

**II YEAR II SEMESTER**

| S. No. | Course Code | Course Title                            | L         | T        | P        | Credits   |
|--------|-------------|-----------------------------------------|-----------|----------|----------|-----------|
| 1      | CS401PC     | Discrete Mathematics                    | 3         | 0        | 0        | 3         |
| 2      | SM402MS     | Business Economics & Financial Analysis | 3         | 0        | 0        | 3         |
| 3      | CS403PC     | Operating Systems                       | 3         | 0        | 0        | 3         |
| 4      | CS404PC     | Database Management Systems             | 3         | 1        | 0        | 4         |
| 5      | CS405PC     | Java Programming                        | 3         | 1        | 0        | 4         |
| 6      | CS406PC     | Operating Systems Lab                   | 0         | 0        | 3        | 1.5       |
| 7      | CS407PC     | Database Management Systems Lab         | 0         | 0        | 3        | 1.5       |
| 8      | CS408PC     | Java Programming Lab                    | 0         | 0        | 2        | 1         |
| 9      | *MC409      | Constitution of India                   | 3         | 0        | 0        | 0         |
|        |             | <b>Total Credits</b>                    | <b>18</b> | <b>2</b> | <b>8</b> | <b>21</b> |

**III YEAR I SEMESTER**

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

**III YEAR II SEMESTER**

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

**\*MC609 - Environmental Science – Should be Registered by Lateral Entry Students Only.**

**CS601PC: MACHINE LEARNING****III Year B.Tech. CSE II-Sem**

|          |          |          |          |
|----------|----------|----------|----------|
| <b>L</b> | <b>T</b> | <b>P</b> | <b>C</b> |
| <b>3</b> | <b>1</b> | <b>0</b> | <b>4</b> |

**Prerequisites**

1. Data Structures
2. Knowledge on statistical methods

**Course Objectives**

- This course explains machine learning techniques such as decision tree learning, Bayesian learning etc.
- To understand computational learning theory.
- To study the pattern comparison techniques.

**Course Outcomes**

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

**UNIT - I**

Introduction - Well-posed learning problems, designing a learning system, Perspectives and issues in machine learning

Concept learning and the general to specific ordering – introduction, a concept learning task, concept learning as search, find-S: finding a maximally specific hypothesis, version spaces and the candidate elimination algorithm, remarks on version spaces and candidate elimination, inductive bias.

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

**UNIT - II**

**Artificial Neural Networks-1**– Introduction, neural network representation, appropriate problems for neural network learning, perceptions, multilayer networks and the back-propagation algorithm.

**Artificial Neural Networks-2**- Remarks on the Back-Propagation algorithm, An illustrative example: face recognition, advanced topics in artificial neural networks.

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

**UNIT - III**

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

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

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

**UNIT- IV**

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

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

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

**UNIT - V**

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

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

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

**TEXT BOOK:**

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

**REFERENCE BOOK:**

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

**CS602PC: COMPILER DESIGN****III Year B.Tech. CSE II-Sem**

|          |          |          |          |
|----------|----------|----------|----------|
| <b>L</b> | <b>T</b> | <b>P</b> | <b>C</b> |
| <b>3</b> | <b>1</b> | <b>0</b> | <b>4</b> |

**Prerequisites**

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

**Course Objectives:**

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

**Course Outcomes:**

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

**UNIT - I**

**Introduction:** The structure of a compiler, the science of building a compiler, programming language basics

**Lexical Analysis:** The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex, Finite Automata, From Regular Expressions to Automata, Design of a Lexical-Analyzer Generator, Optimization of DFA-Based Pattern Matchers.

**UNIT - II**

**Syntax Analysis:** Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers, Using Ambiguous Grammars and Parser Generators.

**UNIT - III**

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

**Intermediate-Code Generation:** Variants of Syntax Trees, Three-Address Code, Types and Declarations, Type Checking, Control Flow, Switch-Statements, Intermediate Code for Procedures.

**UNIT - IV**

**Run-Time Environments:** Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management, Introduction to Garbage Collection, Introduction to Trace-Based Collection.

**Code Generation:** Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, A Simple Code Generator, Peephole Optimization, Register Allocation and Assignment, Dynamic Programming Code-Generation.

**UNIT - V**

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

**TEXT BOOK:**

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

**REFERENCE BOOKS:**

1. Lex & Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction, Loudon, Thomson.

**CS603PC: DESIGN AND ANALYSIS OF ALGORITHMS**

III Year B.Tech. CSE II-Sem

|   |   |   |   |
|---|---|---|---|
| L | T | P | C |
| 3 | 1 | 0 | 4 |

**Prerequisites:**

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

**Course Objectives:**

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

**Course Outcomes:**

- Ability to analyze the performance of algorithms
- Ability to choose appropriate data structures and algorithm design methods for a specified application
- Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs

**UNIT - I**

**Introduction:** Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.

**Divide and conquer:** General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

**UNIT - II**

**Disjoint Sets:** Disjoint set operations, union and find algorithms

**Backtracking:** General method, applications, n-queen's problem, sum of subsets problem, graph coloring

**UNIT - III**

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

**UNIT - IV**

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

**UNIT - V**

**Branch and Bound:** General method, applications - Travelling sales person problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

**NP-Hard and NP-Complete problems:** Basic concepts, non deterministic algorithms, NP - Hard and NP-Complete classes, Cook's theorem.

**TEXT BOOK:**

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

**REFERENCE BOOKS:**

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

**CS611PE: CONCURRENT PROGRAMMING (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

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

**Prerequisites**

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

**Course Objectives:** To explore the abstractions used in concurrent programming**Course Outcomes:**

1. Ability to implement the mechanisms for communication and co-ordination among concurrent processes.
2. Ability to understand and reason about concurrency and concurrent objects
3. Ability to implement the locking and non-blocking mechanisms
4. Ability to understand concurrent objects

**UNIT - I**

Introduction - Shared Objects and Synchronization, A Fable, Properties of Mutual Exclusion, The Moral, The Producer–Consumer Problem, The Harsh Realities of Parallelization.

Mutual Exclusion - Time, Critical Sections, 2-Thread Solutions, The Peterson Lock, The Filter Lock, Lamport's Bakery Algorithm.

**UNIT - II**

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

**UNIT - III**

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

**UNIT - IV**

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

**UNIT - V**

Concurrent Queues and the ABA Problem, Concurrent Stacks and Elimination, Transactional Memories

**TEXT BOOKS:**

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

**REFERENCE BOOKS:**

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

**CS612PE: NETWORK PROGRAMMING (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

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

**Course Objectives:**

- To understand inter process and inter-system communication
- To understand socket programming in its entirety
- To understand usage of TCP/UDP / Raw sockets
- To understand how to build network applications

**Course Outcomes:**

- To write socket API based programs
- To design and implement client-server applications using TCP and UDP sockets
- To analyze network programs

**UNIT - I**

**Introduction to Network Programming:** OSI model, Unix standards, TCP and UDP & TCP connection establishment and Format, Buffer sizes and limitation, standard internet services, Protocol usage by common internet application.

**Sockets:** Address structures, value – result arguments, Byte ordering and manipulation function and related functions Elementary TCP sockets – Socket, connect, bind, listen, accept, fork and exec function, concurrent servers. Close function and related function.

**UNIT - II**

**TCP client server:** Introduction, TCP Echo server functions, Normal startup, terminate and signal handling server process termination, Crashing and Rebooting of server host shutdown of server host.

**Elementary UDP sockets:** Introduction UDP Echo server function, lost datagram, summary of UDP example, Lack of flow control with UDP, determining outgoing interface with UDP.

**I/O Multiplexing:** I/O Models, select function, Batch input, shutdown function, poll function, TCP Echo server,

**UNIT - III**

**Socket options:** getsockopt and setsockopt functions. Socket states, Generic socket option IPV6 socket option ICMPV6 socket option IPV6 socket option and TCP socket options.

**Advanced I/O Functions-**Introduction, Socket Timeouts, recv and send Functions, readv and writev Functions, recvmsg and sendmsg Functions, Ancillary Data, How Much Data Is Queued?, Sockets and Standard I/O, T/TCP: TCP for Transactions.

**UNIT - IV**

**Elementary name and Address conversions:** DNS, gethost by Name function, Resolver option, Function and IPV6 support, uname function, other networking information.

**Daemon Processes and inetd Superserver** – Introduction, syslogd Daemon, syslog Function, daemon\_init Function, inetd Daemon, daemon\_inetd Function

**Broadcasting-** Introduction, Broadcast Addresses, Unicast versus Broadcast, dg\_cli Function Using Broadcasting, Race Conditions

**Multicasting-** Introduction, Multicast Addresses, Multicasting versus Broadcasting on A LAN, Multicasting on a WAN, Multicast Socket Options, mcast\_join and Related Functions, dg\_cli Function Using Multicasting, Receiving Mbone Session Announcements, Sending and Receiving, SNTP: Simple Network Time Protocol, SNTP (Continued)

**UNIT - V**

Raw Sockets-Introduction, Raw Socket Creation, Raw Socket Output, Raw Socket Input, Ping Program, Traceroute Program, An ICMP Message Daemon,  
Datalink Access- Introduction, BPF: BSD Packet Filter, DLPI: Data Link Provider Interface, Linux: **SOCK\_PACKET**, **libpcap**: Packet Capture Library, Examining the UDP Checksum Field.  
Remote Login: Terminal line disciplines, Pseudo-Terminals, Terminal modes, Control Terminals, rlogin Overview, RPC Transparency Issues.

**TEXT BOOKS:**

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

**REFERENCE BOOKS:**

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

**CS613PE: SCRIPTING LANGUAGES (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

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

**Prerequisites:**

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

**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 Webrowsers, SOAP and webservices

RubyTk – Simple Tk Application, widgets, Binding events, Canvas, scrolling

**UNIT - II**

Extending Ruby: Ruby Objects in C, the Jukebox extension, Memory allocation, Ruby Type System, Embedding Ruby to Other Languages, Embedding a Ruby Interpreter

**UNIT - III**

Introduction to PERL and Scripting

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

**UNIT - IV**

Advanced perl

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

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

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

**Tk**

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

**TEXT BOOKS:**

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

**REFERENCE BOOKS:**

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

**CS614PE: MOBILE APPLICATION DEVELOPMENT (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

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

**Prerequisites**

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

**Course Objectives**

- To demonstrate their understanding of the fundamentals of Android operating systems
- To improve 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**

- Student understands the working of Android OS Practically.
- Student will be able to develop Android user interfaces
- 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

**CS615PE: SOFTWARE TESTING METHODOLOGIES (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

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

**Prerequisites**

1. A course on "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.

**CS604PC: MACHINE LEARNING LAB****III Year B.Tech. CSE II-Sem**

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

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

**Course Outcomes:** After the completion of the course the student can able to:

- understand complexity of Machine Learning algorithms and their limitations;
- understand modern notions in data analysis-oriented computing;
- be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
- Be capable of performing experiments in Machine Learning using real-world data.

**List of Experiments**

1. The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is the probability that a student is absent given that today is Friday? Apply Baye's rule in python to get the result. (Ans: 15%)
2. Extract the data from database using python
3. Implement k-nearest neighbours classification using python
4. Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids)

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

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

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

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

**CS605PC: COMPILER DESIGN LAB****III Year B.Tech. CSE II-Sem**

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

**Prerequisites**

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

**Co-requisites:**

1. A course on "Web Technologies"

**Course Objectives:**

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

**Course Outcomes:**

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

**List of Experiments****Compiler Design Experiments**

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

```

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

```

`<relop> ::= < | <= | == | >= | > | !=`  
`<addingop> ::= + | -`  
`<term> ::= <term> <multop> <factor> | <factor>`  
`<multop> ::= * | /`  
`<factor> ::= <constant> | <identifier> | <identifier> [ <expression> ]`  
`| ( <expression> )`  
`<constant> ::= <digit> | <digit> <constant>`  
`<identifier> ::= <identifier> <letterordigit> | <letter>`  
`<letterordigit> ::= <letter> | <digit>`  
`<letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z`  
`<digit> ::= 0|1|2|3|4|5|6|7|8|9`  
`<empty>` has the obvious meaning

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

A simple program written in this language is:

```

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

```

**CS621PE: CONCURRENT PROGRAMMING LAB (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

| L | T | P | C |
|---|---|---|---|
| 0 | 0 | 2 | 1 |

**List of Experiments:**

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

**CS622PE: NETWORK PROGRAMMING LAB (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

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

**Course Objectives:**

- To understand inter process and inter-system communication
- To understand socket programming in its entirety
- To understand usage of TCP/UDP / Raw sockets
- To understand how to build network applications

**Course Outcomes:**

- To write socket API based programs
- To design and implement client-server applications using TCP and UDP sockets
- To analyze network programs

**List of Experiments**

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

**TEXT BOOKS:**

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

**CS623PE: SCRIPTING LANGUAGES LAB (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

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

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

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

**Course Outcomes:**

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

**List of Experiments**

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

**CS624PE: MOBILE APPLICATION DEVELOPMENT LAB (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

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

**Prerequisites:** --- NIL---**Course Objectives:**

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

**Course Outcomes:**

- Student understands the working of Android OS Practically.
- Student will be able to develop user interfaces.
- Student will be able to develop, deploy and maintain the Android Applications.

**List of Experiments**

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

**CS625PE: SOFTWARE TESTING METHODOLOGIES LAB (Professional Elective - III)****III Year B.Tech. CSE II-Sem**

| L | T | P | C |
|---|---|---|---|
| 0 | 0 | 2 | 1 |

**Prerequisites:** A basic knowledge of programming.**Course Objectives**

- To provide knowledge of Software Testing Methods.
- To develop skills in software test automation and management using latest tools.

**Course Outcome**

- Design and develop the best test strategies in accordance to the development model.

**List of Experiments:**

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

**\*MC609: ENVIRONMENTAL SCIENCE****III Year B.Tech. CSE II-Sem**

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 0 | 0 |

**Course Objectives:**

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

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

**UNIT - I**

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

**UNIT - II**

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

**UNIT - III**

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

**UNIT - IV**

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

**UNIT - V**

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

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

**TEXT BOOKS:**

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

**REFERENCE BOOKS:**

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