JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B.Tech. in COMPUTER SCIENCE AND ENGINEERING

III YEAR COURSE STRUCTURE AND SYLLABUS (R18)
Applicable From 2018-19 Admitted Batch

III YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	CS501PC	Formal Languages & Automata Theory	3	0	0	3
2	CS502PC	Software Engineering	3	0	0	3
3	CS503PC	Computer Networks	3	0	0	3
4	CS504PC	Web Technologies	3	0	0	3
5		Professional Elective-I	3	0	0	3
6		Professional Elective -II	3	0	0	3
7	CS505PC	Software Engineering Lab	0	0	3	1.5
8	CS506PC	Computer Networks & Web Technologies Lab	0	0	3	1.5
9	EN508HS	Advanced Communication Skills Lab	0	0	2	1
10	*MC510	Intellectual Property Rights	3	0	0	0
		Total Credits	**21**	**0**	**8**	**22**

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	CS601PC	Machine Learning	3	1	0	4
2	CS602PC	Compiler Design	3	1	0	4
3	CS603PC	Design and Analysis of Algorithms	3	1	0	4
4		Professional Elective – III	3	0	0	3
5		Open Elective-I	3	0	0	3
6	CS604PC	Machine Learning Lab	0	0	3	1.5
7	CS605PC	Compiler Design Lab	0	0	3	1.5
8		Professional Elective-III Lab	0	0	2	1
9	*MC609	Environmental Science	3	0	0	0
		Total Credits	**18**	**3**	**8**	**22**

*MC - Environmental Science – Should be Registered by Lateral Entry Students Only.

Note: Industrial Oriented Mini Project/ Summer Internship is to be carried out during the summer vacation between 6th and 7th semesters. Students should submit report of Industrial Oriented Mini Project/ Summer Internship for evaluation.

Professional Elective-I

CS511PE	Information Theory & Coding
CS512PE	Advanced Computer Architecture
CS513PE	Data Analytics
CS514PE	Image Processing
CS515PE	Principles of Programming Languages

Professional Elective - II

CS521PE	Computer Graphics

CS522PE	Advanced Operating Systems
CS523PE	Informational Retrieval Systems
CS524PE	Distributed Databases
CS525PE	Natural Language Processing

Professional Elective - III

CS611PE	Concurrent Programming
CS612PE	Network Programming
CS613PE	Scripting Languages
CS614PE	Mobile Application Development
CS615PE	Software Testing Methodologies

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

CS501PC: FORMAL LANGUAGES AND AUTOMATA THEORY

III Year B.Tech. CSE I-Sem

	L	T	P	C
	3	0	0	3

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

Course Outcomes
1. Able to understand the concept of abstract machines and their power to recognize the languages.
2. Able to employ finite state machines for modeling and solving computing problems.
3. Able to design context free grammars for formal languages.
4. Able to distinguish between decidability and undecidability.
5. Able to gain proficiency with mathematical tools and formal methods.

UNIT - I
Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.
Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.
Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA, Moore and Melay machines

UNIT - II
Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.
Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma.
Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT - III
Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Tress, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.
Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

UNIT - IV
Normal Forms for Context- Free Grammars: Eliminating useless symbols, Eliminating €-Productions. Chomsky Normal form Griebech Normal form.
Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications
Closure Properties of Context-Free Languages: Closure properties of CFL's, Decision Properties of CFL's

Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

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

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

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

CS502PC: SOFTWARE ENGINEERING

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Course Objectives
1. The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
2. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

Course Outcomes
1. Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

UNIT - I
Introduction to Software Engineering: The evolving role of software, changing nature of 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, the 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.
System models: Context models, behavioral models, data models, object models, structured methods.

UNIT - III
Design Engineering: Design process and design quality, design concepts, the design model.
Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT - IV
Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.
Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT - V
Metrics for Process and Products: Software measurement, metrics for software quality.
Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

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

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, Pearson Education.

REFERENCES:
1. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.

CS503PC: COMPUTER NETWORKS

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Prerequisites
1. A course on "Programming for problem solving"
2. A course on "Data Structures"

Course Objectives
1. The objective of the course is to equip the students with a general overview of the concepts and fundamentals of computer networks.
2. Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

Course Outcomes
1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

UNIT - I
Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.
Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.

UNIT - II
Data link layer: Design issues, framing, Error detection and correction.
Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.
Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.
Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

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

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

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

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

REFERENCE BOOKS:
1. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education
2. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

CS504PC: WEB TECHNOLOGIES

III Year B.Tech. CSE I-Sem

	L	T	P	C
	3	0	0	3

Course Objectives:
1. To introduce PHP language for server-side scripting
2. To introduce XML and processing of XML Data with Java
3. To introduce Server-side programming with Java Servlets and JSP
4. To introduce Client-side scripting with Javascript and AJAX.

Course Outcomes
1. gain knowledge of client-side scripting, validation of forms and AJAX programming
2. understand server-side scripting with PHP language
3. understand what is XML and how to parse and use **XML** Data with Java
4. To introduce Server-side programming with Java Servlets and JSP

UNIT- I
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.

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

UNIT - III
Introduction to Servlets: Common Gateway Interface (CGt), Life cycle 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, Using Cookies and session for session tracking, connecting to database in JSP.

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

TEXT BOOKS:
1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP — Steven Holzner, Tata McGraw-Hill

REFERENCE BOOKS
1. Web Programming, building internet applications, Chris Bates 2″ edition, Wiley Dreamtech
2. Java Server Pages —Hans Bergsten, SPD O'Reilly,
3. Java Script, D.Flanagan
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.

CS511PE: INFORMATION THEORY & CODING (Professional Elective - I)

III Year B.Tech. CSE I-Sem

	L	T	P	C
	3	0	0	3

Prerequisite
1. Digital Communications

Course Objectives:
1. To acquire the knowledge in measurement of information and errors.
2. Understand the importance of various codes for communication systems
3. To design encoder and decoder of various codes.
4. To known the applicability of source and channel codes

Course Outcomes: Upon completing this course, the student will be able to
1. Learn measurement of information and errors.
2. Obtain knowledge in designing various source codes and channel codes
3. Design encoders and decoders for block and cyclic codes
4. Understand the significance of codes in various applications

UNIT - I
Coding for Reliable Digital Transmission and storage
Mathematical model of Information, A Logarithmic Measure of Information, Average and Mutual Information and Entropy, Types of Errors, Error Control Strategies.
Source Codes: Shannon-fano coding, Huffman coding

UNIT - II
Linear Block Codes: Introduction to Linear Block Codes, Syndrome and Error Detection, Minimum Distance of a Block code, Error-Detecting and Error-correcting Capabilities of a Block code, Standard array and Syndrome Decoding, Probability of an undetected error for Linear Codes over a BSC, Hamming Codes. Applications of Block codes for Error control in data storage system

UNIT - III
Cyclic Codes: Description, Generator and Parity-check Matrices, Encoding, Syndrome Computation and Error Detection, Decoding, Cyclic Hamming Codes, shortened cyclic codes, Error-trapping decoding for cyclic codes, Majority logic decoding for cyclic codes.

UNIT - IV
Convolutional Codes: Encoding of Convolutional Codes- Structural and Distance Properties, state, tree, trellis diagrams, maximum likelihood decoding, Sequential decoding, Majority- logic decoding of Convolution codes. Application of Viterbi Decoding and Sequential Decoding, Applications of Convolutional codes in ARQ system.

UNIT - V
BCH Codes: Minimum distance and BCH bounds, Decoding procedure for BCH codes, Syndrome computation and iterative algorithms, Error locations polynomials for single and double error correction.

TEXT BOOKS
1. Error Control Coding- Fundamentals and Applications –Shu Lin, Daniel J.Costello,Jr, Prentice Hall, Inc 2014.
2. Error Correcting Coding Theory-Man Young Rhee, McGraw – Hill Publishing 1989

REFERENCE BOOKS

1. Digital Communications- John G. Proakis, 5th ed., , TMH 2008.
2. Introduction to Error Control Codes-Salvatore Gravano-oxford
3. Error Correction Coding – Mathematical Methods and Algorithms – Todd K.Moon, 2006, Wiley India.
4. Information Theory, Coding and Cryptography – Ranjan Bose, 2nd Edition, 2009, TMH.

CS512PE: ADVANCED COMPUTER ARCHITECTURE (Professional Elective - I)

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Prerequisites: 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 design modern computer systems

Course Outcomes: Gain knowledge of
1. Computational models and Computer Architectures.
2. Concepts of parallel computer models.
3. Scalable Architectures, Pipelining, Superscalar processors, multiprocessors

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
Principals 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 Multicomputers, Multiprocessor system interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputers, Message-passing Mechanisms, Multivetor 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, Principals of Multithreading, Fine-Grain Multicomputers, Scalable and multithreaded Architectures, Dataflow and hybrid Architectures.

TEXT BOOK:
1. Advanced Computer Architecture Second Edition, Kai Hwang, Tata McGraw Hill Publishers.

REFERENCE BOOKS:
1. Computer Architecture, Fourth edition, J. L. Hennessy and D.A. Patterson. ELSEVIER.

2. Advanced Computer Architectures, S.G. Shiva, Special Indian edition, CRC, Taylor &Francis.
3. Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.
4. Advanced Computer Architecture, D. Sima, T. Fountain, P. Kacsuk, Pearson education.
5. Computer Architecture, B. Parhami, Oxford Univ. Press.

CS513PE: DATA ANALYTICS (Professional Elective - I)

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Prerequisites
1. A course on "Database Management Systems".
2. Knowledge of probability and statistics.

Course Objectives:
1. To explore the fundamental concepts of data analytics.
2. To learn the principles and methods of statistical analysis
3. Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms.
4. To understand the various search methods and visualization techniques.

Course Outcomes: After completion of this course students will be able to
1. Understand the impact of data analytics for business decisions and strategy
2. Carry out data analysis/statistical analysis
3. To carry out standard data visualization and formal inference procedures
4. Design Data Architecture
5. Understand various Data Sources

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

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

UNIT - III
Regression – Concepts, Blue property assumptions, Least Square Estimation, Variable Rationalization, and Model Building etc.
Logistic Regression: Model Theory, Model fit Statistics, Model Construction, Analytics applications to various Business Domains etc.

UNIT - IV
Object Segmentation: Regression Vs Segmentation – Supervised and Unsupervised Learning, Tree Building – Regression, Classification, Overfitting, Pruning and Complexity, Multiple Decision Trees etc.
Time Series Methods: Arima, Measures of Forecast Accuracy, STL approach, Extract features from generated model as Height, Average Energy etc and Analyze for prediction

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

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

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

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

CS514PE: IMAGE PROCESSING (Professional Elective - I)

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

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

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

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

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

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

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

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

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

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

REFERENCES:
1. Fundamentals of Digital Image Processing: **A. K. Jain** , PHI.

2. Digital Image Processing using MAT LAB: Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins: Pearson Education India, 2004.
3. Digital Image Processing: William K. Pratt, John Wilely, 3rd Edition, 2004.

CS515PE: PRINCIPLES OF PROGRAMMING LANGUAGES (Professional Elective - I)

III Year B.Tech. CSE 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. Introduce important paradigms of programming languages
2. To provide conceptual understanding of high-level language design and implementation
3. 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
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

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

UNIT - II
Names, Bindings, and Scopes: Introduction, Names, Variables, Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants
Data Types: Introduction, Primitive Data Types, Character String Types, User Defined Ordinal Types, Array, Associative Arrays, Record, Union, Tuple Types, List Types, Pointer and Reference Types, Type Checking, Strong Typing, Type Equivalence
Expressions and Statements: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment
Control Structures – Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

UNIT - III
Subprograms and Blocks: Fundamentals of Sub-Programs, Design Issues for Subprograms, Local Referencing Environments, Parameter Passing Methods, Parameters that Are Subprograms, Calling Subprograms Indirectly, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User Defined Overloaded Operators, Closures, Coroutines
Implementing Subprograms: General Semantics of Calls and Returns, Implementing Simple Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping
Abstract Data Types: The Concept of Abstraction, Introductions to Data Abstraction, Design Issues, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulations

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

UNIT - V
Functional Programming Languages: Introduction, Mathematical Functions, Fundamentals of Functional Programming Language, LISP, Support for Functional Programming in Primarily Imperative Languages, Comparison of Functional and Imperative Languages
Logic Programming Language: Introduction, an Overview of Logic Programming, Basic Elements of Prolog, Applications of Logic Programming.
Scripting Language: Pragmatics, Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library. (Text Book 2)

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

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

CS521PE/IT525PE: COMPUTER GRAPHICS (Professional Elective - II)

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Prerequisites
1. Familiarity with the theory and use of coordinate geometry and of linear algebra such as matrix multiplication.
2. A course on "Computer Programming and Data Structures"

Course Objectives
1. The aim of this course is to provide an introduction of fundamental concepts and theory of computer graphics.
2. Topics covered include 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
1. Acquire familiarity with the relevant mathematics of computer graphics.
2. Be able to design basic graphics application programs, including animation
3. Be able to design applications that display graphic images to given specifications

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), mid-point 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, Bezier and B-Spline surfaces. 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

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.

CS522PE: ADVANCED OPERATING SYSTEMS (Professional Elective - II)

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

Course Objectives
1. To study, learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems)
2. Hardware and software features that support these systems.

Course Outcomes
1. Understand the design approaches of advanced operating systems
2. Analyze the design issues of distributed operating systems.
3. Evaluate design issues of multi processor operating systems.
4. Identify the requirements Distributed File System and Distributed Shared Memory.
5. Formulate the solutions to schedule the real time applications.

UNIT - I
Architectures of Distributed Systems: System Architecture Types, Distributed Operating Systems, Issues in Distributed Operating Systems, Communication Primitives. **Theoretical Foundations:** Inherent Limitations of a Distributed System, Lamport's Logical Clocks, Vector Clocks, Causal Ordering of Messages, Termination Detection.

UNIT - II
Distributed Mutual Exclusion: The Classification of Mutual Exclusion Algorithms, **Non-Token – Based Algorithms:** Lamport's Algorithm, The Ricart-Agrawala Algorithm, Maekawa's Algorithm, **Token-Based Algorithms:** Suzuki-Kasami's Broadcast Algorithm, Singhal's Heurisric Algorithm, Raymond's Heuristic Algorithm.

UNIT - III
Distributed Deadlock Detection: Preliminaries, Deadlock Handling Strategies in Distributed Systems, Issues in Deadlock Detection and Resolution, Control Organizations for Distributed Deadlock Detection, Centralized- Deadlock – Detection Algorithms, Distributed Deadlock Detection Algorithms, Hierarchical Deadlock Detection Algorithms

UNIT - IV
Multiprocessor System Architectures: Introduction, Motivation for multiprocessor Systems, Basic Multiprocessor System Architectures **Multi Processor Operating Systems**: Introduction, Structures of Multiprocessor Operating Systems, Operating Design Issues, Threads, Process Synchronization, Processor Scheduling.
Distributed File Systems: Architecture, Mechanisms for Building Distributed File Systems, Design Issues

UNIT - V
Distributed Scheduling: Issues in Load Distributing, Components of a Load Distributed Algorithm, Stability, Load Distributing Algorithms, Requirements for Load Distributing, Task Migration, Issues in task Migration
Distributed Shared Memory: Architecture and Motivation, Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues

TEXT BOOK:
1. Advanced Concepts in Operating Systems, Mukesh Singhal, Niranjan G. Shivaratri, Tata McGraw-Hill Edition 2001

REFERENCES:
1. Distributed Systems: Andrew S. Tanenbaum, Maarten Van Steen, Pearson Prentice Hall, Edition – 2, 2007

CS523PE: INFORMATION RETRIEVAL SYSTEMS (Professional Elective - II)

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Prerequisites:
1. Data Structures

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

Course Outcomes:
1. Ability to apply IR principles to locate relevant information large collections of data
2. Ability to design different document clustering algorithms
3. Implement retrieval systems for web search tasks.
4. Design an Information Retrieval System for web search tasks.

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

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

UNIT - III
Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages
Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters

UNIT - IV
User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems, Searching the INTERNET and Hypertext
Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies

UNIT - V
Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems
Multimedia Information Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval

TEXT BOOK

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

REFERENCES

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.

CS524PE: DISTRIBUTED DATABASES (Professional Elective - II)

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Prerequisites
1. A course on "Database Management Systems"

Course Objectives:
1. 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.
2. Introduce basic principles and implementation techniques of distributed database systems.
3. Equip students with principles and knowledge of parallel and object-oriented databases.
4. Topics include distributed DBMS architecture and design; query processing and optimization; distributed transaction management and reliability; parallel and object database management systems.

Course Outcomes:
1. Understand theoretical and practical aspects of distributed database systems.
2. Study and identify various issues related to the development of distributed database system.
3. Understand the design aspects of object-oriented database system and related development.

UNIT - I
Introduction; Distributed Data Processing, Distributed Database System, Promises of DDBSs, Problem areas.
Distributed DBMS Architecture: Architectural Models for Distributed DBMS, DDMBS Architecture.
Distributed Database Design: Alternative Design Strategies, Distribution Design issues, Fragmentation, Allocation.

UNIT - II
Query processing and decomposition: Query processing objectives, characterization of query processors, layers of query processing, query decomposition, localization of distributed data.
Distributed query Optimization: Query optimization, centralized query optimization, distributed query optimization algorithms.

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

UNIT - IV
Distributed DBMS Reliability: Reliability concepts and measures, fault-tolerance in distributed systems, failures in Distributed DBMS, local & distributed reliability protocols, site failures and network partitioning.
Parallel Database Systems: Parallel database system architectures, parallel data placement, parallel query processing, load balancing, database clusters.

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

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

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

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

CS525PE: NATURAL LANGUAGE PROCESSING (Professional Elective - II)

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Prerequisites: Data structures, finite automata and probability theory

Course Objectives
1. Introduce to some of the problems and solutions of NLP and their relation to linguistics and statistics.

Course Outcomes
1. Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.
2. Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems
3. Able to manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.
4. Able to design, implement, and analyze NLP algorithms
5. Able to design different language modeling Techniques.

UNIT - I
Finding the Structure of Words: Words and Their Components, Issues and Challenges, Morphological Models
Finding the Structure of Documents: Introduction, Methods, Complexity of the Approaches, Performances of the Approaches

UNIT - II
Syntax Analysis: Parsing Natural Language, Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms, Models for Ambiguity Resolution in Parsing, Multilingual Issues

UNIT - III
Semantic Parsing: Introduction, Semantic Interpretation, System Paradigms, Word Sense Systems, Software.

UNIT - IV
Predicate-Argument Structure, Meaning Representation Systems, Software.

UNIT - V
Discourse Processing: Cohension, Reference Resolution, Discourse Cohension and Structure
Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Parameter Estimation, Language Model Adaptation, Types of Language Models, Language-Specific Modeling Problems, Multilingual and Crosslingual Language Modeling

TEXT BOOKS:
1. Multilingual natural Language Processing Applications: From Theory to Practice –   Daniel M. Bikel and Imed Zitouni, Pearson Publication
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary

REFERENCE:
1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications

CS505PC: SOFTWARE ENGINEERING LAB

III Year B.Tech. CSE I-Sem	**L**	**T**	**P**	**C**
	0	**0**	**3**	**1.5**

Prerequisites
1. A course on "Programming for Problem Solving"

Co-requisite
1. A Course on "Software Engineering"

Course Objectives
1. 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
1. Ability to translate end-user requirements into system and software requirements
2. Ability to generate a high-level design of the system from the software requirements
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

List of Experiments
Do the following 8 exercises for any two projects given in the list of sample projects or any other projects:
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) Study and usage of any Design phase CASE tool
5) Performing the Design by using any Design phase CASE tools.
6) Develop test cases for unit testing and integration testing
7) Develop test cases for various white box and black box testing techniques.

Sample Projects:
1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

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, Pearson Education.

CS506PC: COMPUTER NETWORKS AND WEB TECHNOLOGIES LAB

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Course Objectives
1. To understand the working principle of various communication protocols.
2. To understand 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. Implement data link layer farming methods
2. Analyze error detection and error correction codes.
3. Implement and analyze routing and congestion issues in network design.
4. Implement Encoding and Decoding techniques used in presentation layer
5. To be able to work with different network tools

List of Experiments
1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijsktra's algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting technique used in buffers.
10. **Wireshark**
    i.   Packet Capture Using Wire shark
    ii.   Starting Wire shark
    iii. Viewing Captured Traffic
    iv.   Analysis and Statistics & Filters.
11. How to run Nmap scan
12. Operating System Detection using Nmap
13. Do the following using NS2 Simulator
    i.   NS2 Simulator-Introduction
    ii.   Simulate to Find the Number of Packets Dropped
    iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv.   Simulate to Find the Number of Packets Dropped due to Congestion
    v.   Simulate to Compare Data Rate& Throughput.
    vi.   Simulate to Plot Congestion for Different Source/Destination
    vii. Simulate to Determine the Performance with respect to Transmission of Packets

Web Technologies Experiments
1. Write a PHP script to print prime numbers between 1-50.
2. PHP script to
    a. Find the length of a string.
    b. Count no of words in a string.
    c. Reverse a string.
    d. Search for a specific string.

3. Write a PHP script to merge two arrays and sort them as numbers, in descending order.
4. Write a PHP script that reads data from one file and write into another file.
5. Develop static pages (using 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
6. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.
7. Create and save an XML document on the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.
8. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
9. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

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, 8th Edition
5. Web technologies, Black Book, Dreamtech press.
6. Gopalan N.P. and Akilandeswari J., "Web Technology", Prentice Hall of India

EN508HS: ADVANCED COMMUNICATION SKILLS LAB

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

1. INTRODUCTION:

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

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

- Gathering ideas and information to organize ideas relevantly and coherently.
- Engaging in debates.
- Participating in group discussions.
- Facing interviews.
- Writing project/research reports/technical reports.
- Making oral presentations.
- Writing formal letters.
- Transferring information from non-verbal to verbal texts and vice-versa.
- Taking part in social and professional communication.

2. OBJECTIVES:

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

1. To improve the students' fluency in English, through a well-developed vocabulary and enable them to listen to English spoken at normal conversational speed by educated English speakers and respond appropriately in different socio-cultural and professional contexts.
2. Further, they would be required to communicate their ideas relevantly and coherently in writing.
3. To prepare all the students for their placements.

3. SYLLABUS:

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

1. **Activities on Fundamentals of Inter-personal Communication and Building Vocabulary** - Starting a conversation – responding appropriately and relevantly – using the right body language – Role Play in different situations & Discourse Skills- using visuals - Synonyms and antonyms, word roots, one-word substitutes, prefixes and suffixes, study of word origin, business vocabulary, analogy, idioms and phrases, collocations & usage of vocabulary.
2. **Activities on Reading Comprehension** –General Vs Local comprehension, reading for facts, guessing meanings from context, scanning, skimming, inferring meaning, critical reading& effective googling.
3. **Activities on Writing Skills** – Structure and presentation of different types of writing – *letter writing/Resume writing/ e-correspondence/Technical report writing/* – planning for writing – improving one's writing.
4. **Activities on Presentation Skills** – Oral presentations (individual and group) through JAM sessions/seminars/**PPTs** and written presentations through posters/projects/reports/ e-mails/assignments etc.
5. **Activities on Group Discussion and Interview Skills** – Dynamics of group discussion, intervention, summarizing, modulation of voice, body language, relevance, fluency and organization of ideas and rubrics for evaluation- Concept and process, pre-interview planning, opening strategies, answering strategies, interview through tele-conference & video-conference and Mock Interviews.

4. MINIMUM REQUIREMENT:

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

- Spacious room with appropriate acoustics.
- Round Tables with movable chairs
- Audio-visual aids
- LCD Projector
- Public Address system
- P – IV Processor, Hard Disk – 80 GB, RAM–512 MB Minimum, Speed – 2.8 GHZ
- T. V, a digital stereo & Camcorder
- Headphones of High quality

5. SUGGESTED SOFTWARE:

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

- Oxford Advanced Learner's Compass, 7th Edition
- DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.
- Lingua TOEFL CBT Insider, by Dream tech
- TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)

TEXT BOOKS:

1. Effective Technical Communication by M Asharaf Rizvi. McGraw Hill Education (India) Pvt. Ltd. 2nd Edition
2. Academic Writing: A Handbook for International Students by Stephen Bailey, Routledge, 5th Edition.

REFERENCES:

1. Learn Correct English – A Book of Grammar, Usage and Composition by Shiv K. Kumar and Hemalatha Nagarajan. Pearson 2007
2. Professional Communication by Aruna Koneru, McGraw Hill Education (India) Pvt. Ltd, 2016.
3. Technical Communication by Meenakshi Raman & Sangeeta Sharma, Oxford University Press 2009.
4. Technical Communication by Paul V. Anderson. 2007. Cengage Learning pvt. Ltd. New Delhi.
5. English Vocabulary in Use series, Cambridge University Press 2008.
6. Handbook for Technical Communication by David A. McMurrey & Joanne Buckley. 2012. Cengage Learning.
7. Communication Skills by Leena Sen, PHI Learning Pvt Ltd., New Delhi, 2009.
8. Job Hunting by Colm Downes, Cambridge University Press 2008.
9. English for Technical Communication for Engineering Students, Aysha Vishwamohan, Tata Mc Graw-Hill 2009.

*MC510: INTELLECTUAL PROPERTY RIGHTS

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

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 copy rights: Fundamental of copy right law, originality of material, rights of reproduction, rights to perform the work publicly, copy right ownership issues, copy right registration, notice of copy right, international copy right law.
Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

UNIT – IV
Trade Secrets: Trade secrete law, determination of trade secrete status, liability for misappropriations of trade secrets, protection for submission, trade secrete litigation.
Unfair competition: Misappropriation right of publicity, false advertising.

UNIT – V
New development of intellectual property: new developments in trade mark law; copy right law, patent law, intellectual property audits.
International overview on intellectual property, international – trade mark law, copy right law, international patent law, and international development in trade secrets law.

TEXT BOOKS & REFERENCES:
1. Intellectual property right, Deborah. E. Bouchoux, Cengage learning.

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

CS601PC: MACHINE LEARNING

III Year B.Tech. CSE II-Sem

L	T	P	C
3	1	0	4

Prerequisites
1. Data Structures
2. Knowledge on statistical methods

Course Objectives
1. This course explains machine learning techniques such as decision tree learning, Bayesian learning etc.
2. To understand computational learning theory.
3. To study the pattern comparison techniques.

Course Outcomes
1. Understand the concepts of computational intelligence like machine learning
2. Ability to get the skill to apply machine learning techniques to address the real time problems in different areas
3. Understand the Neural Networks and its usage in machine learning application.

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**.**

Decision Tree Learning – Introduction, decision tree representation, appropriate problems for decision tree learning, the basic decision tree learning algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning.

UNIT - II

Artificial Neural Networks-1– Introduction, neural network representation, appropriate problems for neural network learning, perceptions, multilayer 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.

Evaluation Hypotheses – Motivation, estimation hypothesis accuracy, basics of sampling theory, a general approach for deriving confidence intervals, difference in error of two hypotheses, comparing learning algorithms.

UNIT - III

Bayesian learning – Introduction, Bayes theorem, Bayes theorem and concept learning, Maximum Likelihood and least squared error hypotheses, maximum likelihood hypotheses for predicting probabilities, minimum description length principle, Bayes optimal classifier, Gibs algorithm, Naïve Bayes classifier, an example: learning to classify text, Bayesian belief networks, the EM algorithm.

Computational learning theory – Introduction, probably learning an approximately correct hypothesis, sample complexity for finite hypothesis space, sample complexity for infinite hypothesis spaces, the mistake bound model of learning.

Instance-Based Learning- Introduction, *k*-nearest neighbour algorithm, locally weighted regression, radial basis functions, case-based reasoning, remarks on lazy and eager learning.

UNIT- IV

Genetic Algorithms – Motivation, Genetic algorithms, an illustrative example, hypothesis space search, genetic programming, models of evolution and learning, parallelizing genetic algorithms.

Learning Sets of Rules – Introduction, sequential covering algorithms, learning rule sets: summary, learning First-Order rules, learning sets of First-Order rules: FOIL, Induction as inverted deduction, inverting resolution.

Reinforcement Learning – Introduction, the learning task, *Q*–learning, non-deterministic, rewards and actions, temporal difference learning, generalizing from examples, relationship to dynamic programming.

UNIT - V

Analytical Learning-1- Introduction, learning with perfect domain theories: PROLOG-EBG, remarks on explanation-based learning, explanation-based learning of search control knowledge.

Analytical Learning-2-Using prior knowledge to alter the search objective, using prior knowledge to augment search operators.

Combining Inductive and Analytical Learning – Motivation, inductive-analytical approaches to learning, using prior knowledge to initialize the hypothesis.

TEXT BOOKS:
1. Machine Learning – Tom M. Mitchell, - MGH

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

CS602PC: COMPILER DESIGN

III Year B.Tech. CSE II-Sem

L	T	P	C
3	1	0	4

Prerequisites
1. A course on "Formal Languages and Automata Theory"
2. A course on "Computer Organization and architecture"
3. A course on "Computer Programming and Data Structures"

Course Objectives:
1. Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
2. Topics include phases of compiler, parsing, syntax directd translation, type checking use of symbol tables, code optimization techniques, intermediate code generation, code generation and data flow analysis.

Course Outcomes:
1. Demonstrate the ability to design a compiler given a set of language features.
2. Demonstrate the the knowledge of patterns, tokens & regular expressions for lexical analysis.
3. Acquire skills in using lex tool & yacc tool for devleoping a scanner and parser.
4. Design and implement LL and LR parsers
5. Design algorithms to do code optimization in order to improve the performance of a program in terms of space and time complexity.
6. Design algorithms to generate machine code.

UNIT - I
Introduction: The structure of a compiler, the science of building a compiler, programming language basics
Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex, Finite Automata, From Regular Expressions to Automata, Design of a Lexical-Analyzer Generator, Optimization of DFA-Based Pattern Matchers.

UNIT - II
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, Using Ambiguous Grammars and Parser Generators.

UNIT - III
Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax-Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's.
Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code, Types and Declarations, Type Checking, Control Flow, Switch-Statements, Intermediate Code for Procedures.

UNIT - IV
Run-Time Environments: Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management, Introduction to Garbage Collection, Introduction to Trace-Based Collection.
Code Generation: Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, A Simple Code Generator, Peephole Optimization, Register Allocation and Assignment, Dynamic Programming Code-Generation.

UNIT - V
Machine-Independent Optimization: The Principal Sources of Optimization, Introduction to Data-Flow Analysis, Foundations of Data-Flow Analysis, Constant Propagation, Partial-Redundancy Elimination, Loops in Flow Graphs.

TEXT BOOK:
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.

CS603PC: DESIGN AND ANALYSIS OF ALGORITHMS

III Year B.Tech. CSE II-Sem
L T P C
3 1 0 4

Prerequisites
1. A course on "Computer Programming and Data Structures"
2. A course on "Advanced Data Structures"

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

Course Outcomes
1. Ability to analyze the performance of algorithms
2. Ability to choose appropriate data structures and algorithm design methods for a specified application
3. Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs

UNIT - I
Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.
 Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

UNIT - II
Disjoint Sets: Disjoint set operations, union and find algorithms
Backtracking: General method, applications, n-queen's problem, sum of subsets problem, graph coloring

UNIT - III
Dynamic Programming: General method, applications- Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Traveling sales person problem, Reliability design.

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

UNIT - V
Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.
NP-Hard and NP-Complete problems: Basic concepts, non deterministic algorithms, NP - Hard and NP-Complete classes, Cook's theorem.

TEXT BOOKS:
1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharan, University Press.

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

CS611PE: CONCURRENT PROGRAMMING (Professional Elective - III)

III Year B.Tech. CSE II-Sem

L	T	P	C
3	0	0	3

Prerequisites
1. A course on "Operating Systems"
2. A course on "Java Programming"

Course Objectives: To explore the abstractions used in concurrent programming
Course Outcomes:
1. Ability to implement the mechanisms for communication and co-ordination among concurrent processes.
2. Ability to understand and reason about concurrency and concurrent objects
3. Ability to implement the locking and non-blocking mechanisms
4. Ability to understand concurrent objects

UNIT - I
Introduction - Shared Objects and Synchronization, A Fable, Properties of Mutual Exclusion, The Moral, The Producer–Consumer Problem, The Harsh Realities of Parallelization.
Mutual Exclusion - Time, Critical Sections, 2-Thread Solutions, The Peterson Lock, The Filter Lock, Lamport's Bakery Algorithm.

UNIT - II
Concurrent Objects - Concurrency and Correctness, Sequential Objects, Quiescent consistency, Sequential Consistency, Linearizability, Linearization Points, Formal Definitions
Linearizability, Compositional Linearizability, The Nonblocking Property, Progress conditions, Dependent Progress Conditions, The Java Memory Model, Locks and synchronized Blocks, Volatile Fields, Final Fields.

UNIT - III
Synchronization Operations, Consensus Numbers, Consensus Protocols, The compareAndSet() Operation, Introduction Universality, A Lock-Free Universal, Construction Wait-Free Universal Construction, Spin Locks , Test-And-Set Locks

UNIT - IV
Linked Lists: The Role of Locking, Introduction, List-Based Sets, Concurrent Reasoning, Coarse-Grained Synchronization, Fine-Grained Synchronization, Optimistic Synchronization, Lazy Synchronization, Non-Blocking Synchronization

UNIT - V
Concurrent Queues and the ABA Problem, Concurrent Stacks and Elimination, Transactional Memories

TEXT BOOKS:
1. The Art of Multiprocessor Programming, by Maurice Herlihy and Nir Shavit, Morgan Kaufmman Publishers, 1st Edition, Indian Reprint 2012.

REFERENCE BOOKS:
1. Java Concurrency in Practice by Brian Goetz, Tim Peierls, Joshua Block, Joseph Bowbeer, David Holmes and Doug Lea, Addison Wesley, 1st Edition, 2006.
2. Concurrent Programming in Java™: Design Principles and Patterns, Second Edition by Doug Lea, Publisher: Addison Wesley, Pub Date: October 01, 1999.

CS612PE: NETWORK PROGRAMMING (Professional Elective - III)

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

Course Objectives:
1. To understand inter process and inter-system communication
2. To understand socket programming in its entirety
3. To understand usage of TCP/UDP / Raw sockets
4. To understand how to build network applications

Course Outcomes:
1. To write socket API based programs
2. To design and implement client-server applications using TCP and UDP sockets
3. To analyze network programs

UNIT - I
Introduction to Network Programming: OSI model, Unix standards, TCP and UDP & TCP connection establishment and Format, Buffer sizes and limitation, standard internet services, Protocol usage by common internet application.
Sockets : Address structures, value – result arguments, Byte ordering and manipulation function and related functions Elementary TCP sockets – Socket, connect, bind, listen, accept, fork and exec function, concurrent servers. Close function and related function.

UNIT - II
TCP client server : Introduction, TCP Echo server functions, Normal startup, terminate and signal handling server process termination, Crashing and Rebooting of server host shutdown of server host.
Elementary UDP sockets: Introduction UDP Echo server function, lost datagram, summary of UDP example, Lack of flow control with UDP, determining outgoing interface with UDP.
I/O Multiplexing: I/O Models, select function, Batch input, shutdown function, poll function, TCP Echo server,

UNIT - III
Socket options: getsockopt and setsockopt functions. Socket states, Generic socket option IPV6 socket option ICMPV6 socket option IPV6 socket option and TCP socket options.
Advanced I/O Functions-Introduction, Socket Timeouts, recv and send Functions,readv and writev Functions, recvmsg and sendmsg Functions, Ancillary Data, How Much Data Is Queued?, Sockets and Standard I/O, T/TCP: TCP for Transactions.

UNIT - IV
Elementary name and Address conversions: DNS, gethost by Name function, Resolver option, Function and IPV6 support, uname function, other networking information.
Daemon Processes and inetd Superserver – Introduction, syslogd Daemon, syslog Function, daemon_init Function, inetd Daemon, daemon_inetd Function
Broadcasting- Introduction, Broadcast Addresses, Unicast versus Broadcast, dg_cli Function Using Broadcasting, Race Conditions
Multicasting- Introduction, Multicast Addresses, Multicasting versus Broadcasting on A LAN, Multicasting on a WAN, Multicast Socket Options, mcast_join and Related Functions, dg_cli Function Using Multicasting, Receiving MBone Session Announcements, Sending and Receiving, SNTP: Simple Network Time Protocol, SNTP (Continued)

UNIT - V

Raw Sockets-Introduction, Raw Socket Creation, Raw Socket Output, Raw Socket Input, Ping Program, Traceroute Program, An ICMP Message Daemon,

Datalink Access- Introduction, BPF: BSD Packet Filter, DLPI: Data Link Provider Interface, Linux: **SOCK_PACKET, libpcap**: Packet Capture Library, Examining the UDP Checksum Field.

Remote Login: Terminal line disciplines, Pseudo-Terminals, Terminal modes, Control Terminals, rlogin Overview, RPC Transparency Issues.

TEXT BOOKS:

1. UNIX Network Programming, by W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Pearson Education
2. UNIX Network Programming, 1st Edition, - W. Richard Stevens. PHI.

REFERENCES:

1. UNIX Systems Programming using C++ T CHAN, PHI.
2. UNIX for Programmers and Users, 3rd Edition Graham GLASS, King abls, Pearson Education
3. Advanced UNIX Programming 2nd Edition M. J. ROCHKIND, Pearson Education

CS613PE: SCRIPTING LANGUAGES (Professional Elective - III)

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

Prerequisites:
1. A course on "Computer Programming and Data Structures"
2. A course on "Object Oriented Programming Concepts"

Course Objectives:
1. This course introduces the script programming paradigm
2. Introduces scripting languages such as Perl, Ruby and TCL.
3. Learning TCL

Course Outcomes:
1. Comprehend the differences between typical scripting languages and typical system and application programming languages.
2. Gain knowledge of the strengths and weakness of Perl, TCL and Ruby; and select an appropriate language for solving a given problem.
3. Acquire programming skills in scripting language

UNIT - I
Introduction: Ruby, Rails, The structure and Excution of Ruby Programs, Package Management with RUBYGEMS, Ruby and web: Writing CGI scripts, cookies, Choice of Webservers, SOAP and webservices
RubyTk – Simple Tk Application, widgets, Binding events, Canvas, scrolling

UNIT - II
Extending Ruby: Ruby Objects in C, the Jukebox extension, Memory allocation, Ruby Type System, Embedding Ruby to Other Languages, Embedding a Ruby Interperter

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

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

UNIT - V
TCL
TCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, input/output, procedures, strings, patterns, files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, trapping errors, event driven programs, making applications internet aware, Nuts and Bolts Internet Programming, Security Issues, C Interface.
Tk
Tk-Visual Tool Kits, Fundamental Concepts of Tk, Tk by example, Events and Binding, Perl-Tk.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CS614PE: MOBILE APPLICATION DEVELOPMENT (Professional Elective - III)

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

Prerequisites
1. Acquaintance with JAVA programming
2. A Course on DBMS

Course Objectives
1. To demonstrate their understanding of the fundamentals of Android operating systems
2. To improves their skills of using Android software development tools
3. To demonstrate their ability to develop software with reasonable complexity on mobile platform
4. To demonstrate their ability to deploy software to mobile devices
5. To demonstrate their ability to debug programs running on mobile devices

Course Outcomes
1. Student understands the working of Android OS Practically.
2. Student will be able to develop Android user interfaces
3. Student will be able to develop, deploy and maintain the Android Applications.

UNIT - I
Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Android Studio, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools
Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes
Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes

UNIT - II
Android User Interface: Measurements – Device and pixel density independent measuring UNIT - s
Layouts – Linear, Relative, Grid and Table Layouts
User Interface (UI) Components – Editable and non-editable TextViews, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers
Event Handling – Handling clicks or changes of various UI components
Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

UNIT - III
Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS
Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity
Notifications – Creating and Displaying notifications, Displaying Toasts

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

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

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

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

CS615PE: SOFTWARE TESTING METHODOLOGIES (Professional Elective - III)

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

Prerequisites
1. A course on "Software Engineering"

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

Course Outcomes: Design and develop the best test strategies in accordance to the development model.

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

UNIT - II
Transaction Flow Testing: transaction flows, transaction flow testing techniques. Dataflow testing: Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing. Domain Testing: domains and paths, Nice & ugly domains, domain
testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT - III
Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.
Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

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

UNIT - V
Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Win-runner).

Text Books:
1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

References:
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.

CS604PC: MACHINE LEARNING LAB

III Year B.Tech. CSE II-Sem

L	T	P	C
0	0	3	1.5

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

Course Outcomes: After the completion of the course the student can able to:
1. understand complexity of Machine Learning algorithms and their limitations;
2. understand modern notions in data analysis-oriented computing;
3. be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
4. Be capable of performing experiments in Machine Learning using real-world data.

List of Experiments
1. The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is theprobability that a student is absent given that today is Friday? Apply Baye's rule in python to get the result. (Ans: 15%)

2. Extract the data from database using python

3. Implement k-nearest neighbours classification using python

4. Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids)

VAR1	VAR2	CLASS
1.713	1.586	0
0.180	1.786	1
0.353	1.240	1
0.940	1.566	0
1.486	0.759	1
1.266	1.106	0
1.540	0.419	1
0.459	1.799	1
0.773	0.186	1

5. The following training examples map descriptions of individuals onto high, medium and low credit-worthiness.
   medium skiing design   single twenties no -> highRisk
   high   golf   trading   married forties yes -> lowRisk
   low   speedway transport married thirties yes -> medRisk
   medium football banking   single thirties yes -> lowRisk
   high   flying media   married fifties yes -> highRisk
   low   football security   single twenties no -> medRisk
   medium golf   media   single thirties yes -> medRisk
   medium golf   transport married forties yes -> lowRisk
   high   skiing banking   single thirties yes -> highRisk
   low   golf   unemployed married forties yes -> highRisk

Input attributes are (from left to right) income, recreation, job, status, age-group, home-owner. Find the unconditional probability of `golf' and the conditional probability of `single' given `medRisk' in the dataset?

6. Implement linear regression using python.
7. Implement Naïve Bayes theorem to classify the English text
8. Implement an algorithm to demonstrate the significance of genetic algorithm
9. Implement the finite words classification system using Back-propagation algorithm

CS605PC: COMPILER DESIGN LAB

III Year B.Tech. CSE II-Sem
                L T P C
                0 0 3 1.5

Prerequisites
1. A Course on "Objected Oriented Programming through Java"

Co-requisites
1. A course on "Web Technologies"

Course Objectives
1. To provide hands-on experience on web technologies
2. To develop client-server application using web technologies
3. To introduce server-side programming with Java servlets and JSP
4. To understand the various phases in the design of a compiler.
5. To understand the design of top-down and bottom-up parsers.
6. To understand syntax directed translation schemes.
7. To introduce lex and yacc tools.

Course Outcomes
1. Design and develop interactive and dynamic web applications using HTML, CSS, JavaScript and XML
2. Apply client-server principles to develop scalable and enterprise web applications.
3. Ability to design, develop, and implement a compiler for any language.
4. Able to use lex and yacc tools for developing a scanner and a parser.
5. Able to design and implement LL and LR parsers.

List of Experiments
Compiler Design Experiments
1. Write a LEX Program to scan reserved word & Identifiers of C Language
2. Implement Predictive Parsing algorithm
3. Write a C program to generate three address code.
4. Implement SLR(1) Parsing algorithm
5. Design LALR bottom up parser for the given language

```
<program> ::= <block>
<block> ::= { <variabledefinition> <slist> }
      | { <slist> }
<variabledefinition> ::= int <vardeflist> ;
<vardeflist> ::= <vardec> | <vardec> , <vardeflist>
<vardec> ::= <identifier> | <identifier> [ <constant> ]
<slist> ::= <statement> | <statement> ; <slist>
<statement> ::= <assignment> | <ifstatement> | <whilestatement>
        | <block> | <printstatement> | <empty>
<assignment> ::= <identifier> = <expression>
        | <identifier> [ <expression> ] = <expression>
<ifstatement> ::= if <bexpression> then <slist> else <slist> endif
        | if <bexpression> then <slist> endif
<whilestatement> ::= while <bexpression> do <slist> enddo
<printstatement> ::= print ( <expression> )
<expression> ::= <expression> <addingop> <term> | <term> | <addingop> <term>
<bexpression> ::= <expression> <relop> <expression>
<relop> ::= < | <= | == | >= | > | !=
```

<addingop> ::= + | -

<term> ::= <term> <multop> <factor> | <factor>

<multop> ::= * | /

<factor> ::= <constant> | <identifier> | <identifier> [<expression>]
 | (<expression>)

<constant> ::= <digit> | <digit> <constant>

<identifier> ::= <identifier> <letterordigit> | <letter>

<letterordigit> ::= <letter> | <digit>

<letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<empty> has the obvious meaning

Comments (zero or more characters enclosed between the standard C/Java-style comment brackets /*...*/) can be inserted.   The language has rudimentary support for 1-dimensional arrays. The declaration int a[3] declares an array of three elements, referenced as a[0], a[1] and a[2]. Note also that you should worry about the scoping of names.

A simple program written in this language is:

```
{ int a[3],t1,t2;
  t1=2;
  a[0]=1; a[1]=2; a[t1]=3;
  t2=-(a[2]+t1*6)/(a[2]-t1);
  if t2>5 then
    print(t2);
  else {
    int t3;
    t3=99;
    t2=-25;
    print(-t1+t2*t3);   /* this is a comment
                    on 2 lines */
  }
  endif
}
```

CS621PE: CONCURRENT PROGRAMMING LAB (Professional Elective - III)

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

List of Experiments:
1. Design and implement Two-thread mutual exclusion algorithm (Peterson's Algorithm) using multithreaded programming.
2. Design and implement Filter Lock algorithm and check for deadlock-free and starvation-free conditions using multithreaded programming.
3. Design and implement Lamport's Bakery Algorithm and check for deadlock-free and starvation-free conditions using multithreaded programming.
4. Design and implement Lock-based concurrent FIFO queue data structure using multithreaded programming.
5. Design a consensus object using read–write registers by implementing a deadlock-free or starvation-free mutual exclusion lock. (Use CompareAndSet( ) Primitive).
6. Design and implement concurrent List queue data structure using multithreaded programming. (Use Atomic Primitives)
7. Design and implement concurrent Stack queue data structure using multithreaded programming. (Use Atomic Primitives)
8. Design and implement concurrent FIFO queue data structure using multithreaded programming. (Use Atomic Primitives)

CS622PE: NETWORK PROGRAMMING LAB (Professional Elective - III)

III Year B.Tech. CSE II-Sem

L	T	P	C
0	0	2	1

Course Objectives:
1. To understand inter process and inter-system communication
2. To understand socket programming in its entirety
3. To understand usage of TCP/UDP / Raw sockets
4. To understand how to build network applications

Course Outcomes:
1. To write socket API based programs
2. To design and implement client-server applications using TCP and UDP sockets
3. To analyze network programs

List of Experiments
1. Implement programs for Inter Process Communication using PIPE, Message Queue and Shared Memory.
2. Write a programme to create an integer variable using shared memory concept and increment the variable simultaneously by two processes. Use semaphores to avoid race conditions.
3. Design TCP iterative Client and server application to reverse the given input sentence
4. Design TCP iterative Client and server application to reverse the given input sentence
5. Design TCP client and server application to transfer file
6. Design a TCP concurrent server to convert a given text into upper case using multiplexing system call "select"
7. Design a TCP concurrent server to echo given set of sentences using poll functions
8. Design UDP Client and server application to reverse the given input sentence
9. Design UDP Client server to transfer a file
10. Design using poll client server application to multiplex TCP and UDP requests for converting a given text into upper case.
11. Design a RPC application to add and subtract a given pair of integers

TEXT BOOKS:
1. UNIX Network Programming, by W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Pearson Education.
2. UNIX Network Programming, 1st Edition, - W. Richard Stevens. PHI.

CS623PE: SCRIPTING LANGUAGES LAB (Professional Elective - III)

III Year B.Tech. CSE II-Sem

L	T	P	C
0	0	2	1

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

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

Course Outcomes:
- Ability to understand the differences between Scripting languages and programming languages
- Able to gain some fluency programming in Ruby, Perl, TCL

List of Experiments
1. Write a Ruby script to create a new string which is n copies of a given string where n is a non-negative integer
2. Write a Ruby script which accept the radius of a circle from the user and compute the parameter and area.
3. Write a Ruby script which accept the user's first and last name and print them in reverse order with a space between them
4. Write a Ruby script to accept a filename from the user print the extension of that
5. Write a Ruby script to find the greatest of three numbers
6. Write a Ruby script to print odd numbers from 10 to 1
7. Write a Ruby scirpt to check two integers and return true if one of them is 20 otherwise return their sum
8. Write a Ruby script to check two temperatures and return true if one is less than 0 and the other is greater than 100
9. Write a Ruby script to print the elements of a given array
10. Write a Ruby program to retrieve the total marks where subject name and marks of a student stored in a hash
11. Write a TCL script to find the factorial of a number
12. Write a TCL script that multiplies the numbers from 1 to 10
13. Write a TCL script for Sorting a list using a comparison function
14. Write a TCL script to (i)create a list (ii)append elements to the list (iii)Traverse the list (iv)Concatenate the list
15. Write a TCL script to comparing the file modified times.
16. Write a TCL script to Copy a file and translate to native format.
17. a) Write a Perl script to find the largest number among three numbers.
    b) Write a Perl script to print the multiplication tables from 1-10 using subroutines.
18. Write a Perl program to implement the following list of manipulating functions
    a)Shift
    b)Unshift
    c)Push
19. a) Write a Perl script to substitute a word, with another word in a string.
    b) Write a Perl script to validate IP address and email address.
20. Write a Perl script to print the file in reverse order using command line arguments

CS624PE: MOBILE APPLICATION DEVELOPMENT LAB (Professional Elective - III)

III Year B.Tech. CSE II-Sem

L	T	P	C
0	0	2	1

Prerequisites: --- NIL---

Course Objectives:
1. To learn how to develop Applications in android environment.
2. To learn how to develop user interface applications.
3. To learn how to develop URL related applications.

Course Outcomes:
1. Student understands the working of Android OS Practically.
2. Student will be able to develop user interfaces.
3. Student will be able to develop, deploy and maintain the Android Applications.

List of Experiments
1. Create an Android application that shows Hello + name of the user and run it on an emulator. (b) Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.
2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.
3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on right fragment instead of second screen with back button. Use Fragment transactions and Rotation event listener.
4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.
5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.
6. Create an application that uses a text file to store user names and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with Login Failed message.
7. Create a user registration application that stores the user details in a database table.
8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.
9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.
10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.
11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.
12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.
13. Create an application that shows the given URL (from a text field) in a browser.

CS625PE: SOFTWARE TESTING METHODOLOGIES LAB (Professional Elective - III)

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

Prerequisites: A basic knowledge of programming.

Course Objectives
1. To provide knowledge of Software Testing Methods.
2. To develop skills in software test automation and management using latest tools.

Course Outcome
1. Design and develop the best test strategies in accordance to the development model.

List of Experiments:
1. Recording in context sensitive mode and analog mode
2. GUI checkpoint for single property
3. GUI checkpoint for single object/window
4. GUI checkpoint for multiple objects
5. a) Bitmap checkpoint for object/window
   a) Bitmap checkpoint for screen area
6. Database checkpoint for Default check
7. Database checkpoint for custom check
8. Database checkpoint for runtime record check
9. a) Data driven test for dynamic test data submission
   b) Data driven test through flat files
   c) Data driven test through front grids
   d) Data driven test through excel test
10. a) Batch testing without parameter passing
    b) Batch testing with parameter passing
11. Data driven batch
12. Silent mode test execution without any interruption
13. Test case for calculator in windows application

*MC609: ENVIRONMENTAL SCIENCE

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

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

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

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

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

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

UNIT - IV
Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, **Air Pollution:** Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. **Water pollution:** Sources and types of pollution, drinking water quality standards. **Soil Pollution:** Sources and types, Impacts of modern agriculture, degradation of soil. **Noise Pollution:** Sources and Health hazards, standards, **Solid waste:** Municipal Solid Waste management, composition and characteristics of e-Waste and its management. **Pollution control technologies:** Wastewater Treatment methods: Primary, secondary and Tertiary.
Overview of air pollution control technologies, Concepts of bioremediation. **Global Environmental Problems and Global Efforts: C**limate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol.

UNIT - V
Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio-economical aspects. Strategies for risk assessment, Concepts of Environmental Management Plan

(EMP). **Towards Sustainable Future:** Concept of Sustainable Development, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

TEXT BOOKS:
1. Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
2. Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:
1. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHL Learning Private Ltd. New Delhi.
2. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHI Learning Pvt. Ltd.
3. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
4. Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.
5. Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.