.
- To introduce Server-side programming with Java Servlets.
- To introduce JSP and MVC architecture.
- To introduce PHP language for server-side scripting.

**Course Outcomes:**

- To gain knowledge of html, client-side scripting, validation of forms.
- Implement XML and how to parse and use XML Data with Java.
- To implement JDBC connectivity to any database and to have understanding of server-side scripting using Servlets.
- To introduce Server-side programming with JSP.
- To implement PHP Server side scripts.

**UNIT I**

**Introduction to HTML:** Introduction to HTML: Table, images, forms, frames, CSS, Client side scripting:

**Introduction to Java Script:** Java script language – declaring variables, scope of variables, functions, event handlers (onclick, onsubmit etc.), Document Object Model, Form validation.

**UNIT II**

**XML:** Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemas, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

**UNIT III**

**JDBC:** Introduction, Types of JDBC Drivers, Process to establish a connection, Types of Statements, Result set Metadata.

**Introduction to Servlets:** Common Gateway Interface (CGI), Lifecycle of a Servlet, Deploying a Servlet, The Servlet API, Reading Servlet 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, MVC architecture Using Cookies and Sessions for SessionTracking, Connecting to database using JDBC, Simple AJAX application.

## **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 and Contemporary topics.

## **TEXT BOOKS:**

- [1] Web Technologies, Uttam K Roy, Oxford University Press.
- [2] The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill.

## **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  
Internet and World Wide Web – How to program, Dietel and Nieto, Pearson

**Prerequisites:**

- Computer Programming
- Database Management Systems

**Course Objectives:**

- To understand fundamental principles of Software engineering, and their application in the development of software products.
- To understand and create the software requirements specifications document.
- To understand and use unified modeling language for specifying, analysis and designing.
- To understand testing strategies for testing software applications
- To understand Software metrics and Risk Management strategies to identify potential problems before they occur.

**Course Outcomes:**

- Able to apply the software engineering lifecycle phases communication, planning, analysis, design, construction, and deployment.
- Ability to translate end-user requirements into system and software requirements into Software Requirements specification Document (SRS)
- Able to apply UML in object-oriented software modeling to develop computer software.
- Able to identify problems in software and will be able to develop a simple testing report.
- To understand Software Metrics, potential risk and how to manage them through RMMM plan.

**UNIT – I**

**Introduction to Software Engineering:** 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, COCOMO Model.

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

**UNIT - II: Software Requirements**

**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: Implementation and Quality Assurance**

**Implementation:** Structured coding Techniques, Coding Styles-Standards and Guidelines.

**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: Metrics for Process and Products**

**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, 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 Rumbaugh, Ivar Jacobson, Pearson Education.

### **REFERENCE BOOKS**

- [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.

**Prerequisites:**

- Discrete mathematics, basic probability theory and Statistics
- Knowledge of any programming language and data structures

**Course Objectives:**

- To provide introduction to the problem-solving techniques
- To have an understanding of topics such as heuristic search, blind search, minimax search etc. that play an important role in AI programs
- To familiarize with the knowledge representation techniques used in AI.
- To give exposure to learning methods used in Artificial Intelligence.
- To have a basic understanding of some of the more advanced topics of AI such as the Fuzzy Logic and knowledge processing in expert systems.

**Course Outcomes:**

- Able to formulate an efficient problem space for a problem expressed in natural language.
- Able to identify and trace the different search algorithms.
- Able to Possess the skill for representing knowledge using the appropriate technique for a given problem.
- Able to summarize different learning methods used in AI.
- Able to summarize the significance of Fuzzy Logic and expert systems in AI.

**UNIT -I**

**Introduction to AI:** History, Intelligent Systems, Foundations of AI, Sub areas of AI, Applications.

**Problems, Problem Spaces and Search:** Defining the Problem as a State Space Search, Production Systems, Problem Characteristics, Production System Characteristics, Issues in the Design of Search Programs.

**UNIT -II**

**Basic Search Strategies:** Problem Spaces, Uninformed Search (Breadth-First, Depth-First Search, Depth-first with Iterative Deepening), Heuristic Search (Hill Climbing, Generic Best-First, A\*), Constraint Satisfaction (Backtracking, Local Search).

**Advanced Search:** Constructing Search Trees, Stochastic Search, A\* Search Implementation, Minimax Search, Alpha-Beta Pruning.

**UNIT -III**

**Knowledge Representation:** Issues in Knowledge Representation, Representing Simple Facts in Predicate Logic, Representing Instance and ISA Relations, Computable Functions and Predicates, Resolution, Natural Deduction.

**Representing Knowledge Using Rules:** Procedural Vs Declarative Knowledge, Logic Programming, Forward Vs Backward Reasoning, Matching, Control Knowledge

**UNIT -IV**

**Learning:** What Is Learning? Rote Learning, Learning by Taking Advice, Learning in Problem Solving, Learning from Examples, Winston's Learning Program, Decision Trees.



## **UNIT -V**

**Expert Systems:** Representing and Using Domain Knowledge, Shell, Explanation, Knowledge Acquisition.

**Fuzzy Sets** – Concept of a Fuzzy number- Operations on Fuzzy Sets – Typical Membership Functions – Discrete Fuzzy Set

### **TEXT BOOKS:**

- [1] Elaine Rich, Kevin Knight, Shivashankar B Nair, Tata McGraw Hill - Artificial Intelligence, 3rd Edition, 2004.
- [2] Stuart Russell – Peter Narang, Pearson Education Asia - Artificial Intelligence - A modern approach.

### **REFERENCE BOOKS:**

- [1] George F Luger - Artificial Intelligence, Pearson Education Asia Allen B. Downey – (Think Python) Python for software design- How to think like a computer scientist, Cambridge University press, 2009.

Course Code: 19CS3172

**PRINCIPLES OF PROGRAMMING LANGUAGES**

(Professional Elective-I)

**L T P C**

**3 - - 3**

**B.Tech III Year I Sem.**

**Prerequisites:**

- A course on “Mathematical Foundations of Computer Science”
- A course on “Computer Programming and Data Structures”

**Course Objectives:**

- Introduce important paradigms of programming languages
- To provide conceptual understanding of high-level language design and implementation
- Topics include programming paradigms; syntax and semantics; data types, expressions and Statements; subprograms and blocks; abstract data types; concurrency; functional and logic Programming languages; and scripting languages

**Course Outcomes:**

- Acquire the skills for expressing syntax and semantics in formal notation
- Identify and apply a sub programming paradigms for a given computing application
- Able to understand high-level language design and implementation
- Gain knowledge on Interprocess communication.
- Gain knowledge of and able to compare the features of various programming 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, and 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**

**Object Oriented Programming:** Design issues for OOP, OOP in Smalltalk, C++, Java, Ada 95, Ruby, Implementation of Object-Oriented constructs.

**Concurrency:** Introduction, Introduction to Subprogram Level Concurrency, Semaphores, Monitors, Ada support for concurrency, 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 Dream Tech, 2007.

#### **REFERENCE BOOKS:**

- [1] Programming Languages, 2nd Edition, A.B. Tucker, R. E. Noonan, TMH.
- [2] Programming Languages, K. C. Loudon, 2nd Edition, Thomson, 2003

## B.Tech III Year I Sem.

## Course Objectives:

- To introduce the fundamental concepts and theory of computer graphics.
- To introduce graphics systems and input devices; geometric representations and 2D/3D transformations; viewing and projections; illumination and color models; animation; rendering and implementation; visible surface detection.

## Course Outcomes:

- Create interactive graphics applications using one or more graphics application programming interfaces
- Use graphics packages and displaying techniques.
- Be able to design applications that display graphic images to given specifications
- Analyze viewing and geometrical transformations.
- Apply color models.

## UNIT – I

**Introduction:** Image Processing as Picture Analysis, the Advantages of Interactive Graphics, Representative Uses of Computer Graphics, Classification of Applications, Development of Hardware and Software for Computer Graphics, Conceptual Framework for Interactive Graphics.

## UNIT – II

**Programming in the SRGP & Raster Graphics Algorithms:** Drawing with SRGP, Basic Interaction Handling, Raster Graphics Features, Limitations of SRGP, Basic Raster Graphics Algorithms For Drawing 2D Primitives: Overview, Scan Converting Lines, Scan Converting Circles, Filling Rectangles, Filling Polygons, Line Style and Pen Style.

## UNIT – III

**3-D Object Representation:** Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon rendering methods.

## UNIT – IV

**2D Geometrical Transformations:** 2D Transformations, Window-to-Viewport Transformation.

**3D Geometrical Transformations:** Matrix Representation of 3D Transformations.

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

## UNIT – V

**Achromatic & Colored Light:** Achromatic Light, Chromatic Color, Color Models for Raster Graphics, Reproducing Color, Using Color in Computer Graphics.

## 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

**REFERENCE BOOKS:**

- [1] Procedural elements for Computer Graphics, David F Rogers, Tata Mc Graw hill, 2nd edition.
- [2] Principles of Interactive Computer Graphics”, Neuman and Sproul, TMH.
- [3] Computer Graphics: A Practical Approach, Er. Rajiv Chopra, S.Chand.

**Course Objectives:**

Students should be able:

- To introduce the concept of mobile computing paradigm, its novel applications and limitations.
- To introduce the typical mobile networking infrastructure through a popular GSM protocol
- To introduce the issues and solutions of various layers of mobile networks, namely MAC layer, Network Layer & Transport Layer
- To introduce the database issues in mobile environments & data delivery models.
- To introduce the ad hoc networks and related concepts.
- To introduce the platforms and protocols used in mobile environment.

**Course Outcomes:**

- Able to understand the concept and architecture of mobile network through GSM protocol and to develop new mobile application.
- Able to take any new technical issue related to this new paradigm and come up with a solution(s).
- Able to develop new ad hoc network applications and/or algorithms/protocols.
- Able to understand file systems and transactions for mobility support and develop mobile applications
- Able to understand and develop any existing or new protocol related to mobile environment

**UNIT-I**

**Introduction:** Mobile Communications, Mobile Computing – Paradigm, Promises/Novel Applications and Impediments and Architecture; Mobile and Handheld Devices, Limitations of Mobile and Handheld Devices.

**GSM** – Services, System Architecture, Radio Interfaces, Protocols, Localization, Calling, Handover, Security, New Data Service- GPRS.

**UNIT-II**

**(Wireless) Medium Access Control (MAC):** Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA, comparison of S/T/F/CDMA.

**Mobile Network Layer:** IP and Mobile IP Network Layers, Packet Delivery and Handover Management, Location Management, Registration, Tunneling and Encapsulation, Route Optimization, DHCP.

**UNIT-III**

**Mobile Transport Layer** - Conventional TCP/IP Protocols, Indirect TCP, Snooping TCP, Mobile TCP, Other Transport Layer Protocols for Mobile Networks.

**Database Issues:** Database Hoarding & Caching Techniques, Client-Server Computing & Adaptation, Transactional Models, Query processing, Data Recovery Process & QoS Issues

#### **UNIT-IV**

**Mobile Ad hoc Networks (MANETs):** Introduction, Applications & Challenges of a MANET, Routing, Classification of Routing Algorithms, Algorithms such as DSR, AODV, DSDV. Mobile Agents, Service Discovery.

**Wireless Sensor Networks.** VANET, Applications of VANET, PAN.

#### **UNIT-V**

**Protocols and Platforms for Mobile Computing-** WAP, Bluetooth, XML, J2ME, Java Card, Palm OS, Windows CE, Symbian OS, Linux for Mobile Devices, Android.

#### **TEXT BOOKS:**

- [1] Jochen Schiller, “Mobile Communications”, Addison-Wesley, Second Edition, 2009.
- [2] Raj Kamal, “Mobile Computing”, Oxford University Press, 2007, ISBN: 0195686772.

#### **REFERENCE BOOKS:**

- [1] Jochen Schiller, “Mobile Communications”, Addison-Wesley, Second Edition, 2004.
- [2] Stojmenovic and Cacute, “Handbook of Wireless Networks and Mobile Computing”, Wiley, 2002, ISBN 0471419028.
- [3] Reza Behravanfar, “Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML”, ISBN: 0521817331, Cambridge University Press, Oct 2004.

**Course Objective:**

To enable the students to imbibe and internalize the Values and Ethical Behaviour in the personal and Professional lives.

**Course Outcome:**

The students will understand the importance of Values and Ethics in their personal lives and professional careers. The students will learn the rights and responsibilities as an employee, team member and a global citizen.

**UNIT - I**

**Introduction to Professional Ethics:** Basic Concepts, Governing Ethics, Personal & Professional Ethics, Ethical Dilemmas, Life Skills, Emotional Intelligence, Thoughts of Ethics, Value Education, Dimensions of Ethics, Profession and professionalism, Professional Associations, Professional Risks, Professional Accountabilities, Professional Success, Ethics and Profession.

**UNIT - II**

**Basic Theories:** Basic Ethical Principles, Moral Developments, Deontology, Utilitarianism, Virtue Theory, Rights Theory, Casuist Theory, Moral Absolution, Moral Rationalism, Moral Pluralism, Ethical Egoism, Feminist Consequentialism, Moral Issues, Moral Dilemmas, Moral Autonomy.

**UNIT - III**

**Professional Practices in Engineering:** Professions and Norms of Professional Conduct, Norms of Professional Conduct vs. Profession; Responsibilities, Obligations and Moral Values in Professional Ethics, Professional codes of ethics, the limits of predictability and responsibilities of the engineering profession. Central Responsibilities of Engineers - The Centrality of Responsibilities of Professional Ethics; lessons from 1979 American Airlines DC-10 Crash and Kansas City Hyatt Regency Walk away Collapse.

**UNIT - IV**

Work Place Rights & Responsibilities, Ethics in changing domains of Research, Engineers and Managers; Organizational Complaint Procedure, difference of Professional Judgment within the Nuclear Regulatory Commission (NRC), the Hanford Nuclear Reservation. Ethics in changing domains of research - The US government wide definition of research misconduct, research misconduct distinguished from mistakes and errors, recent history of attention to research misconduct, the emerging emphasis on understanding and fostering responsible conduct, responsible authorship, reviewing & editing.

**UNIT - V**

Global issues in Professional Ethics: Introduction – Current Scenario, Technology Globalization of MNCs, International Trade, World Summits, Issues, Business Ethics and Corporate Governance, Sustainable Development Ecosystem, Energy Concerns, Ozone Deflection, Pollution, Ethics in Manufacturing and Marketing, Media Ethics; War Ethics; Bio Ethics, Intellectual Property Rights.



**TEXT BOOKS:**

1. Professional Ethics: R. Subramanian, Oxford University Press, 2015.
2. Ethics in Engineering Practice & Research, Caroline Whitbeck, 2e, Cambridge University Press 2015.

**REFERENCES:**

1. Engineering Ethics, Concepts Cases: Charles E Harris Jr., Michael S Pritchard, Michael J Rabins, 4e , Cengage learning, 2015.
2. Business Ethics concepts & Cases: Manuel G Velasquez, 6e, PHI, 2008.

<b>Prerequisites:</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
	-	-	3	1.5

- Basic Knowledge of java
- Working Principle of Algorithm

**Course Objectives:**

- To write programs in java to solve problems using divide and conquer strategy.
- To write programs in java to solve problems using backtracking strategy.
- To write programs in java to solve problems using greedy and dynamic programming techniques.

**Course Outcomes:**

- Ability to write programs in java to solve problems using algorithm design techniques such as Divide and Conquer, Greedy, Dynamic programming, and Backtracking.

**List of Experiments:**

1. Write a java program to implement Merge Sort algorithm for sorting a list of integers in ascending order
2. Write a java program to implement Quick Sort algorithm for sorting a list of integers in ascending order
3. Write a java program to implement the dfs algorithm for a graph.
4. Write a java program to implement the bfs algorithm for a graph.
5. Write a java program to implement greedy algorithm for job sequencing with deadlines.
6. Write a java program to implement Dijkstra's algorithm for the Single source shortest path problem.
7. Write a java program that implements Prim's algorithm to generate minimum cost spanning tree.
8. Write a java program that implements Kruskal's algorithm to generate minimum cost spanning tree.
9. Write a java program to implement Dynamic Programming algorithm for the 0/1 Knapsack problem.
10. Write a java program to implement Dynamic Programming algorithm for the Optimal Binary Search Tree Problem.
11. Write a java program to implement Floyd's algorithm for the all pairs shortest path problem.
12. Write a java programs to implement backtracking algorithm for the N-queens problem.

13. Write a java program to implement the backtracking algorithm for the sum of subsets problem.
14. Write a java program to implement the backtracking algorithm for the Hamiltonian Circuits problem.
15. Write a java program to Implement Graph Coloring using Back Tracking.

### **TEXT BOOKS**

- [1] Data structures, Algorithms and Applications in java, 2nd Edition, S. Sahani, Universities Press.
- [2] Data structures and Algorithms in java, 3rd edition, A. Drozdek, Cengage Learning.
- [3] Data structures with Java, J. R. Hubbard, 2nd edition, Schaum's Outlines, TMH.

### **REFERENCE BOOKS**

- [1] Data structures and algorithms in Java, 2nd Edition, R. Lafore, Pearson Education.
- [2] Data Structures using Java, D. S. Malik and P.S. Nair, Cengage Learning

**WEB TECHNOLOGIES & UNIFIED MODELING LANGUAGE LAB**

**B.Tech. III Year I Sem.**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>

**Course Objectives:**

- To enable the student to program web applications using the following technologies HTML, JavaScript , PHP, Tomcat Server, Servlets, JSP.
- To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

**Course Outcomes:**

- Students will be in a position to implement client-side applications by using technologies like HTML, JavaScript.
- Use LAMP Stack for web applications.
- Use Tomcat Server for Servlets and JSPs for dynamic development.
- Students will be in a position to implement JDBC applications to connect to any data base.
- Students will have hands on experience to prepare Software Requirement Specification Document and usage of any Design phase CASE tool

**Note:**

- Use LAMP Stack (Linux, Apache, MySQL and PHP) for the Lab Experiments. Though not mandatory, encourage the use of Eclipse platform wherever applicable
- The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed

**EXPERIMENTS:**

1. Write the following HTML Programs.
  - a) HTML Programs to work with Lists, Tables and CSS.
  - b) HTML Program to design Login page and Registration page.
2. Write a JavaScript program that takes a given set of integer numbers and shows them after sorting in descending order.
3. 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 to 12.
  - c) Phone number contains 10 digits.
  - d) E-mail must follow some predefined format([example@domain.com](mailto:example@domain.com))

4. Write a Java Script program 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 Java Script program for simple calculator web application that takes two numbers and an operator (+, -, /, \* and %) from an HTML page and returns the result page with the operation performed on the operands.
6. Write the following XML Programs
  - a) Create a DTD document to validate the XML document.
  - b) Create a XML Schema document to validate the XML document
7. Install the following on the local machine
  - a) Apache Web Server (if not installed)
  - b) Tomcat Application Server locally
  - c) Install MySQL (if not installed)
  - d) Install PHP and configure it to work with Apache web server and MySQL
8. Write a Servlet program to read the parameters from user interface and display welcome message.  
Write a Servlet program to read initialization parameters using ServletConfig and ServletContext object.
9. Write a Servlet program to work with HttpSession Object.  
Write a Servlet program to work with Cookie.
10. Write a JSP Program to handle the exceptions.  
Write a JSP Program to retrieve the student data from database
11. Write a JSP Program to access bean information using useBean tag.  
Write a JSP Program to authenticate the login details. If user is valid forward the control to “**Success.html**” otherwise forward to “**Fail.html**”.
12. Write a JDBC Program to create a student table in the database using servlets.  
Write a JDBC Program to perform various DML Operations on the database using Statement using JSP.
13. Write a PHP program to work with associative arrays.  
Write a PHP program to find factorial using Recursion.
14. Write a PHP Program to perform various DDL operations on MySQL database.  
Write a PHP Program to perform various DML operations on MySQL database.

### **UML - List of Experiments:**

**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

### **TEXT BOOKS:**

- [1] Web Technologies, Uttam K Roy, Oxford University Press.
- [2] The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill.
- [3] The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

### **REFERENCE BOOKS:**

- [1] The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill.
- [2] Web Programming, building internet applications, Chris Bates 2nd edition, Wiley Dreamtech.
- [3] Java Server Pages –Hans Bergsten, SPD O'Reilly.
- [4] Java Script, D.Flanagan, O'Reilly, SPD.
- [5] Internet and World Wide Web – How to program, Dietel and Nieto, Pearson.
- [6] Software Engineering, A practitioner's Approach- Roger S. Pressman, 6<sup>th</sup> edition McGraw Hill International Edition.
- [7] Software Engineering- Sommerville, 7<sup>th</sup> edition, Pearson Education.

**Course Code: 19HS3151**

**ADVANCED ENGLISH COMMUNICATION SKILLS LAB**

**B.Tech. III Year I Sem.**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>2</b>	<b>1</b>

**Introduction**

A course on Advanced English Communication Skills (AECS) Lab is considered essential at the third year level of B.Tech and Pharmacy courses. At this stage, the students need to prepare themselves for their career which requires them to listen to, read, speak and write in English both for their professional and interpersonal communication. The main purpose of this course is to prepare the students of Engineering for their placements.

**Course Objectives:**

This Lab focuses on using multi-media instruction 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
- To make students industry-ready
- To help students acquire behavioral skills for their personal and professional life
- To respond appropriately in different socio-cultural and professional contexts

**Course Outcomes: Students will be able to:**

- Acquire vocabulary and use it contextually
- Listen and speak effectively
- Develop proficiency in academic reading and writing
- Increase possibilities of job prospects
- Communicate confidently in formal and informal contexts

**UNIT-I**

Inter-personal Communication and Building Vocabulary –Starting a Conversation– Responding Appropriately and Relevantly –Using Appropriate Body Language –Role Play in Different Situations –Synonyms and Antonyms, One-word Substitutes, Prefixes and Suffixes, Idioms and Phrases and Collocations.

**UNIT-II**

Reading Comprehension –General Vs Local Comprehension, Reading for Facts, Guessing Meanings from Context, Skimming, Scanning, Inferring Meaning.

**Unit-III**

Writing Skills –Structure and Presentation of Different Types of Writing –Letter Writing/Resume Writing/ e-correspondence/ Technical Report Writing.

**UNIT-IV**

Presentation Skills –Oral Presentations (individual or group) through JAM Sessions/Seminars/PPTs and Written Presentations through Posters/Projects/Reports/ emails/Assignments...etc.,

## **UNIT-V**

Group Discussion and Interview Skills –Dynamics of Group Discussion, Intervention, Summarizing, Modulation of Voice, Body Language, Relevance, Fluency and Organization of Ideas and Rubrics of Evaluation-Concept and Process, Pre-interview Planning, Opening Strategies, Answering Strategies, Interview through Tele-conference & Video-conference and Mock Interviews.

### **REFERENCE BOOKS:**

- [1] Kumar, Sanjay and Pushp Lata. English for Effective Communication, Oxford University Press, 2015.
- [2] Konar, Nira. English Language Laboratories –A Comprehensive Manual, PHI Learning Pvt. Ltd., 2011



**Prerequisites:**

- Knowledge on computer networks

**Course Objectives:**

- To impart knowledge on network security issues, services, goals and mechanisms
- To classify various algorithms and fundamental ideas of public-key cryptography
- To explain the authentication codes and digital signatures
- To analyze the security of communication systems, networks and protocols

**Course Outcomes:**

- Demonstrate the knowledge of cryptography and network security concepts and applications
- Ability to compare various cryptographic algorithms
- Ability to identify the requirement of Message Authentication and hash functions
- Ability to apply security principles in networks and mobile devices
- Ability to analyze various E-Mail privacy techniques

**UNIT I**

**Introduction to Cryptography:** Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security Cryptography Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks.

**UNIT II**

**Symmetric and Asymmetric key Ciphers:** Block Cipher principles, DES, AES, Blowfish, RC5, IDEA, Differential and Linear Cryptanalysis, Block cipher modes of operation, Stream ciphers, RC4. Principles of public key cryptosystems, RSA algorithm, Elgamal Cryptography, Diffie-Hellman Key Exchange, Knapsack Algorithm.

**UNIT III**

**Message Authentication Algorithms and Hash Functions:** Message Authentication, MD5, Secure Hash Algorithm (SHA512), Message authentication codes: Authentication requirements, HMAC, CMAC, Digital signatures, Elgamal Digital Signature Scheme Symmetric Key Distribution Using Symmetric & Asymmetric Encryption, Distribution of Public Keys, Kerberos, X.509 Authentication Service, Public – Key Infrastructure.

**UNIT IV**

**Transport-level and Wireless Network Security:** Web security considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction (SET), HTTPS, Secure Shell (SSH), Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security.

## UNIT V

**E-Mail and IP Security:** Pretty Good Privacy, S/MIME ,IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, Internet Key Exchange

**Case Studies on Cryptography and Security:** Secure Multiparty Calculation, Virtual Elections, Single sign On, Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability.

### TEXT BOOKS:

- [1] Cryptography and Network Security- Principles and Practice: William Stallings, Pearson Education, 6th Edition
- [2] Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition

### REFERENCE BOOKS:

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

**Prerequisites:**

- Knowledge on database systems

**Course objectives:**

- To describe the concepts related to data warehousing, On-Line Analytical Processing (OLAP).
- To understand stages in building a Data Mining.
- To analyze and evaluate performance of algorithms for Association Rules.
- To analyze classification algorithms and evaluate prediction techniques.
- To describe methods for data-clustering approaches.

**Course outcomes:**

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

**UNIT- I:**

**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- II:**

**Introduction to data mining and its issues:** Data, Types of data, Need for data mining, KDD process, Transactional databases, Data Mining functionalities, Applications, Classification of data mining systems, Data Mining Task Primitives, Major issues in Data Mining.

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

**UNIT- III:**

**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- IV**

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

## **UNIT-V**

**Clustering:** Types of Data in Cluster Analysis, Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, Density based method, Strengths and Weaknesses, Outlier Analysis.

### **TEXT BOOKS:**

- [1] Data Mining – Concepts and Techniques - Jiawei Han & Micheline Kamber, 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

**B.Tech III Year II Semester****Course Objectives:**

- Automata and compiler Design mainly deals with the languages which are formal and regular and also deals with grammar present in the machine.
- A compiler is a program that accepts a program in source language and converts into a machine understandable format.
- The push down automata is the major one it's a five tuple set containing states, alphabets, transition function and accept states.

**Course Outcomes:**

- Graduate should be able to understand the concept of abstract machines and their power to recognize the languages.
- Attains the knowledge of language classes & grammars relationship among them with the help of Chomsky hierarchy.
- Ability to understand the design of a compiler given features of the languages.
- Ability to implement practical aspects of automata theory.
- Gain knowledge of powerful compiler generation tools.

**UNIT-I**

**Formal Language and Regular Expressions:** Languages, Definition Languages regular expressions, Finite Automata – DFA, NFA. Conversion of regular expression to NFA, NFA to DFA. Applications of Finite Automata to lexical analysis, lex tools.

**Context Free grammars and parsing:** Context free grammars, derivation, parse trees, ambiguity

**UNIT-II**

**Top-Down Parsing:** Recursive descent Parsing, LL (K) grammars and LL (1) parsing.

**Bottom-up parsing:** handle pruning LR Grammar Parsing, LALR parsing, parsing ambiguous grammars, YACC programming specification.

**Semantics:** Syntax directed translation, S-attributed and L-attributed grammars, Intermediate code – abstract syntax tree, translation of simple statements and control flow statements.

**UNIT-III**

**Context Sensitive features** – Chomsky hierarchy of languages and recognizers. Type checking, type conversions, equivalence of type expressions, overloading of functions and operations.

**UNIT-IV**

**Run time storage:** Storage organization, storage allocation strategies scope access to now local names, parameters, language facilities for dynamics storage allocation, symbol table management.

**Code optimization:** Principal sources of optimization, optimization of basic blocks, peephole optimization, flow graphs, Data flow analysis of flow graphs.

**UNIT-V**

**Code generation:** Machine dependent code generation, object code forms, generic code generation algorithm, Register allocation and assignment. Using DAG representation of Block.

**TEXT BOOKS:**

- [1] Introduction to Theory of computation. Sipser, 2nd Edition, Thomson.
- [2] Compilers Principles, Techniques and Tools Aho, Ullman, Ravisethi, Pearson Education.

**REFERENCES:**

- [1] Modern Compiler Construction in C, Andrew W.Appel Cambridge University Press. Compiler Construction, LOUDEN, Thomson.
- [2] Elements of Compiler Design, A. Meduna, Auerbach Publications, Taylor and Francis Group.
- [3] Principles of Compiler Design, V. Raghavan, TMH.
- [4] Engineering a Compiler, K. D. Cooper, L. Torczon, ELSEVIER.
- [5] Introduction to Formal Languages and Automata Theory and Computation - Kamala Krithivasan and Rama R, Pearson.
- [6] Modern Compiler Design, D. Grune and others, Wiley-India.
- [7] A Text book on Automata Theory, S. F. B. Nasir, P. K. Srimani, Cambridge Univ. Press.
- [8] Automata and Language, A. Meduna, Springer.

**Pre-requisites:**

- A course on “Computer Networks”
- A course on “Operating Systems”

**Course Objectives:**

- To explain the evolving computer model called cloud computing.
- To introduce the various levels of services that can be achieved by cloud.
- To describe the security aspects in cloud.

**Course Outcomes:**

- Ability to define the different types of Computing models and their uses
- Ability to explain characteristics of Cloud Computing
- Ability to analyze and illustrate the cloud computing architecture
- Ability to classify the types of Services and explain its advantages and disadvantages
- Ability to Summarize all Cloud service providers features.

**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, EMC IT, 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, Sales force, Sales Cloud, Service Cloud: Knowledge as a Service, Rack space, VMware, Manjra soft, Aneka Platform

### **TEXT BOOKS:**

- [1] Essentials of cloud Computing: K. Chandrashekar, CRC press, 2014.

### **REFERENCE BOOKS :**

- [1] Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and 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 Kumaraswamy, Shahed Latif, O'Reilly, SPD, rp2011



**B.Tech. III Year II Semester**

**Prerequisites:**

- A course on “Digital Logic Design”
- A course on “Computer Organization and Architecture”

**Course Objectives:**

- To provide an overview of principles of Embedded System
- To understand the core of an embedded system
- To provide a clear understanding of role of firmware, operating systems in correlation with hardware systems.
- To understands the RTOS and inter-process communication.

**Course Outcomes:**

Upon successful completion of this course, students will be able to:

- Understand the selection procedure of processors in the embedded domain.
- Understand Embedded Firmware design approaches
- Visualize the role of real-time operating systems in embedded systems.
- Evaluate the correlation between task synchronization and latency issues
- Learn the Embedded System Development Environment.

**UNIT – I**

**Introduction to Embedded Systems:** Definition of Embedded System, Embedded Systems Vs General Computing Systems, History of Embedded Systems, Classification of Embedded Systems, Major application areas, Purpose of Embedded Systems, Characteristics and Quality attributes of Embedded Systems.

**UNIT – II**

**The Typical Embedded System:**Core of the Embedded System, Memory, Sensors and Actuators, Communication Interface, Embedded Firmware, Other System components.

**UNIT – III**

**Embedded Firmware Design and Development:** Embedded Firmware Design approaches, Embedded Firmware Design, Embedded Firmware Development Languages, and Programming in Embedded C.

**UNIT – IV**

**RTOS Based Embedded System Design:** Operating System basics, Types of Operating Systems, Tasks, Process, Threads, Multiprocessing and Multi-tasking, Task Scheduling, Threads-Processes-Scheduling putting them together, Task Communication, Task Synchronization, Device Drivers, How to choose an RTOS.

## UNIT – V

**Integration and Testing of Embedded Hardware and Firmware:** Integration of Hardware and Firmware, Boards Power up.

**The Embedded System Development Environment:** The Integrated Development Environment (IDE), Types of files generated on Cross-Compilation, Disassembler/Decompiler, Simulators, Emulators and Debugging, Target Hardware Debugging, Boundary Scan.

### TEXT BOOK:

[1] Shibu K V, “Introduction to Embedded Systems”, Second Edition, Mc Graw Hill

### REFERENCES:

[1] Rajkamal, Embedded Systems Architecture, Programming and Design, TATA McGraw-Hill

[2] Frank Vahid and Tony Givargis, “Embedded Systems Design” - A Unified Hardware/Software Introduction, John Wiley

[3] David E.Simon, an Embedded Software Primer, Pearson Education Asia, First Indian Reprint 2000.

[4] Lyla B Das, Embedded Systems- An integrated approach, Pearson education 2012.

**Course Objectives:**

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

**Course Outcomes:**

- Ability to understand IR principles to locate relevant information in large collections of data
- Ability to understand information extraction using indexing, and various data structure algorithms.
- Ability to design different document clustering algorithms and understand automatic indexing.
- Ability to use various search algorithms and perform information visualization.
- Ability to understand the ways to design an Information Retrieval System for web search tasks.

**UNIT – I**

**Introduction to Information Retrieval Systems:** Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses

**Information Retrieval System Capabilities:** Search Capabilities, Browse Capabilities, Miscellaneous Capabilities

**UNIT - II**

**Cataloging and Indexing:** History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction

**Data Structure:** Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models

**UNIT - III**

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

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

**UNIT - IV**

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

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

## **UNIT - V**

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

**Multimedia Information Retrieval:** Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval, Web search basics. Web crawling and indexes.

### **TEXT BOOK:**

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

### **REFERENCE BOOKS:**

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

**Prerequisites:**

- A course on “Software Engineering”

**Course Objectives:**

- To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
- To develop skills in performing manual testing.
- To develop skills in software test automation and management using latest tools.
- To provide knowledge in taking decision of when to use automation testing and manual testing based on the context to be tested.

**Course Outcomes:**

- Acquire knowledge on STLC phases and consequences of bugs.
- Acquire skills to perform various testing techniques.
- Acquire knowledge on logic based testing and regular expressions.
- Demonstrate the test planning and management.
- Ability to develop skills in software test automation and management using latest tools.

**UNIT -I:**

**Introduction:** Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs, STLC (software testing life cycle) phases.

**Flow graphs and Path testing:** - Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

**UNIT -II:**

**Integration Testing:** Integration Testing as a Type of Testing, Integration Testing as a Phase of Testing, Scenario Testing, Defect Bash.

**System and Acceptance Testing:** Overview, Functional Versus Non-Functional, Functional System Testing & Non-Functional, Acceptance Testing.

**Dataflow testing:** - Basics of data flow testing, strategies in data flow testing, application of dataflow testing.

**Regression Testing:** Introduction, Types, When to do Regression testing, how to do Regression Testing, Best Practices in Regression Testing.

**UNIT -III:**

**White Box Testing:** Static Testing, Structural Testing, Challenges; **Black Box Testing** and

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

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

**UNIT -IV**

**State, State Graphs and Transition testing:** State graphs, good & bad state graphs, state testing, Testability tips, Automation Testing versus Manual Flow Testing.

**Test Planning, Management, Execution and Reporting:** Introduction, Planning, Management, Process, and Reporting, Best Practices.

## **UNIT -V**

**Software Test Automation:** Terms used in Automation, Skills needed for Automation, What to Automate, Scope of Automation, Design and Architecture for Automation, Generic Requirements for Test Tools, Process Model for Automation, Selecting a Test Tool, Automation for Extreme Programming Model, Challenges.

**Test Metrics and Measurements:** Metrics & Measurements, Types, Project, Progress, Productivity, Release

### **TEXT BOOKS:**

- [1] Software Testing techniques - BarisBeizer, Dreamtech, second edition.
- [2] Software Testing Tools – Dr.K.V.K.K.Prasad, Dreamtech.
- [3] Srinivasa Desikan & Gopalaswamy Ramesh, “Software Testing – Principles and Practices”, Pearson Education, 2007.

### **REFERENCE BOOKS:**

- [1] The craft of software testing - Brian Marick, Pearson Education.
- [2] Software Testing Techniques – SPD(Oreille)
- [3] Software Testing in the Real World – Edward Kit, Pearson.
- [4] Effective methods of Software Testing, Perry, John Wiley.
- [5] Art of Software Testing – Meyers, John Wiley.
- [6] “Software Testing – Effective Methods, Tools and Techniques”, RenuRajani, Pradeep Oak, TMK.
- [7] Effective methods of Software Testing, Perry, John Wiley.
- [8] Art of Software Testing – Meyers, John Wiley.

**Course Objectives:**

- To understand basics operations involved in encryption and decryption algorithms.
- To be able to implement Conventional Encryption Algorithms.
- To be able to implement Public Key Cryptography Algorithms.
- To be able to implement Message Authentication Algorithms.
- To be able to implement cryptographic Hash Functions algorithms.

**Course Outcomes:**

- Student will be able to understand basic mathematical operations involved in cryptography.
- Student will be able to implement Conventional Encryption Algorithms i.e. DES, AES, Blowfish, RC4 algorithms.
- Student will be able to implement Public Key Cryptography Algorithms i.e. RSA, Diffie-Hellman algorithms
- Student will be able to implement Message Authentication Algorithms.
- Student will be able implement cryptographic Hash Functions algorithms.

**List of Experiments:**

1. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and displays the result.
2. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a Java program to perform encryption and decryption using the following algorithms a. Ceaser cipher b. Substitution cipher c. Hill Cipher
4. Write a C/JAVA program to implement the DES algorithm logic.
5. Write a C/JAVA program to implement the Blowfish algorithm logic.
6. Write a C/JAVA program to implement the Rijndael algorithm logic.
7. Write the RC4 logic in Java Using Java cryptography; encrypt the text "Hello world" using Blowfish. Create your own key using Java key tool.
8. Write a Java program to implement RSA algorithm.
9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
11. Calculate the message digest of a text using the MD5 algorithm in JAVA.

**TEXT BOOKS:**

- [1] Cryptography and Network Security- Principles and Practice: William Stallings, Pearson Education, 6th Edition
- [2] Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition

**REFERENCE BOOKS:**

- [1] Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition.
- [2] Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 3rd Edition

**Course Code: 19IT3252**

<b>DATA WAREHOUSING AND DATA MINING LAB</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>B.Tech. III Year II Sem.</b>	-	-	<b>3</b>	<b>1.5</b>

**Prerequisites:**

- A Course on “Database systems”

**Course Objectives:**

- To obtain practical experience using data mining techniques on real world data sets.
- Emphasize hands-on experience working with all real data sets.

**Course Objectives:**

- The course is intended to obtain hands-on experience using data mining software.
- Intended to provide practical exposure of the concepts in data mining algorithms
- Implement various data mining functionalities
- Design a data warehouse by using an ETL tool
- To learn recent developments in the data warehousing and On Line Analytical Processing (OLAP) fields

**Course Outcomes:**

- Apply preprocessing statistical methods for any given raw data.
- Implement Multidimensional data models.
- Implement Classification algorithms.
- Implement Association rule mining algorithms, Classification algorithms and Clustering algorithms with modern data mining tool such as WEKA.
- Design small projects with data mining and data warehousing.

**EXPERIMENTS:**

1. Data Processing Techniques: (i) Data cleaning (ii) Data transformation - Normalization (iii)Data integration
2. Implement the following Multidimensional Data Models
  - i. Star Schema
  - ii. Snowflake Schema
  - iii. Fact Constellation
3. Data Extraction, Transformations & Loading operations
4. Implementation of Apriori algorithm
5. Implementation of FP – Growth algorithm
6. Calculating Information gain measures
7. Implement the Decision Tree Induction classification algorithms
8. Implement the KNN classification algorithms
9. Implement K-means clustering algorithms
10. Implement K-medoids clustering algorithms

**TEXT BOOKS:**

- [1] Data Mining – Concepts and Techniques - Jiawei Han & Micheline Kamber, 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 Algorithmsl, Oxford

**GENERAL INSTRUCTIONS**

1. **Pick The Right Topic:** Make sure that you find this topic genuinely interesting, or find some aspect of it that is especially cool.
2. **Skim Your Textbook:** Look over the syllabus, read the newspapers regularly, look through recent issues of relevant journals and magazines, surf the net, watch the Technical news, talk to your classmates and friends to get innovative Ideas.
3. **Narrow down Your Topic:** Many good ideas are wasted because students have a hard time focusing on a narrow enough topics. If your topic is way too broad, try homing in on some part of that topic, and exploring that area in more depth.
4. **My Topic is Too Narrow:** Generalize to similar or related topics (cloning of humans vs. cloning of animals, unexpected social problems that might result from cloning, technical aspects of cloning, moral or religious issues related to cloning, etc.). But keep your focus clear throughout
5. **Organize Your Thoughts In A Good Outline:** Outlining is a genuine pain, especially in the early stages of your paper, by forcing you to come in terms of writing the topic. Try to reveal major deficiencies in your approach. Use it on your first draft to get your bearings, or on your final draft to check the way you've organized your paragraphs.