ACE
Engineering College
An Autonomous Institution
(NBA ACCREDITED B.TECH COURSES, ACCORDED NAAC 'A'GRADE)
Ghatkesar, Hyderabad, 501 301

R22 - B.TECH. FOUR YEAR DEGREE COURSE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE STRUCTURE - SYLLABUS

III Year II Semester - Regulation R22

S.No.	Course Code	Course Title	Periods Per Week			Credits
			L	T	P	
1	CS601PC	Machine Learning	3	0	0	3
2	CS602PC	Formal Languages and Automata Theory	3	0	0	3
3	CS603PC	Artificial Intelligence	3	0	0	3
		Professional Elective - III				
4	CS631PE	Full Stack Development	3	0	0	3
	CS632PE	Internet of Things				
	CS633PE	Scripting Languages				
	CS634PE	Mobile Application Development				
	CS635PE	Software Testing Methodologies				
5		**Open Elective-I**	3	0	0	3
6	CS604PC	Machine Learning Lab	0	0	2	1
7	CS605PC	Artificial Intelligence Laboratory	0	0	2	1
		Professional Elective - III Lab				
8	CS611PE	Full Stack Development Lab	0	0	2	1
	CS612PE	Internet of Things Lab				
	CS613PE	Scripting Languages Lab				
	CS614PE	Mobile Application Development Lab				
	CS615PE	Software Testing Methodologies Lab				
9	CS606PC	Industrial Oriented Mini Project/ Internship/ Skill Development Course (Big data-Spark)	0	0	2	2
10	*MC609	Environmental Science	3	0	0	0
		Total Credits	**18**	**0**	**8**	**2**

Note: *MC = Satisfactory/Unsatisfactory

Environmental Science in III Yr II Sem Should be Registered by Lateral Entry Students Only.

Open Elective -1:

1. CS611OE: Data Structures
2. CS612OE: Database Management Systems

CS601PC: MACHINE LEARNING

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

Course Objectives:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To have a thorough understanding of the Supervised and Unsupervised learning techniques.
- To study the various probability, based learning techniques.

Course Outcomes:

- Distinguish between, supervised, unsupervised and semi, supervised learning.
- Design and Explain learning systems to address specific learning tasks effectively.
- Explain the motivation behind genetic algorithms, reinforcement learning tasks and Q, learning algorithm.
- Understand algorithms for building classifiers applied on datasets of non, linearly separable Classes.
- Understand the principles of evolutionary computing algorithms.
- Design an ensembler to increase the classification accuracy.

UNIT - I

Learning: Types of Machine Learning, Supervised Learning , The Brain and the Neuron , Design a Learning System , Perspectives and Issues in Machine Learning , Concept Learning Task , Concept Learning as Search , Finding a Maximally Specific Hypothesis , Version Spaces and the Candidate Elimination Algorithm , Linear Discriminants , Perceptron , Linear Separability , Linear Regression.

UNIT - II

Multi-layer Perceptron: Going Forwards , Going Backwards: Back Propagation Error , Multi-layer Perceptron in Practice , Examples of using the MLP , Overview , Deriving Back-Propagation , Radial Basis Functions and Splines , Concepts , RBF Network , Curse of Dimensionality , Interpolations and Basis Functions , Support Vector Machines.

UNIT - III

Learning with Trees: Decision Trees , Constructing Decision Trees , Classification and Regression Trees , Ensemble Learning , Boosting , Bagging , Different ways to Combine Classifiers , Basic Statistics , Gaussian Mixture Models, Nearest Neighbor Methods , Unsupervised Learning , K means Algorithms.

UNIT - IV

Dimensionality Reduction: Linear Discriminant Analysis , Principal Component Analysis , Factor Analysis, Independent Component Analysis , Locally Linear Embedding , Isomap , Least Squares Optimization.

Evolutionary Learning: Genetic algorithms , Genetic Offspring: , Genetic Operators , Using Genetic Algorithms.

UNIT - V

Reinforcement Learning: Overview , Getting Lost Example.

Markov Chain Monte Carlo Methods: Sampling , Proposal Distribution , Markov Chain Monte Carlo , Graphical Models , Bayesian Networks , Markov Random Fields , Hidden Markov Models , Tracking Methods.

TEXT BOOKS:

1. Stephen Marsland, Machine Learning , An Algorithmic Perspective, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.

REFERENCE BOOKS:

1. Tom M Mitchell, Machine Learning, First Edition, McGraw Hill Education, 2013.
2. Peter Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data‖, First Edition, Cambridge University Press, 2012.
3. Jason Bell, Machine learning , Hands on for Developers and Technical Professionals‖, First Edition, Wiley, 2014.
4. Ethem Alpaydin, Introduction to Machine Learning 3e (Adaptive Computation and Machine Learning Series), Third Edition, MIT Press, 2014.

CS602PC: FORMAL LANGUAGES AND AUTOMATA THEORY

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

Course Objectives:

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

Course Outcomes:

- Understand the concept of abstract machines and their power to recognize the languages.
- Employ finite state machines for modeling and solving computing problems.
- Design context free grammars for formal languages.
- Distinguish between decidability and undecidability.
- Design different types of grammars for formal languages.
- Develop Mathematical tools and formal methods to solve computational problems.

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.Є

Deterministic Finite Automata: Definition of DFA, How a DFA Process Strings, The language of DFA, Conversion of NFA with Є -transitions to NFA without Є -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 ε -Productions. Chomsky Normal form Greibach Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications.

Closure Properties of Context-Free Languages: Closure properties of CFL's, Decision Properties of CFL's Turing Machines - Introduction to Turing Machine, Formal Description, Instantaneous description, the language of a Turing machine.

UNIT - V

Types of Turing machine: Turing machines and halting

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CS603PC: ARTIFICIAL INTELLIGENCE

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

Prerequisites:

1. Programming for problem solving, Data Structures.

Course Objectives:

- To learn the distinction between optimal reasoning Vs. human like reasoning
- To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
- To learn different knowledge representation techniques.
- To understand the applications of AI, namely game playing, theorem proving, and machine learning.

Course Outcomes:

- Understand search strategies and intelligent agents.
- Understand different adversarial search techniques.
- Apply propositional logic, predicate logic for knowledge representation.
- Apply AI techniques to solve problems of game playing, and machine learning.
- Analyze various classical planning techniques.
- Differentiate uninformed and informed search strategies.

UNIT - I

Introduction to AI: Intelligent Agents, problem-Solving Agents, Searching for Solutions.

Uninformed Search Strategies: Breadth-first search, Uniform cost search, Depth-first search, Iterative deepening Depth-first search, Bidirectional search.

Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions, Beyond Classical Search: Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces.

UNIT - II

Problem Solving by Search-II and Propositional Logic

Adversarial Search: Games, Optimal Decisions in Games, Alpha–Beta Pruning, Imperfect Real-Time Decisions. **Constraint Satisfaction Problems:** Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems.

Propositional Logic: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic.

UNIT - III

Logic and Knowledge Representation:

First-Order Logic: Representation, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

UNIT - IV

Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.
Classical Planning: Definition of Classical Planning, Algorithms for Planning with State-Space Search, Planning Graphs, other Classical Planning Approaches, analysis of Planning approaches.

UNIT - V

Uncertain knowledge and Learning Uncertainty: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and its Use.
Probabilistic Reasoning: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.

TEXT BOOK:

1. Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH).
2. Artificial Intelligence, 3rd Edn., Patrick Henry Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education.

CS631PE: FULL STACK DEVELOPMENT (Professional Elective – III)

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

Pre-Requisites:

1. Object Oriented Programming
2. Web Technologies

Course Objectives:

- Students will become familiar to implement fast, efficient, interactive and scalable web applications using run time environment provided by the full stack components.

Course Outcomes:

- Understand Full stack components for developing web application.
- Apply packages of Node.js to work with Data, Files, Http Requests and Responses.
- Use MongoDB data base for storing and processing huge data and connects with Node.js application.
- Design faster and effective single page applications using Express and Angular.
- Create interactive user interfaces with react components.
- Design multi page applications using react.

UNIT-I
Introduction to Full Stack Development:

Understanding the Basic Web Development Framework- user, browser, webserver, backend services. Full Stack Components - Node.js, MongoDB, Express, React, Angular. JavaScript Fundamentals. Node.js-Understanding Node.js, Installing Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks.

UNIT-II
Node.js:

Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Accessing the File System from Node.js- Opening, Closing, Writing, Reading Files and Other File System Tasks. Implementing HTTP Services in Node.js- Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects, Implementing HTTP Clients and Servers in Node.js, Implementing HTTPS Servers and Clients. Using Additional Node.js Modules-Using the os Module, Using the util Module, Using the dns Module, Using the crypto Module.

UNIT-III
MongoDB:

Need of NoSQL, Understanding MongoDB, MongoDB Data Types, Planning Your Data Model, Building the MongoDB Environment, Administering User Accounts, Configuring Access Control, Administering Databases, Managing Collections, Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Understanding the Objects Used in the MongoDB Node.js Driver, Accessing and Manipulating Databases, Accessing and Manipulating Collections

UNIT-IV

Express and Angular:

Getting Started with Express, Configuring Routes, Using Request Objects, Using Response Objects. Angular: Importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing Angular Services in Web Applications.

UNIT-V

React:

Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React, Rendering and Lifecycle Methods in React, Working with Forms in React, Integrating Third-Party Libraries, Routing in React.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress, 2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. KirupaChinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

CS632PE: INTERNET OF THINGS (Professional Elective – III)

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

Pre-Requisites: Computer organization, Computer Networks.

Course Objectives:

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

Course Outcomes:

- Interpret the impact and challenges posed by IoT networks leading to new architectural models.
- Compare and contrast the deployment of smart objects and the technologies to connect them to network.
- Appraise the role of IoT protocols for efficient network communication.
- Identify the applications of IoT in Industry.
- Design IoT applications using python script.
- Design IoT devices using Rasberry PI.

UNIT - I

Introduction to Internet of Things : Definition and Characteristics of IoT, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels and Deployment Templates.
Domain Specific IoTs : Home automation, Environment, Agriculture, Health and Lifestyle

UNIT - II

IoT and M2M : M2M, Difference between IoT and M2M, SDN and NFV for IoT.
IoT System Management with NETCOZF-YANG: Need for IoT system Management, Simple Network management protocol, Network operator requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG.

UNIT - III

IoT Systems - Logical design using Python: Introduction to Python, Python Data types & Data structures, Control flow, Functions, Modules, Packaging, File handling, Data/Time operations, Classes, Exception, Python packages of Interest for IoT.

UNIT - IV:

IoT Physical Devices and Endpoints: Raspberry Pi, Linux on Raspberry Pi, Raspberry Pi Interfaces, Programming Raspberry PI with Python, Other IoT devices.
IoT Physical Servers and Cloud Offerings: Introduction to Cloud Storage models and communication APIs, WAMP-AutoBahn for IoT, Xively Cloud for IoT, Python web application framework
–Django, Designing a RESTful web API

UNIT V

Case studies: Home Automation, Environment-weather monitoring-weather reporting- air pollution monitoring, Agriculture.

TEXT BOOK:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547.

REFERENCE BOOK:

1. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759.

CS633PE: SCRIPTING LANGUAGES (Professional Elective – III)

B.Tech. III Year II Sem.

L	T	P	C
3	0	0	3

Prerequisites:

- A course on “Computer Programming and Data Structures”.
- A course on “Object Oriented Programming Concepts”.

Course Objectives:

- This course introduces the script programming paradigm.
- Introduces scripting languages such as Perl, Ruby and TCL.
- Learning TCL.

Course Outcomes:

- Comprehend the differences between typical scripting languages and typical system and application programming languages.
- Gain knowledge of the strengths and weakness of Perl, TCL and Ruby; and select an appropriate language for solving a given problem.
- Acquire programming skills in scripting language.
- Develop server-side scripts using PHP and Python’s CGI facilities.
- Apply the basic concepts of scripting using Perl.
- Develop applications using Tk Script.

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 web services.

Ruby Tk – 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 Pramatic Progammers guide by Dabve Thomas Second edition.

REFERENCE BOOKS:

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

CS634PE: MOBILE APPLICATION DEVELOPMENT (Professional Elective – III)

B.Tech. III Year II Sem.

L	T	P	C
3	0	0	3

Prerequisites:

- Acquaintance with JAVA programming.
- A Course on DBMS.

Course Objectives:

- To demonstrate their understanding of the fundamentals of Android operating systems.
- To improves their skills of using Android software development tools.
- To demonstrate their ability to develop software with reasonable complexity on mobile platform.
- To demonstrate their ability to deploy software to mobile devices.
- To demonstrate their ability to debug programs running on mobile devices.

Course Outcomes:

- Understand the working of Android OS Practically.
- Identify various concepts of mobile programming that make it unique from programming for other platforms.
- Develop Android user interfaces.
- Develop, deploy and maintain the Android Applications.
- Deploy applications to the Android marketplace for distribution.
- Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces.

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 units, 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 deleting data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update).

TEXT BOOK:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012.

REFERENCE BOOKS:

1. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013.
2. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013.

CS635PE: SOFTWARE TESTING METHODOLOGIES (Professional Elective – III)

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

Prerequisites:

1. 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 the latest tools.

Course Outcomes:

- Understand purpose of testing and path testing.
- Understand strategies in data flow testing and domain testing.
- Develop logic-based test strategies.
- Understand graph matrices and its applications.
- Implement test cases using any testing automation tool.
- Develop test cases using J meter or Win-runner.

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.

Unit - III

Domain Testing: Domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

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

Unit - IV

Syntax Testing: Introduction, Backus-Naur notation (BNF) , Operators in BNF and Syntax Testing Drivers ,

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

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.

CS604PC: MACHINE LEARNING LAB

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

Course Objective:

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

Course Outcomes:

- Understand modern notions in predictive data analysis.
- Select data, model selection, model complexity and identify the trends.
- Understand a range of machine learning algorithms along with their strengths and weaknesses.
- Build predictive models from data and analyze their performance.
- Develop applications using classification algorithms.
- Develop applications using supervised learning algorithms.

List of Experiments:

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode Measure of Dispersion: Variance, Standard Deviation
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy
3. Study of Python Libraries for ML application such as Pandas and Matplotlib
4. Write a Python program to implement Simple Linear Regression
5. Implementation of Multiple Linear Regression for House Price Prediction using sklearn
6. Implementation of Decision tree using sklearn and its parameter tuning
7. Implementation of KNN using sklearn
8. Implementation of Logistic Regression using sklearn
9. Implementation of K-Means Clustering
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

TEXT BOOK:

1. Machine Learning – Tom M. Mitchell, - MGH.

REFERENCE BOOK:

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

CS605PC: ARTIFICIAL INTELLIGENCE LAB

B.Tech. III Year II Sem.

L	T	P	C
0	0	2	1

Course Objectives:

- Become familiar with basic principles of AI toward problem solving, knowledge representation, and learning.

Course Outcomes:

- Apply basic principles of AI in solutions that require problem solving, knowledge representation, and learning.
- Implement simple PEAS descriptions for given AI tasks.
- Develop programs to implement simulated annealing and genetic algorithms.
- Demonstrate the ability to solve problems using searching and backtracking.
- Ability to implement simple reasoning systems using either backward or forward inference.
- apply appropriate algorithms for solving given AI problems.

LIST OF EXPERIMENTS:

Write Programs to implement the following using Python.

1. Breadth First Search
2. Depth First Search
3. Tic-Tac-Toe game
4. 8-Puzzle problem
5. Water-Jug problem
6. Travelling Salesman Problem
7. Tower of Hanoi
8. Monkey Banana Problem
9. Alpha-Beta Pruning
10. 8-Queens Problem

TEXT BOOK:

1. Artificial Intelligence a Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henny Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.

CS611PE: FULL STACK DEVELOPMENT LAB (Professional Elective – III)

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

Pre-Requisites:

- Object Oriented Programming
- Web Technologies

Course Objectives:

- Introduce fast, efficient, interactive and scalable web applications using run time environment provided by the full stack components.

Course Outcomes:

- Design flexible and responsive Web applications using Node JS, React, Express and Angular.
- Perform CRUD operations with MongoDB on huge amount of data.
- Develop real time applications using react components.
- Use various full stack modules to handle http requests and responses.

List of Experiments:

1. Create an application to setup node JS environment and display “Hello World”.
2. Create a Node JS application for user login system.
3. Write a Node JS program to perform read, write and other operations on a file.
4. Write a Node JS program to read form data from query string and generate response using NodeJS.
5. Create a food delivery website where users can order food from a particular restaurant listed in the website for handling http requests and responses using NodeJS.
6. Implement a program with basic commands on databases and collections using MongoDB.
7. Implement CRUD operations on the given dataset using MongoDB.
8. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.
9. Develop an angular JS form to apply CSS and Events.
10. Develop a Job Registration form and validate it using angular JS.
11. Write an angular JS application to access JSON file data of an employee from a server using $http service.
12. Develop a web application to manage student information using Express and Angular JS.
13. Write a program to create a simple calculator Application using React JS.
14. Write a program to create a voting application using React JS
15. Develop a leave management system for an organization where users can apply different types of leaves such as casual leave and medical leave. They also can view the available number of days using react application.
16. Build a music store application using react components and provide routing among the web pages.
17. Create a react application for an online store which consist of registration, login, and product information pages and implement routing to navigate through these pages.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley,2019.
2. Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress,2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. Brad Green& Seshadri. Angular JS. 1st Edition. O'Reilly Media, 2013.
4. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

CS612PE: INTERNET OF THINGS LAB (PROFESSIONAL ELECTIVE – III)

B.Tech. III Year II Sem.

L	T	P	C
0	0	2	1

Course Objectives:

- To introduce the raspberry PI platform, that is widely used in IoT applications.
- To introduce the implementation of distance sensor on IoT devices.

Course Outcomes:

- Ability to introduce the concept of M2M (machine to machine) with necessary protocols and get awareness in implementation of distance sensor.
- Get the skills to program using python scripting language which is used in many IoT devices.
- Implement interfacing of various sensors with Arduino/Raspberry Pi.
- Demonstrate the ability to transmit data wirelessly between different devices.
- Show an ability to upload/download sensor data on cloud and server.
- Examine various SQL queries from MySQL database.

List of Experiments:

1. Using Raspberry pi
    a. Calculate the distance using a distance sensor.
    b. Interface an LED and switch with Raspberry pi.
    c. Interface an LDR with Raspberrry Pi.

2. Using Arduino
    a. Calculate the distance using a distance sensor.
    b. Interface an LED and switch with Aurdino.
    c. Interface an LDR with Aurdino
    d. Calculate temperature using a temperature sensor.

3. Using Node MCU
    a. Calculate the distance using a distance sensor.
    b. Interface an LED and switch with Raspberry pi.
    c. Interface an LDR with Node MCU
    d. Calculate temperature using a temperature sensor.

4. Installing OS on Raspberry Pi
    i. Installation using PiImager
    ii. Installation using image file
        - Downloading an Image
        - Writing the image to an SD card
            - using Linux
            - using Windows
        - Booting up Follow the instructions given in the URL https://www.raspberrypi.com/documentation/computers/getting-started.html

5. Accessing GPIO pins using Python
    a) Installing GPIO Zero library. update your repositories list:
       install the package for Python 3:
    b) Blinking an LED connected to one of the GPIO pin
    c) Adjusting the brightness of an LED Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) using the in-built PWM wavelength.

6. Create a DJANGO project and an app.
7. Create a DJANGO view for weather station REST API
8. Create DJANGO template
9. Configure MYSQL with DJANGO framework

TEXT BOOKS:

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.

REFERENCE BOOKS:

1. Bernd Scholz-Reiter, Florian Michahelles, “Architecting the Internet of Things”, ISBN 978-3-642-19156-5 e-ISBN 978-3-642-19157-2, Springer, 2016.
2. N. Ida, Sensors, Actuators and Their Interfaces, Scitech Publishers, 2014.

CS613PE: SCRIPTING LANGUAGES LAB (Professional Elective – III)

B.Tech. III Year 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.
- Gain some fluency programming in Ruby, Perl, TCL.
- Write PHP scripts using operators, decision controls, loops, functions (including built-in) in PHP.
- Develop skills to handle form submissions in PHP.
- Create, insert, update a MySQL database.
- Combine PHP code and MySQL to develop full-fledged websites.

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 users 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.

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 Progammers guide by Dabve Thomas Second edition.

REFERENCE BOOKS:

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

CS614PE: MOBILE APPLICATION DEVELOPMENT LAB (Professional Elective – III)

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

Prerequisites: --- NIL—

Course Objectives:

- To learn how to develop Applications in an android environment.
- To learn how to develop user interface applications.
- To learn how to develop URL related applications.

Course Outcomes:

- Understand the working of Android OS Practically.
- Develop, deploy and maintain the Android Applications.
- Apply Layout Management and Multi layout definition techniques to create adaptable User Interface.
- Develop user interface for mobile Application using widgets with event handling.
- Design push notifications for incoming messages
- Deploy applications to the Android marketplace for distribution.

LIST OF EXPERIMENTS:

1. a) 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 the 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 (Datepicker), 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 the right fragment instead of the second screen with the back button. Use Fragment transactions and Rotation event listeners.
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 usernames 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 a 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.

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, 2013.

REFERENCE BOOK:

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

CS615PE: SOFTWARE TESTING METHODOLOGIES LAB (Professional Elective – III)

B.Tech. III Year II Sem.

L	T	P	C
0	0	2	1

Prerequisites:

- A basic knowledge of programming.

Course Objectives:

- To provide knowledge of software testing methods.
- To develop skills in automation of software testing and software test automation management using the latest tools.

Course Outcomes:

- Design and develop the best test strategies in accordance with the development model.
- Design and develop GUI, Bitmap and database checkpoints.
- Develop database checkpoints for different checks.
- Perform batch testing with and without parameter passing.
- Develop the functional and system testing methods.
- Develop applications to solve various issues for object oriented testing.

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.
    a) Bitmap checkpoint for object/window.
    b) Bitmap checkpoint for screen area.
5. Database checkpoint for Default check.
6. Database checkpoint for custom check.
7. Database checkpoint for runtime record check.
    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.
8. Batch testing without parameter passing.
9. Batch testing with parameter passing.
10. Data driven batch.
11. Silent mode test execution without any interruption.
12. 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.

CS606PC: BIG DATA-SPARK

B.Tech. III Year II Sem.	**L**	**T**	**P**	**C**
	0	**0**	**4**	**2**

Course Objectives:

- The main objective of the course is to process Big Data with advance architecture like spark and streaming data in Spark.

Course Outcomes:

- Develop MapReduce Programs to analyze large dataset Using Hadoop and Spark
- Write Hive queries to analyze large dataset Outline the Spark Ecosystem and its components
- Perform the filter, count, distinct, map, flatMap RDD Operations in Spark.
- Build Queries using Spark SQL.
- Apply Spark joins on Sample Data Sets.
- Make use of sqoop to import and export data from hadoop to database and vice-versa.

List of Experiments:

1. To Study of Big Data Analytics and Hadoop Architecture.
   (i) know the concept of big data architecture.
   (ii) know the concept of Hadoop architecture.

2. Loading DataSet in to HDFS for Spark Analysis Installation of Hadoop and cluster management.
   (i) Installing Hadoop single node cluster in ubuntu environment.
   (ii) Knowing the differencing between single node clusters and multi-node clusters.
   (iii) Accessing WEB-UI and the port number.
   (iv) Installing and accessing the environments such as hive and sqoop.

3. File management tasks & Basic linux commands.
   (i) Creating a directory in HDFS
   (ii) Moving forth and back to directories
   (iii) Listing directory contents
   (iv) Uploading and downloading a file in HDFS
   (v) Checking the contents of the file
   (vi) Copying and moving files
   (vii) Copying and moving files between local to HDFS environment
   (viii) Removing files and paths
   (ix) Displaying few lines of a file
   (x) Display the aggregate length of a file
   (xi) Checking the permissions of a file
   (xii) Zipping and unzipping the files with & without permission pasting it to a location
   (xiii) Copy, Paste commands

4. Map-reducing

(i) Definition of Map-reduce
(ii) Its stages and terminologies
(iii) Word-count program to understand map-reduce (Mapper phase, Reducer phase, Driver code)

5. Implementing Matrix-Multiplication with Hadoop Map-reduce.

6. Compute Average Salary and Total Salary by Gender for an Enterprise.

7. (i) Creating hive tables (External and internal)
   (ii) Loading data to external hive tables from sql tables(or)Structured c.s.v using scoop
   (iii) Performing operations like filterations and updations
   (iv) Performing Join (inner, outer etc)
   (v) Writing User defined function on hive tables

8. Create a sql table of employees Employee table with id,designation Salary table (salary ,dept id) Create external table in hive with similar schema of above tables,Move data to hive using scoop and load the contents into tables,filter a new table and write a UDF to encrypt the table with AES-algorithm, Decrypt it with key to show contents

9. (i) Pyspark Definition(Apache Pyspark) and difference between Pyspark, Scala, pandas
   (ii) Pyspark files and class methods
   (iii) get(file name)
   (iv) get root directory()

10. Pyspark -RDD'S
    (i) what is RDD's?
    (ii) ways to Create RDD
    (iii) parallelized collections
    (iv) external dataset
    (v) existing RDD's
    (vi) Spark RDD's operations (Count, foreach(), Collect, join,Cache()

11. Perform pyspark transformations
    (i) map and flatMap
    (ii) to remove the words, which are not necessary to analyze this text.
    (iii) groupBy
    (iv) What if we want to calculate how many times each word is coming in corpus ?
    (v) How do I perform a task (say count the words 'spark' and 'apache' in rdd3) separatly on each partition and get the output of the task performed in these partition ?
    (vi) unions of RDD
    (vii) join two pairs of RDD Based upon their key

12. Pyspark sparkconf-Attributes and applications
    (i) What is Pyspark spark conf ()

(ii) Using spark conf create a spark session to write a dataframe to read details in a c.s.v and later Move that c.s.v to another location

TEXT BOOKS:

1. Spark in Action, Marko Bonaci and Petar Zecevic, Manning.
2. PySpark SQL Recipes: With HiveQL, Dataframe and Graphframes, Raju Kumar Mishra and Sundar Rajan Raman, Apress Media.

WEB LINKS:

1. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01330150584451891225182_shared/overview
2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01258388119638835242_shared/overview
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0126052684230082561692_shared/overview

*MC609: ENVIRONMENTAL SCIENCE

B.Tech. III Year 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 Issues 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. NAPCC-GoI Initiatives.

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

TEXT BOOKS:

1. 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.
6. Introduction to Environmental Science by Y. Anjaneyulu, BS. Publications.

CS611OE: DATA STRUCTURES (Open Elective - I)

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

Prerequisites: Programming for Problem Solving.

Course Objectives:

- Exploring basic data structures such as stacks and queues.
- Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.
- Introduces sorting and pattern matching algorithms.

Course Outcomes:

- Understand the importance of Linked list and Skip list.
- Understand the applications of stacks and queues.
- Understand the importance of different trees in a data structures.
- Understand the Graph and its applications in a real-world environment.
- Understand the importance of hashing.
- Ability to select the data structures that efficiently model the information in a problem.

UNIT -I

Introduction to Data Structures: abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks- Operations, array and linked representations of stacks, stack applications, Queues- operations, array and linked representations.

UNIT -II

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.

Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT –III

TREES: Tree definition, Glossary, Binary trees – Types, Properties, Tree traversals.

Search Trees: Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion, B- Trees, B+ Trees, AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching,

Red –Black, Splay Trees.

UNIT -IV

Graphs: Graph Implementation Methods. Graph Traversal Methods.

Sorting: Quick Sort, Heap Sort, External Sorting-Model for external sorting, MergeSort.

UNIT -V

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

1. FundamentalsofDataStructuresinC,2ndEdition,E.Horowitz,S.SahniandSusanAnderson Freed, Universities Press.
2. DataStructuresusingC–A.S.Tanenbaum,Y.Langsam,andM.J.Augenstein,PHI/PearsonEducation.

REFERENCE BOOK:

1. DataStructures:APseudocodeApproachwithC,2ndEdition,R.F.GilbergandB.A.Forouzan, Cengage Learning

CS612OE: DATABASE MANAGEMENT SYSTEMS (Open Elective – I)

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

Prerequisites: A course on "Data Structures".

Course Objectives:

- To understand the basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

Course Outcomes:

- Gain knowledge of fundamentals of DBMS, database design and normal forms.
- Understand Relational Model and write queries using Relational Algebra and Calculus.
- Master the basics of SQL for retrieval and management of data.
- Be acquainted with the basics of transaction processing and concurrency control.
- Familiarity with database storage structures and access techniques.
- Demonstrate their understanding of key concepts of query evaluation and optimization techniques.

UNIT – I

Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS.

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design with the ER Model.

UNIT - II

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views.

Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT - III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies, FOURTH normal form, FIFTH normal form.

UNIT - IV

Transaction Management: Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions.

UNIT - V

Data Management: Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM).
B+ Trees - A Dynamic Index Structure.

TEXT BOOKS:

1. Database System Concepts, Silberschatz, Korth, McGraw hill, V edition.3rd Edition.
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill.

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education.
3. Introduction to Database Systems, C. J. Date, Pearson Education.
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL,Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.