

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**  
**B.Tech. in INFORMATION TECHNOLOGY**

**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	IT503PC	Data Communication & Computer Networks	3	1	0	4
4	IT504PC	Web Programming	2	0	0	2
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	IT506PC	Computer Networks & Web Programming 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
		<b>Total Credits</b>	<b>20</b>	<b>1</b>	<b>8</b>	<b>22</b>

**III YEAR II SEMESTER**

S. No.	Course Code	Course Title	L	T	P	Credits
1	IT601PC	Introduction to Embedded Systems	3	0	0	3
2	IT602PC	Principles of Compiler Construction	3	0	0	3
3	IT603PC	Algorithm Design and Analysis	3	0	0	3
4	IT604PC	Internet of Things	3	0	0	3
5		Professional Elective –III	3	0	0	3
6		Open Elective-I	3	0	0	3
7	IT605PC	Embedded Systems & Internet of Things Lab	0	0	3	1.5
8	IT606PC	Compiler Construction Lab	0	0	3	1.5
9		Professional Elective-III Lab	0	0	2	1
10	*MC609	Environmental Science	3	0	0	0
		<b>Total Credits</b>	<b>21</b>	<b>0</b>	<b>8</b>	<b>22</b>

**\*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**

IT511PE	Biometrics
CS512PE	Advanced Computer Architecture
CS513PE	Data Analytics
CS514PE	Image Processing
CS515PE	Principles of Programming Languages

**Professional Elective - II**

IT521PE	Database Security
CS522PE	Advanced Operating Systems
IT523PE	Machine Learning
IT524PE	Pattern Recognition
CS521PE	Computer Graphics

**Professional Elective - III**

IT611PE	Ethical Hacking
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. IT 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  $\epsilon$ -transitions to NFA without  $\epsilon$ -transitions. Conversion of NFA to DFA, Moore and Melay machines

**UNIT - II**

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

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

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

**UNIT - III**

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

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

**UNIT - IV**

**Normal Forms for Context- Free Grammars:** Eliminating useless symbols, Eliminating  $\epsilon$ -Productions. Chomsky Normal form Griebach 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, 3<sup>rd</sup> Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekar, 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, 3<sup>rd</sup> edition, Cengage Learning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

**CS502PC: SOFTWARE ENGINEERING**

III Year B.Tech. IT 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, 6<sup>th</sup> edition, Mc Graw Hill International Edition.
2. Software Engineering- Sommerville, 7<sup>th</sup> edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rumbaugh, 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.

**IT503PC: DATA COMMUNICATION AND COMPUTER NETWORKS****III Year B.Tech. IT I-Sem**

L	T	P	C
3	1	0	4

**Course Objectives:**

- To introduce the fundamental various types of computer networks.
- To demonstrate the TCP/IP and OSI models with merits and demerits.
- To explore the various layers of OSI Model.
- To introduce UDP and TCP Models.

**Course Outcomes:**

- Students should be understand and explore the basics of Computer Networks and Various Protocols. He/She will be in a position to understand the World Wide Web concepts.
- Students will be in a position to administrate a network and flow of information further he/she can understand easily the concepts of network security, Mobile and ad hoc networks.

**UNIT – I**

**Data Communications:** Components – Direction of Data flow – Networks – Components and Categories – Types of Connections – Topologies –Protocols and Standards – ISO / OSI model, Example Networks such as ATM, Frame Relay, ISDN Physical layer: Transmission modes, Multiplexing, Transmission Media, Switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks.

**UNIT – II**

**Data link layer:** Introduction, Framing, and Error – Detection and Correction – Parity – LRC – CRC Hamming code, Flow and Error Control, Noiseless Channels, Noisy Channels, HDLC, Point to Point Protocols. 111 Medium Access sub layer: ALOHA, CSMA/CD, LAN – Ethernet IEEE 802.3, IEEE 802.5 – IEEE 802.11, Random access, Controlled access, Channelization.

**UNIT – III**

**Network layer:** Logical Addressing, Internetworking, Tunneling, Address mapping, ICMP, IGMP, Forwarding, Uni-Cast Routing Protocols, Multicast Routing Protocols.

**UNIT – IV**

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

**UNIT – V**

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

**TEXT BOOKS:**

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

**REFERENCES:**

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

**IT504PC: WEB PROGRAMMING****III Year B.Tech. IT I-Sem**

L	T	P	C
3	0	0	3

**Course Objectives:** The student should be able to:

- Understand the technologies used in Web Programming.
- Know the importance of object-oriented aspects of Scripting.
- Understand creating database connectivity using JDBC.
- Learn the concepts of web-based application using sockets.

**Course Outcomes:** Upon Completion of the course, the students will be able to

- Design web pages.
- Use technologies of Web Programming.
- Apply object-oriented aspects to Scripting.
- Create databases with connectivity using JDBC.
- Build web-based application using sockets.

**UNIT - I****SCRIPTING.**

Web page Designing using HTML, Scripting basics- Client side and server side scripting. Java Script- Object, names, literals, operators and expressions- statements and features- events - windows - documents - frames - data types - built-in functions- Browser object model - Verifying forms.-HTML5- CSS3- HTML 5 canvas - Web site creation using tools.

**UNIT – II****JAVA**

Introduction to object-oriented programming-Features of Java – Data types, variables and arrays – Operators – Control statements – Classes and Methods – Inheritance. Packages and Interfaces – Exception Handling – Multithreaded Programming – Input/Output – Files – Utility Classes – String Handling.

**UNIT – III****JDBC**

JDBC Overview – JDBC implementation – Connection class – Statements - Catching Database Results, handling database Queries. Networking– InetAddress class – URL class- TCP sockets – UDP sockets, Java Beans –RMI.

**UNIT – IV****APPLETS**

Java applets- Life cycle of an applet – Adding images to an applet – Adding sound to an applet. Passing parameters to an applet. Event Handling. Introducing AWT: Working with Windows Graphics and Text. Using AWT Controls, Layout Managers and Menus. Servlet – life cycle of a servlet. The Servlet API, Handling HTTP Request and Response, using Cookies, Session Tracking. Introduction to JSP.

**UNIT – V****XML AND WEB SERVICES**

Xml – Introduction-Form Navigation-XML Documents- XSL – XSLT- Web services-UDDI-WSDL-Java web services – Web resources.

**TEXT BOOKS:**

1. Harvey Deitel, Abbey Deitel, Internet and World Wide Web: How To Program 5th Edition.



2. Herbert Schildt, Java - The Complete Reference, 7th Edition. Tata McGraw- Hill Edition.
3. Michael Morrison XML Unleashed Tech media SAMS.

**REFERENCES:**

1. John Pollock, Javascript - A Beginners Guide, 3rd Edition -- Tata McGraw-Hill Edition.
2. Keyur Shah, Gateway to Java Programmer Sun Certification, Tata McGraw Hill, 2002.

**IT511PE: BIOMETRICS (Professional Elective - I)****III Year B.Tech. IT I-Sem**

L	T	P	C
3	0	0	3

**Prerequisites:**

1. Information security
2. Network Security

**Course Objectives:**

1. Will learn the biometric technologies.
2. Learn the computational methods involved in the biometric systems.
3. Learn methods for evaluation of the reliability and quality of the biometric systems.

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

1. Identify the various Biometric technologies.
2. Design of biometric recognition for the organization.
3. Develop simple applications for privacy.
4. Understand the watermarking techniques of biometrics.
5. Understand the research on biometric techniques.
6. Understand the need of biometric in the society.

**UNIT - I**

**Introduction & Handwritten Character Recognition:** Introduction, history, type of Biometrics, General Architecture of Biometric Systems, Basic Working of biometric Matching, Biometric System Error and performance Measures, Design of Biometric Systems, Applications of Biometrics, Benefits of Biometrics Versus Traditional Authentication Methods, character Recognition, System Overview, Gesture Extraction for character Recognition, Neura; Network for handwritten Character Recognition, Multilayer Neural Network for Handwritten Character Recognition, Devanagari Numeral Recognition, Isolated Handwritten Devanagari Character Recognition using Fourier Descriptor and Hidden markov Model.

**UNIT - II**

**Face Biometrics & Retina And Iris Biometrics** Introduction, Background of Face Recognition, Design of Face Recognition System, Neural Network for Face Recognition, Face Detection in Video Sequences, Challenges in Face Biometrics, Face Recognition Methods, Advantages and Disadvantages, Performance of Biometrics, Design of Retina Biometrics, Iris Segmentation Method, Determination of Iris Region, Experimental Results of Iris Localization, Applications of Iris Biometrics, Advantages and Disadvantages. Vein and Fingerprint Biometrics & Biometric Hand Gesture Recognition For Indian Sign Language. Biometrics Using Vein Pattern of Palm, Fingerprint Biometrics, Fingerprint Recognition System, Minutiae Extraction, Fingerprint Indexing, Experimental Results, Advantages and Disadvantages, Basics of Hand Geometry, Sign Language, Indian Sign Language, SIFT Algorithms- Practical Approach Advantages and Disadvantages.

**UNIT - III**

**Privacy Enhancement Using Biometrics & Biometric Cryptography And Multimodal Biometrics:** Introduction, Privacy Concerns Associated with Biometric Developments, Identity and Privacy, Privacy Concerns, Biometrics with Privacy Enhancement, Comparison of Various Biometrics in Terms of Privacy, Soft Biometrics - Introduction to Biometric Cryptography, General Purpose Cryptosystem, Modern Cryptography and Attacks, Symmetric Key Ciphers, Cryptographic Algorithms, Introduction to Multimodal Biometrics, Basic Architecture of Multimodal Biometrics, Multimodal Biometrics Using Face and Ear, Characteristics and Advantages of Multimodal Biometrics Characters, AADHAAR : An Application of Multimodal Biometrics.

#### **UNIT - IV**

**Watermarking Techniques & Biometrics : Scope And Future** Introduction, Data Hiding Methods, Basic Framework of Watermarking, Classification of Watermarking, Applications of Watermarking, Attacks on Watermarks, Performance Evaluation, Characteristics of Watermarks, General Watermarking Process, Image Watermarking Techniques, Watermarking Algorithm, Experimental Results, Effect of Attacks on Watermarking Techniques, Scope and Future Market of Biometrics, Biometric Technologies, Applications of Biometrics -Biometrics, and Information Technology Infrastructure, Role of Biometrics in Enterprise Security, Role of Biometrics in Border Security, Smart Card Technology and Biometric, Radio Frequency Identification Biometrics, DNA Biometrics, Comparative Study of Various Biometrics Techniques.

#### **UNIT - V**

**Image Enhancement Techniques & Biometrics Stands:** Introduction, current Research in image Enhancement Techniques, Image Enhancement, Frequency Domain Filters, Databases and Implementation, Standard Development Organizations, Application Programming Interface, Information Security and Biometric Standards, Biometric Template Interoperability.

#### **TEXT BOOKS:**

1. G r Sinha and Sandeep B. Patil, Biometrics: concepts and applications, Wiely, 2013.
2. Paul Reid, Biometrics for Network Security, Pearson Education.

#### **REFERENCE BOOKS:**

1. Samir Nanavathi, Micheal Thieme and Raj Nanavathi, Biometrics, Identity verification in a networked world, Wiley, dream Tech.
2. John D. Woodward and Jr. Wiley Dreamtech, Biometrics, The Ultimate Reference.

#### **Online websites / Materials:**

1. <https://www.biometricsinstitute.org>
2. [https://www.tutorialspoint.com/biometrics/biometrics\\_quick\\_guide.htm](https://www.tutorialspoint.com/biometrics/biometrics_quick_guide.htm)

#### **Online Courses:**

1. <http://nptel.ac.in>
2. <https://www.coursera.org>

**CS512PE: ADVANCED COMPUTER ARCHITECTURE (Professional Elective - I)**

III Year B.Tech. IT 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, Multivector and SIMD computers, Vector Processing Principals, Multivector Multiprocessors, Compound Vector processing, SIMD computer Organizations, The connection machine CM-5,

**UNIT - V**

Scalable, Multithreaded and Dataflow Architectures, Latency-hiding techniques, 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. IT 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, 3<sup>rd</sup> Edition, Morgan Kaufmann Publishers.

**REFERENCE BOOKS:**

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

**CS514PE: IMAGE PROCESSING (Professional Elective - I)**

III Year B.Tech. IT 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, 2<sup>nd</sup> 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 Wiley, 3<sup>rd</sup> Edition, 2004.

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

III Year B.Tech. IT 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. Loudon, 2nd Edition, Thomson, 2003

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

III Year B.Tech. IT I-Sem

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

**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, 2<sup>nd</sup> 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. IT 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 Heuristic 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, Niranjana G. Shivaratri, Tata McGraw-Hill Edition 2001

**REFERENCES:**

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

**IT521PE: DATABASE SECURITY (Professional Elective - II)**

III Year B.Tech. IT I-Sem

L	T	P	C
3	0	0	3

**Course Objectives:**

- To learn the security of databases
- To learn the design techniques of database security
- To learn the secure software design

**Course Outcomes:**

- Ability to carry out a risk analysis for large database.
- Ability to set up, and maintain the accounts with privileges and roles.

**UNIT - I****Introduction:** Introduction to Databases Security Problems in Databases Security Controls

Conclusions

**Security Models -1:** Introduction Access Matrix Model Take-Grant Model Acten Model PN Model Hartson and Hsiao's Model Fernandez's Model Bussolati and Martella's Model for Distributed databases**UNIT - II****Security Models -2:** Bell and LaPadula's Model Biba's Model Dion's Model Sea View Model Jajodia and Sandhu's Model The Lattice Model for the Flow Control conclusion**Security Mechanisms:** Introduction User Identification/Authentication Memory Protection Resource Protection Control Flow Mechanisms Isolation Security Functionalities in Some Operating Systems Trusted Computer System Evaluation Criteria**UNIT - III****Security Software Design:** Introduction A Methodological Approach to Security Software Design Secure Operating System Design Secure DBMS Design Security Packages Database Security Design**Statistical Database Protection & Intrusion Detection Systems:** Introduction Statistics Concepts and Definitions Types of Attacks Inference Controls evaluation Criteria for Control Comparison. Introduction IDES System RETISS System ASES System Discovery**UNIT - IV****Models for the Protection of New Generation Database Systems -1:** Introduction A Model for the Protection of Frame Based Systems A Model for the Protection of Object-Oriented Systems SORION Model for the Protection of Object-Oriented Databases**UNIT - V****Models for the Protection of New Generation Database Systems -2:** A Model for the Protection of New Generation Database Systems: the Orion Model ajodia and Kogan's Model A Model for the Protection of Active Databases Conclusions**TEXT BOOKS:**

1. Database Security by Castano Pearson Edition (Iie) Database Security and Auditing: Protecting Data Integrity and Accessibility, 1st Edition, Hassan Afyouni, THOMSON Edition.

**REFERENCE BOOK:**

1. Database security by Alfred basta, melissazgola, CENGAGE learning.



**IT523PE: MACHINE LEARNING (Professional Elective - II)**

III Year B.Tech. IT I-Sem

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

**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, Gibbs 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

**IT524PE: PATTERN RECOGNITION (Professional Elective - II)****III Year B.Tech. IT I-Sem**

L	T	P	C
3	0	0	3

**Prerequisites**

- Students are expected to have knowledge basic linear algebra, basic probability theory and basic programming techniques;
- A course on “Computational Mathematics”
- A course on “Computer Oriented Statistical Methods”

**Course Objectives**

- This course introduces fundamental concepts, theories, and algorithms for pattern recognition and machine learning.
- Topics include: Pattern Representation, Nearest Neighbor Based Classifier, Bayes Classifier, Hidden Markov Models, Decision Trees, Support Vector Machines, Clustering, and an application of hand-written digit recognition.

**Course Outcomes**

- Understand the theory, benefits, inadequacies and possible applications of various machine learning and pattern recognition algorithms
- Identify and employ suitable machine learning techniques in classification, pattern recognition, clustering and decision problems.

**UNIT - I:** Introduction: What is Pattern Recognition, Data Sets for Pattern Recognition, Different Paradigms for Pattern Recognition. Representation: Data Structures for Pattern Representation, Representation of Clusters, Proximity Measures, Size of Patterns, Abstractions of the Data Set, Feature Extraction, Feature Selection, Evaluation of Classifier, Evaluation of Clustering.

**UNIT - II:** Nearest Neighbor Based Classifier: Nearest Neighbor Algorithm, Variants of the NN Algorithm use of the Nearest Neighbor Algorithm for Transaction Databases, Efficient Algorithms, Data Reduction, Prototype Selection. Bayes Classifier: Bayes Theorem, Minimum Error Rate Classifier, Estimation of Probabilities, Comparison with the NNC, Naïve Bayes Classifier, Bayesian Belief Network.

**UNIT - III:** Hidden Markov Models: Markov Models for Classification, Hidden Markov Models, Classification using HMMs. Decision Trees: Introduction, Decision Tree for Pattern Classification, Construction of Decision Trees, Splitting at the Nodes, Overfitting and Pruning, Examples of Decision Tree Induction.

**UNIT - IV:** Support Vector Machines: Introduction, Learning the Linear Discriminant Functions, Neural Networks, SVM for Classification. Combination of Classifiers: Introduction, Methods for Constructing Ensembles of Classifiers, Methods for Combining Classifiers.

**UNIT - V:** Clustering: Why is Clustering Important, Hierarchical Algorithms, Partitional Clustering, Clustering Large Data Sets. An Application-Hand Written Digit Recognition: Description of the Digit Data, Preprocessing of Data, Classification Algorithms, Selection of Representative Patterns, Results.

**TEXT BOOK:**

1. Pattern Recognition: An Algorithmic Approach: Murty, M. Narasimha, Devi, V. Susheela, Springer Pub, 1st Ed.

**REFERENCES:**

1. Machine Learning - Mc Graw Hill, Tom M. Mitchell.
2. Fundamentals Of Speech Recognition: Lawrence Rabiner and Biing- Hwang Juang. Prentice-Hall Pub.

**CS505PC: SOFTWARE ENGINEERING LAB****III Year B.Tech. IT I-Sem**

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

**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, 6<sup>th</sup> edition, Mc Graw Hill International Edition.
2. Software Engineering- Sommerville, 7<sup>th</sup> edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

**IT506PC: COMPUTER NETWORKS & WEB TECHNOLOGIES LAB****III Year B.Tech. IT 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 framing 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 Dijkstra'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](http://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, 8<sup>th</sup> 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. IT I-Sem

L	T	P	C
3	0	0	3

**1. INTRODUCTION:**

The introduction of the Advanced Communication Skills Lab is considered essential at 3<sup>rd</sup> 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, 7<sup>th</sup> 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. 2<sup>nd</sup> Edition
2. Academic Writing: A Handbook for International Students by Stephen Bailey, Routledge, 5<sup>th</sup> 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. IT 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

**IT601PC: INTRODUCTION TO EMBEDDED SYSTEMS**

III Year B.Tech. IT II -Sem

L	T	P	C
3	0	0	3

**Prerequisites**

1. A course on "Digital Logic Design and Microprocessors"
2. A course on "Computer Organization and Architecture"

**Course Objectives**

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

**Course Outcomes**

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

**UNIT - I****Introduction to Embedded Systems:**

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

**UNIT - II****The Typical Embedded System:**

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

**UNIT - III****Embedded Firmware Design and Development:**

Embedded Firmware Design, Embedded Firmware Development Languages, Programming in Embedded C.

**UNIT - IV****RTOS Based Embedded System Design:**

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

**UNIT - V****Integration and Testing of Embedded Hardware and Firmware:**

Integration of Hardware and Firmware, Boards Bring up

**The Embedded System Development Environment:**

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

**TEXT BOOKS:**

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

**REFERENCES:**

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

**IT602PC: PRINCIPLES OF COMPILER CONSTRUCTION**

III Year B.Tech. IT II -Sem

L	T	P	C
3	0	0	3

**Course Objectives:**

- To understand the various phases in the design of a compiler.
- To study various data structures used
- To understand the design of top-down and bottom-up parsers.
- To understand syntax directed translation schemes.
- To introduce lex and yacc tools.
- To learn intermediate languages
- To learn to develop algorithms to generate code for a target machine.
- To learn how to optimize machine code

**Course Outcomes:**

- Ability to design, develop, and implement a compiler for any language.
- Able to use lex and yacc tools for developing a scanner and a parser.
- Able to design and implement LL and LR parsers.
- Able to design algorithms to perform code optimization in order to improve the performance of a program in terms of space and time complexity.
- Ability to design algorithms to generate machine code

**UNIT - I****Introduction:** Phases of compiler, Grouping of phases.**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.**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.**UNIT - III****Syntax-Directed Translation:** Syntax-Directed Definitions, Construction of syntax trees, Bottom-up evaluation of S-attributed definitions, L-attributed definitions, Top down translation, Bottom-up evaluation of inherited attributes.**Type checking:** Type systems, Specification of a simple type checker, Equivalence of type expressions.**Intermediate-Code Generation:** Intermediate languages, Declarations**UNIT - IV****Run-Time Environments:** Storage organization, Storage allocation strategies, Symbol tables.**Code Generation:** Issues in the Design of a Code Generator, The Target Machine, Basic Blocks and Flow Graphs, A Simple Code Generator, Register Allocation and Assignment, Generation of DAGs, Generating code from DAGs.**UNIT - V****Machine-Independent Optimizations:** Introduction, The Principal Sources of Optimization, **Introduction to Data-Flow Analysis**, Foundations of Data-Flow Analysis.**TEXT BOOKS:**

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

**REFERENCE BOOKS:**

1. Compiler Construction-Principles and Practice, Kenneth C Loudon, Cengage Learning.
2. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
3. The Theory and Practice of Compiler writing, J. P. Tremblay and P. G. Sorenson, TMH
4. Writing compilers and interpreters, R. Mak, 3<sup>rd</sup> edition, Wiley student edition.
5. lex & yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly

**IT603PC: ALGORITHM DESIGN AND ANALYSIS**

III Year B.Tech. IT II -Sem

L	T	P	C
3	0	0	3

**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.

**IT604PC: INTERNET OF THINGS****III Year B.Tech. IT II -Sem**

L	T	P	C
3	0	0	3

**Course Objectives**

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

**Course Outcomes**

1. Interpret the impact and challenges posed by IoT networks leading to new architectural models.
2. Compare and contrast the deployment of smart objects and the technologies to connect them to network.
3. Appraise the role of IoT protocols for efficient network communication.
4. Elaborate the need for Data Analytics and Security in IoT.
5. Illustrate different sensor technologies for sensing real world entities and identify the applications of IoT in Industry.

**UNIT - I**

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT – IoT Protocols, IoT communication models, IoT Communication APIs IoT enabled Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems, IoT Levels and Templates Domain Specific IoTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle

**UNIT - II**

IoT and M2M – Software defined networks, network function virtualization, difference between SDN and NFV for IoT Basics of IoT System Management with NETCOZF, YANG- NETCONF, YANG, SNMP NETOPEER

**UNIT - III**

Introduction to Python - Language features of Python, Data types, data structures, Control of flow, functions, modules, packaging, file handling, data/time operations, classes, Exception handling Python packages - JSON, XML, HTTPLib, URLLib, SMTPLib

**UNIT - IV**

IoT Physical Devices and Endpoints - Introduction to Raspberry PI-Interfaces (serial, SPI, I2C) Programming – Python program with Raspberry PI with focus of interfacing external gadgets, controlling output, reading input from pins.

**UNIT - V**

IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs Webserver – Web server for IoT, Cloud for IoT, Python web application framework Designing a RESTful web API



**TEXT BOOK**

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
2. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759

**IT611PE: ETHICAL HACKING (Professional Elective - III)****III Year B.Tech. IT II -Sem**

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

**Prerequisites:**

1. A course on "Operating Systems"
2. A course on "Computer Networks"
3. A course on "Network Security and Cryptography"

**Course Objectives:**

1. The aim of the course is to introduce the methodologies and framework of ethical hacking for enhancing the security.
2. The course includes-Impacts of Hacking; Types of Hackers; Information Security Models; Information Security Program; Business Perspective; Planning a Controlled Attack; Framework of Steps (Reconnaissance, Enumeration, Vulnerability Analysis, Exploitation, Deliverable and Integration)

**Course Outcomes:**

1. Gain the knowledge of the use and availability of tools to support an ethical hack
2. Gain the knowledge of interpreting the results of a controlled attack
3. Understand the role of politics, inherent and imposed limitations and metrics for planning of a test
4. Comprehend the dangers associated with penetration testing

**UNIT - I****Introduction:** Hacking Impacts, The Hacker**Framework:** Planning the test, Sound Operations, Reconnaissance, Enumeration, Vulnerability Analysis, Exploitation, Final Analysis, Deliverable, Integration**Information Security Models:** Computer Security, Network Security, Service Security, Application Security, Security Architecture**Information Security Program:** The Process of Information Security, Component Parts of Information Security Program, Risk Analysis and Ethical Hacking**UNIT - II****The Business Perspective:** Business Objectives, Security Policy, Previous Test Results, Business Challenges**Planning for a Controlled Attack:** Inherent Limitations, Imposed Limitations, timing is Everything, Attack Type, Source Point, Required Knowledge, Multi-Phased Attacks, Teaming and Attack Structure, Engagement Planner, The Right Security Consultant, The Tester, Logistics, Intermediates, Law Enforcement**UNIT - III****Preparing for a Hack:** Technical Preparation, Managing the Engagement**Reconnaissance:** Social Engineering, Physical Security, Internet Reconnaissance**UNIT - IV****Enumeration:** Enumeration Techniques, Soft Objective, Looking Around or Attack, Elements of Enumeration, Preparing for the Next Phase**Exploitation:** Intuitive Testing, Evasion, Threads and Groups, Operating Systems, Password Crackers, RootKits, applications, Wardialing, Network, Services and Areas of Concern

**UNIT - V**

**Deliverable:** The Deliverable, The Document, Overall Structure, Aligning Findings, Presentation

**Integration:** Integrating the Results, Integration Summary, Mitigation, Defense Planning, Incident Management, Security Policy, Conclusion

**TEXT BOOK:**

1. James S. Tiller, "The Ethical Hack: A Framework for Business Value Penetration Testing", Auerbach Publications, CRC Press

**REFERENCE BOOKS:**

1. EC-Council, "Ethical Hacking and Countermeasures Attack Phases", Cengage Learning
2. Michael Simpson, Kent Backman, James Corley, "Hands-On Ethical Hacking and Network Defense", Cengage Learning

**CS612PE: NETWORK PROGRAMMING (Professional Elective - III)**

III Year B.Tech. IT II-Sem

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

**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, 1<sup>st</sup> 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. IT 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 Execution 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 Interpreter

**UNIT - III**

Introduction to PERL and Scripting

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

**UNIT - IV**

Advanced perl

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

**UNIT - V****TCL**

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

**Tk**

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

**TEXT BOOKS:**

1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly
3. "Programming Ruby" The Pragmatic Programmers guide by Dave 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. IT 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 improve 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. IT 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.

**IT605PC: EMBEDDED SYSTEMS & INTERNET OF THINGS LAB**

III Year B.Tech. IT II-Sem

L	T	P	C
0	0	3	1.5

**List of Experiments:****1. Functional Testing Of Devices**

Flashing the OS on to the device into a stable functional state by porting desktop environment with necessary packages.

**2. Exporting Display On To Other Systems**

Making use of available laptop/desktop displays as a display for the device using SSH client & X11 display server.

**3. GPIO Programming**

Programming of available GPIO pins of the corresponding device using native programming language. Interfacing of I/O devices like LED/Switch etc., and testing the functionality.

**4. Interfacing Chronos eZ430**

Chronos device is a programmable texas instruments watch which can be used for multiple purposes like PPT control, Mouse operations etc., Exploit the features of the device by interfacing with devices.

**5. ON/OFF Control Based On Light Intensity**

Using the light sensors, monitor the surrounding light intensity & automatically turn ON/OFF the high intensity LED's by taking some pre-defined threshold light intensity value.

**6. Battery Voltage Range Indicator**

Monitor the voltage level of the battery and indicating the same using multiple LED's (for ex: for 3V battery and 3 led's, turn on 3 led's for 2-3V, 2 led's for 1-2V, 1 led for 0.1-1V & turn off all for 0V)

**7. Dice Game Simulation**

Instead of using the conventional dice, generate a random value similar to dice value and display the same using a 16X2 LCD. A possible extension could be to provide the user with option of selecting single or double dice game.

**8. Displaying RSS News Feed On Display Interface**

Displaying the RSS news feed headlines on a LCD display connected to device. This can be adapted to other websites like twitter or other information websites. Python can be used to acquire data from the internet.

**9. Porting Openwrt To the Device**

Attempt to use the device while connecting to a wifi network using a USB dongle and at the same time providing a wireless access point to the dongle.

**10. Hosting a website on Board**

Building and hosting a simple website(static/dynamic) on the device and make it accessible online. There is a need to install server(eg: Apache) and thereby host the website.

**11. Webcam Server**

Interfacing the regular usb webcam with the device and turn it into fully functional IP webcam & test the functionality.

## 12. FM Transmission

Transforming the device into a regular fm transmitter capable of transmitting audio at desired frequency (generally 88-108 Mhz)

**Note:** Devices mentioned in the above lists include Arduino, Raspbery Pi, Beaglebone

**IT606PC: COMPILER CONSTRUCTION LAB**

III Year B.Tech. IT 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

```

**IT621PE: ETHICAL HACKING LAB (Professional Elective - III)****III Year B.Tech. IT II-Sem**

L	T	P	C
0	0	2	1

**Course Objectives**

1. The aim of the course is to introduce the methodologies framework tools of ethical hacking to get awareness in enhancing the security
2. To get knowledge on various attacks and their detection

**Course Outcomes**

1. Gain the knowledge of the use and availability of tools to support an ethical hack
2. Gain the knowledge of interpreting the results of a controlled attack

**List of Experiments**

1. Setup a honey pot and monitor the honey pot on network
2. Write a script or code to demonstrate SQL injection attacks
3. Create a social networking website login page using phishing techniques
4. Write a code to demonstrate DoS attacks
5. Install rootkits and study variety of options
6. Study of Techniques uses for Web Based Password Capturing.
7. Install jcrypt tool (or any other equivalent) and demonstrate Asymmetric, Symmetric Crypto algorithm, Hash and Digital/PKI signatures studied in theory Network Security And Management
8. Implement Passive scanning, active scanning, session hijacking, cookies extraction using Burp suit tool

**CS622PE: NETWORK PROGRAMMING LAB (Professional Elective - III)**

III Year B.Tech. IT 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, 1<sup>st</sup> Edition, - W. Richard Stevens. PHI.



**CS623PE: SCRIPTING LANGUAGES LAB (Professional Elective - III)**

III Year B.Tech. IT 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 script to check two integers and return true if one of them is 20 otherwise return their sum
8. Write a Ruby script to check two temperatures and return true if one is less than 0 and the other is greater than 100
9. Write a Ruby script to print the elements of a given array
10. Write a Ruby program to retrieve the total marks where subject name and marks of a student stored in a hash
11. Write a TCL script to find the factorial of a number
12. Write a TCL script that multiplies the numbers from 1 to 10
13. Write a TCL script for Sorting a list using a comparison function
14. Write a TCL script to (i)create a list (ii )append elements to the list (iii)Traverse the list (iv)Concatenate the list
15. Write a TCL script to comparing the file modified times.
16. Write a TCL script to Copy a file and translate to native format.
17. a) Write a Perl script to find the largest number among three numbers.  
b) Write a Perl script to print the multiplication tables from 1-10 using subroutines.
18. Write a Perl program to implement the following list of manipulating functions  
a)Shift  
b)Unshift  
c)Push
19. a) Write a Perl script to substitute a word, with another word in a string.  
b) Write a Perl script to validate IP address and email address.
20. Write a Perl script to print the file in reverse order using command line arguments

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

III III Year B.Tech. IT 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. IT 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. IT 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:** Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol.

**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, 4<sup>th</sup> Edition, New age international publishers.
5. Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.