JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B.Tech in COMPUTER ENGG. (SOFTWARE ENGINEERING)
III & IV YEAR COURSE STRUCTURE & TENTATIVE SYLLABUS (R18)

Applicable From 2020-21 Admitted Batch

III YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Design and Analysis of Algorithms	3	0	0	3
2		Computer Networks	3	0	0	3
3		Database Management Systems	3	0	0	3
4		Software Requirements & Estimation	3	0	0	3
5		Professional Elective – I	3	0	0	3
6		Professional Elective – II	3	0	0	3
7		Computer Networks Lab	0	0	3	1.5
8		Database Management Systems Lab	0	0	3	1.5
9		Advanced Communication Skills Lab	0	0	2	1
10		Intellectual Property Rights	3	0	0	0
		Total Credits	**21**	**0**	**8**	**22**

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Automata Theory and Compiler Design	3	1	0	4
2		Software Testing Methodologies	3	1	0	4
3		Software Architecture and Design Patterns	3	1	0	4
4		Professional Elective – III	3	0	0	3
5		Open Elective – I	3	0	0	3
6		Compiler Design Lab	0	0	3	1.5
7		Software Testing Lab	0	0	3	1.5
8		Professional Elective - III Lab	0	0	2	1
9		Environmental Science	3	0	0	0
		Total Credits	**18**	**3**	**8**	**22**

IV YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Agile Software Development	3	0	0	3
2		Machine Learning	2	0	0	2
3		Professional Elective – IV	3	0	0	3
4		Professional Elective – V	3	0	0	3
5		Open Elective – II	3	0	0	3
6		Machine Learning Lab	0	0	2	1
7		Industrial Oriented Mini Project/ Summer Internship	0	0	0	2*
8		Seminar	0	0	2	1
9		Project Stage – I	0	0	6	3
		Total Credits	**14**	**0**	**10**	**21**

IV YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1		Organizational Behaviour	3	0	0	3
2		Professional Elective – VI	3	0	0	3
3		Open Elective – III	3	0	0	3
4		Project Stage – II	0	0	14	7
		Total Credits	**9**	**0**	**14**	**16**

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

Professional Elective-I

	Data Warehousing and Business Intelligence
	Artificial Intelligence
	Web Programming
	Image Processing
	Computer Graphics

Professional Elective - II

	Mining Massive Datasets
	Information Retrieval Systems
	Internet of Things
	DevOps
	Software Design Methodologies

Professional Elective - III

	Object Oriented Analysis & Design
	Introduction to Data Science
	Scripting Languages
	Mobile Application Development
	Cryptography and Network Security

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

Professional Elective - IV

	Quantum Computing
	Data Visualization Techniques
	Natural Language Processing
	Information Storage Management
	Software Project Management

Professional Elective - V

	Privacy Preserving in Data Mining
	Cloud Computing
	Data Stream Mining
	Software Quality Assurance
	Exploratory Data Analysis

Professional Elective – VI

	Software Metrics
	Web security
	Computational Complexity
	Blockchain Technology
	Parallel and Distributed Computing

AUTOMATA THEORY AND COMPILER DESIGN

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

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

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

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

UNIT - II
Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.
Pumping Lemma for Regular Languages: Statement of the pumping lemma, Applications of the Pumping Lemma.
Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Parse Trees, Ambiguity in Grammars and Languages.

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

UNIT - IV
Introduction: The structure of a compiler,
Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical- Analyzer Generator Lex,
Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom- Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers

UNIT - V

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

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

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

TEXT BOOKS:
1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
3. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd Edition, PHI.

REFERENCE BOOKS:
1. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
2. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
3. lex & yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
4. Compiler Construction, Kenneth C. Louden, Thomson. Course Technology.

SOFTWARE TESTING METHODOLOGIES

B.Tech. III Year II Sem.

L	T	P	C
3	1	0	4

Prerequisites: A course on "Software Engineering"

Course Objectives
- To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
- To develop skills in 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.

REFERENCE BOOKS:
1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

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

Prerequisite: A course on "Software Engineering".

Course Objectives:
1. To understand the concept of patterns and the Catalog.
2. To discuss the Presentation tier design patterns and their effect on: sessions, client access, validation and consistency.
3. To understand the variety of implemented bad practices related to the Business and Integration tiers.
4. To highlight the evolution of patterns.

Course Outcomes:
1. Ability to add functionality to designs while minimizing complexity
2. Understand what design patterns really are, and are not
3. Learn specific design patterns.
4. Able to design patterns to keep code quality high without overdesign.

UNIT - I
Envisioning Architecture
The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views.
Creating an Architecture
Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

UNIT - II
Analyzing Architectures
Architecture Evaluation, Architecture design decision making, ATAM, CBAM.

UNIT - III
Moving from one system to many
Software Product Lines, Building systems from off the shelf components, Software architecture in future.

UNIT - IV
Patterns
Pattern Description, Organizing catalogs, role in solving design problems, Selection and usage.
Creational and Structural patterns
Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight, Proxy.

UNIT - V
Behavioral patterns
Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.
Case Studies
A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development

TEXT BOOKS:
1. Software Architecture in Practice, second edition, Len Bass,Pau Clements & Rick Kazman, Pearson Education, 2003.
2. Design Patterns, Erich Gamma, Pearson Education, 1995.

REFERENCE BOOKS:
1. Beyond Software architecture, Luke Hohmann, Addison wesley, 2003.
2. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001
3. Software Design, David Budgen, second edition, Pearson education, 2003
4. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007.
5. Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006
6. J2EE Patterns, Deepak Alur, John Crupi & Dan Malks, Pearson education, 2003.
7. Design Patterns in C#, Steven John metsker, Pearson education, 2004.

OBJECT ORIENTED ANALYSIS AND DESIGN (Professional Elective – III)

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

Course Objectives:
1. The main objective is to become familiar with all phases of OOAD and master the main features of the UML.
2. Ability to analyze and solve challenging problems in various domains.
3. Learn the Object design Principles and understand how to apply them towards implementation.

Course Outcomes:
1. Select the basic elements of modeling such as Things, Relationships and Diagrams depending on the. views of UML Architecture and SDLC.
2. Design Class and Object Diagrams that represent Static Aspects of a Software System.
3. Design and analyze component and deployment diagrams of a software systems
4. Understand various stages and phases of software projects.

UNIT - I
Introduction to UML: The meaning of Object Orientation, Object identity, Encapsulation, information hiding, polymorphism, generosity, Importance of modeling, principles of modeling, object-oriented modeling, conceptual model of the UML, Architecture. Basic Structural Modeling: Classes, Relationships, common Mechanisms and diagrams. Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

UNIT - II
Collaboration Diagrams: Terms, concepts, depicting a message, polymorphism in collaboration, iterated messages, use of self in messages. Sequence Diagrams: Terms, concepts, depicting asynchronous messages with/without priority, call back mechanism, broadcast messages. Basic Behavioral Modeling: Use cases, Use case Diagrams, Activity Diagrams. Advanced Behavioral Modeling: Events and signals, state machines, processes and threads, time and space, state chart diagrams. Interactions, Interaction diagrams.

Unit - III
Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams. The Unified process: use case driven, architecture centric, iterative and incremental. The Four Ps: people, project, product and process. Use case driven process: why use case, capturing use cases, analysis, design and implementation to realize the use cases, testing the use cases. Architecture-centric process: Architecture in brief, why we need architecture, use cases and architecture, the steps to architecture, an architecture description.

UNIT - IV
Iterative incremental process: Iterative incremental in brief, why iterative incremental development? The iterative approach is risk driven, the generic iteration. The Generic Iteration Workflow: Phases are the first division workflow, planning proceeds doing, risks affect project planning, use case prioritization, resource needed to assess the iteration and phases. Inception Phase: early in the inception phase, the archetypal inception iteration workflow, execute the core workflows, requirements to test.

UNIT - V
Elaboration Phase: Elaboration phase in brief, early in the elaboration phase ,the architectural elaboration iteration workflow, execute the core workflow –requirements to test. Construction phase: early in the construction phase, the archetypal construction iteration workflow, execute the core

workflow. Transition phase: early in the transition phase, activities in transition phase. Case Study: Automation of a Library, Simulation of a Company

TEXT BOOKS:
1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.
3. Ivar Jacobson, Grady Booch, James Rumbaugh: The Unified Software Development Process, Pearson Edition.

REFERENCE BOOKS:
1. Meilir Page-Jones: Fundamentals of Object-Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Mark Priestley: Practical Object-Oriented Design with UML, TATA McGraw Hill.

INTRODUCTION TO DATA SCIENCE (Professional Elective – III)

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

Course Objectives:
1. Learn concepts, techniques and tools they need to deal with various facets of data science practice, including data collection and integration
2. Understand the basic types of data and basic statistics
3. Identify the importance of data reduction and data visualization techniques

Course Outcomes: After completion of the course, the student should be able to
1. Understand basic terms what Statistical Inference means.
2. Identify probability distributions commonly used as foundations for statistical modelling. Fit a model to data
3. describe the data using various statistical measures
4. utilize R elements for data handling
5. perform data reduction and apply visualization techniques.

UNIT - I
Introduction: Definition of Data Science- Big Data and Data Science hype – and getting past the hype - Datafication - Current landscape of perspectives - Statistical Inference - Populations and samples - Statistical modeling, probability distributions, fitting a model – Over fitting. **Basics of R:** Introduction, R-Environment Setup, Programming with R, Basic Data Types.

UNIT - II
Data Types & Statistical Description
Types of Data: Attributes and Measurement, What is an Attribute? The Type of an Attribute, The Different Types of Attributes, Describing Attributes by the Number of Values, Asymmetric Attributes, Binary Attribute, Nominal Attributes, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes. Basic Statistical Descriptions of Data: Measuring the Central Tendency: Mean, Median, and Mode, Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Inter-quartile Range, Graphic Displays of Basic Statistical Descriptions of Data.

UNIT - III
Vectors: Creating and Naming Vectors, Vector Arithmetic, Vector sub setting, **Matrices:** Creating and Naming Matrices, Matrix Sub setting, Arrays, Class. **Factors and Data Frames:** Introduction to Factors: Factor Levels, Summarizing a Factor, Ordered Factors, Comparing Ordered Factors, Introduction to Data Frame, subsetting of Data Frames, Extending Data Frames, Sorting Data Frames.
Lists: Introduction, creating a List: Creating a Named List, Accessing List Elements, Manipulating List Elements, Merging Lists, Converting Lists to Vectors

UNIT - IV
Conditionals and Control Flow: Relational Operators, Relational Operators and Vectors, Logical Operators, Logical Operators and Vectors, Conditional Statements. **Iterative Programming in R:** Introduction, While Loop, For Loop, Looping Over List. **Functions in R:** Introduction, writing a Function in R, Nested Functions, Function Scoping, Recursion, Loading an R Package, Mathematical Functions in R.

UNIT - V
Data Reduction: Overview of Data Reduction Strategies, Wavelet Transforms, Principal Components Analysis, Attribute Subset Selection, Regression and Log-Linear Models: Parametric Data Reduction, Histograms, Clustering, Sampling, Data Cube Aggregation. **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. Doing Data Science, Straight Talk from The Frontline. Cathy O'Neil and Rachel Schutt, O'Reilly, 2014
2. Jiawei Han, Micheline Kamber and Jian Pei. Data Mining: Concepts and Techniques, 3rd ed. The Morgan Kaufmann Series in Data Management Systems.
3. K G Srinivas, G M Siddesh, "Statistical programming in R", Oxford Publications.

REFERENCE BOOKS:
1. Introduction to Data Mining, Pang-Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education.
2. Brain S. Everitt, "A Handbook of Statistical Analysis Using R", Second Edition, 4 LLC, 2014.
3. Dalgaard, Peter, "Introductory statistics with R", Springer Science & Business Media, 2008.
4. Paul Teetor, "R Cookbook", O'Reilly, 2011.

SCRIPTING LANGUAGES (Professional Elective – III)

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

Pre-requisites:
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 Interperter.

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

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

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

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

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

MOBILE APPLICATION DEVELOPMENT (Professional Elective – III)

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

Pre-requisites
1. Acquaintance with JAVA programming.
2. A Course on DBMS.

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

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

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

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

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

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

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

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

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

CRYPTOGRAPHY AND NETWORK SECURITY (Professional Elective – III)

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

Course Objectives:
1. Explain the objectives of information security
2. Explain the importance and application of each of confidentiality, integrity, authentication and availability
3. Understand various cryptographic algorithms.
4. Understand the basic categories of threats to computers and networks
5. Describe public-key cryptosystem.
6. Describe the enhancements made to IPv4 by IPSec
7. Understand Intrusions and intrusion detection
8. Discuss the fundamental ideas of public-key cryptography.
9. Generate and distribute a PGP key pair and use the PGP package to send an encrypted e-mail message.
10. Discuss Web security and Firewalls

Course Outcomes:
1. Student will be able to understand basic cryptographic algorithms, message and web authentication and security issues.
2. Ability to identify information system requirements for both of them such as client and server.
3. Ability to understand the current legal issues towards information security.

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

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

UNIT - III
Cryptographic Hash Functions: Message Authentication, Secure Hash Algorithm (SHA-512),
Message authentication codes: Authentication requirements, HMAC, CMAC, Digital signatures, Elgamal Digital Signature Scheme.
Key Management and Distribution: Symmetric Key Distribution Using Symmetric & Asymmetric Encryption, Distribution of Public Keys, Kerberos, X.509 Authentication Service, Public – Key Infrastructure

UNIT - IV
Transport-level Security: Web security considerations, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH)
Wireless Network Security: Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security

UNIT - V

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

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

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

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

COMPILER DESIGN LAB

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

Pre-requisites: A Course on "Object Oriented Programming through Java".

Co-requisites: 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
}
```

SOFTWARE TESTING LAB

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

Pre-requisites: 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 the latest tools.

Course Outcomes:
1. Design and develop the best test strategies in accordance with the development model.
2. Design and develop GUI checkpoints
3. Develop database checkpoints for different checks
4. Perform batch testing with and without parameter passing

List of Experiments
1. Recording in context sensitive mode and analog mode
2. GUI checkpoint for single property
3. GUI checkpoint for single object/window
4. GUI checkpoint for multiple objects
5. a) Bitmap checkpoint for object/window
   b) Bitmap checkpoint for screen area
6. Database checkpoint for Default check
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

TEXT BOOKS:
1. Software Testing techniques, Baris Beizer, 2nd Edition, Dreamtech.
2. Software Testing Tools, Dr. K.V.K.K. Prasad, Dreamtech.

REFERENCE BOOKS:
1. The craft of software testing, Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World, Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing, Meyers, John Wiley.

OBJECT ORIENTED ANALYSIS AND DESIGN LAB (PE – III Lab)

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

Course Objectives:
1. The main objective is to become familiar with all phases of OOAD and master the main features of the UML.
2. Ability to analyze and solve challenging problems in various domains.
3. Learn the Object design Principles and understand how to apply them towards implementation.

Course Outcomes:
1. Select the basic elements of modeling such as Things, Relationships and Diagrams depending on the views of UML Architecture and SDLC.
2. Design Class and Object Diagrams that represent Static Aspects of a Software System.
3. Design and analyze component and deployment diagrams of a software systems.
4. Understand various stages and phases of software project.

List of Experiments:
1. The student should take up the case study of Unified Library Application which is mentioned in the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
2. Draw the following diagrams using UML for an ATM system whose description is given below. UML diagrams to be developed are:
    1. Use Case Diagram.
    2. Class Diagram.
    3. Sequence Diagram.
    4. Collaboration Diagram.
    5. State Diagram
    6. Activity Diagram.
    7. Component Diagram
    8. Deployment Diagram.
    9. Test Design.

Description for an ATM System
The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of Rs. 100, Rs. 500 and Rs. 1000), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)
The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) – both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

The ATM must be able to provide the following services to the customer:
1. A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of Rs. 100 or Rs. 500 or Rs. 1000. Approval must be obtained from the bank before cash is dispensed.

2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.

3. A customer must be able to make a transfer of money between any two accounts linked to the card.

4. A customer must be able to make a balance inquiry of any account linked to the card.

5. A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine. The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.) If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back. If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction. The ATM will provide the customer with a printed receipt for each successful transaction The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

3. Study of any testing tool (e.g. Win runner)
4. Study of any web testing tool (e.g. Selenium)
5. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
6. Study of any test management tool (e.g. Test Director)
7. Study of any open source-testing tool (e.g. Test Link)

TEXT BOOKS:
1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.
3. Ivar Jacobson, Grady Booch, James Rumbaugh: The Unified Software Development Process, Pearson Edition.

REFERENCE BOOKS:
1. Meilir Page-Jones: Fundamentals of Object-Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Mark Priestley: Practical Object-Oriented Design with UML, TATA McGraw Hill.

INTRODUCTION TO DATA SCIENCE LAB (PE – III Lab)

B.Tech. III Year II Sem.

L	T	P	C
0	0	2	1

Course Objectives: The course should enable the students to:
1. Understand the R Programming Language.
2. Exposure on Solving data science problems.
3. Understand The classification and Regression Model

Course Outcomes:
1. Illustrate the use of various data structures.
2. Analyze and manipulate Data using Pandas
3. Creating static, animated, and interactive visualizations using Matplotlib.
4. Understand the implementation procedures for the machine learning algorithms.
5. Apply appropriate data sets to the Machine Learning algorithms
6. Identify and apply Machine Learning algorithms to solve real-world problems

List of Experiments:

1. R AS CALCULATOR APPLICATION
a. Using with and without R objects on console
b. Using mathematical functions on console
c. Write an R script, to create R objects for calculator application and save in a specified location in disk

2. DESCRIPTIVE STATISTICS IN R
 a. Write an R script to find basic descriptive statistics using summary
 b. Write an R script to find subset of dataset by using subset ()

3. READING AND WRITING DIFFERENT TYPES OF DATASETS
 a. Reading different types of data sets (.txt, .csv) from web and disk and writing in file in specific disk location.
 b. Reading Excel data sheet in R.
 c. Reading XML dataset in R.

4. VISUALIZATIONS
 a. Find the data distributions using a box and scatter plot.
 b. Find the outliers using a plot.
 c. Plot the histogram, bar chart and pie chart on sample data

5. CORRELATION AND COVARIANCE
a. Find the correlation matrix.
b. Plot the correlation plot on the dataset and visualize giving an overview of relationships among data on iris data.
c. Analysis of covariance: variance (ANOVA), if data have categorical variables on iris data.

6. REGRESSION MODEL
Import data from web storage. Name the dataset and now do Logistic Regression to find out relation between variables that are affecting the admission of a student in an institute based on his or her GRE score, GPA obtained and rank of the student. Also check if the model is fit or not. require (foreign), require(MASS).

7.   MULTIPLE REGRESSION MODEL
Apply multiple regressions, if data have a continuous independent variable. Apply on the above dataset.

8.   REGRESSION MODEL FOR PREDICTION
Apply regression Model techniques to predict the data on above dataset

9.   CLASSIFICATION MODEL
a. Install relevant packages for classification.
b. Choose a classifier for classification problems.
c. Evaluate the performance of the classifier.

10.   CLUSTERING MODEL
a. Clustering algorithms for unsupervised classification.
b. Plot the cluster data using R visualizations.

TEXT BOOKS:
1. Doing Data Science, Straight Talk from The Frontline. Cathy O'Neil and Rachel Schutt, O'Reilly, 2014.
2. Jiawei Han, Micheline Kamber and Jian Pei. Data Mining: Concepts and Techniques, 3rd ed. The Morgan Kaufmann Series in Data Management Systems.
3. K G Srinivas, G M Siddesh, "Statistical programming in R", Oxford Publications.

REFERENCE BOOK:
1. Yanchang Zhao, "R and Data Mining: Examples and Case Studies", Elsevier, 1st Edition, 2012.

SCRIPTING LANGUAGES LAB (PE – III Lab)

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

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

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

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

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

MOBILE APPLICATION DEVELOPMENT LAB (PE – III Lab)

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

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.

CRYPTOGRAPHY AND NETWORK SECURITY LAB (PE – III Lab)

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

Course Objectives:
1. Explain the objectives of information security.
2. Explain the importance and application of each of confidentiality, integrity, authentication and availability.
3. Understand various cryptographic algorithms.

Course Outcomes:
1. Understand basic cryptographic algorithms, message and web authentication and security issues.
2. Identify information system requirements for both of them such as client and server.
3. Understand the current legal issues towards information security.

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

TEXT BOOKS:
1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition.
2. Cryptography and Network Security: Atul Kahate, McGraw Hill, 3rd Edition.

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

ENVIRONMENTAL SCIENCE

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

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

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

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

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

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

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

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

explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

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

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