T K R COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

B.Tech.V Semester

S.No.	Course Code	Course Title	L	T	P	Credits
1.	BHSFM	Fundamentals of Management	3	0	0	3
2.	B55PC2	Operating Systems	3	0	0	3
3.	B55PC3	Computer Networks	3	0	0	3
4.	B55PC4	Theory of Computation	3	0	0	3
5.	B55PC5	Design and Analysis of Algorithms	3	0	0	3
6.	B55PC6	Web Technologies	3	0	0	1.5
7.	B55PC7	Operating Systems Lab	0	0	3	1.5
8.	B55PC8	Computer Networks Lab	0	0	3	1.5
9.	B55PC9	Web Technologies Lab	0	0	3	1.5
10.	B55MC1	Mandatory Course	0	3	0	0
Total Credits						**21**

T K R COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

B.Tech.VI SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1.	B56PE1	1.Information Security 2.Introduction to Analytics 3.Distributed Systems	3	0	0	3
2.	B56OE2	OPEN ELECTIVE	3	0	0	3
3.	B56PC3	Complier Design	3	0	0	3
4.	B56PC4	R Programming	2	0	0	2
5.	B56PC5	Data Warehousing andData Mining	3	0	0	3
6.	B56PC6	Object Oriented Analysis and Design	3	0	0	3
7.	B56PC7	R Programming Lab	0	0	2	1
8.	B56PC8	Data Warehousing and Data Mining Lab	0	0	2	1
9.	B56PC9	Object Oriented Analysis and Design Lab	0	0	2	1
10.	B56PC10	Language Processor Lab	0	0	2	1
11.	B56MC2	Mandatory Course	0	3	0	0
Total Credits						**21**

T K R COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

B.Tech.V Semester **L/T/P C**
3/0/0 3

Course Objectives:
Learn concepts in practical aspects of business, interpret, understand and develop the management principles in organizations. learn the basic concepts of organizationits principles and functions.

Course Outcomes:
1. Comparisons of Classical Approach and The Behavioral approach. L4
2. Demonstrate decision making problem solving for the issues in corporate in the organization. L3
3. Illustrate the parameters of the entire organization design and structure. L3
4. Construct the strategically decision in selection, requirement training and development. L5
5. Analyze the qualities of a leader, mentor and coach.

Unit – I:
Introduction to Management: Definition, Nature and Scope, Functions, Managerial Roles, Levels of Management, Managerial Skills, Challenges of Management; Evolution of Management- Classical Approach- Scientific and Administrative Management; The Behavioral approach.

Unit – II:
Planning and Decision Making: General Framework for Planning - Planning Process, Types of Plans. Decision making and Problem solving - Programmed and Non Programmed Decisions, Steps in Problem Solving and Decision Making.

Unit – III:
Organization and HRM: Principles of Organization: Organizational Design & Organizational Structures; Departmentalization, Delegation; Empowerment, Centralization, Decentralization, Recentralization;
Human Resource Management & Business Strategy: Talent Management, Talent Management Models and Strategic Human Resource Planning; Recruitment and Selection; Training and Development; Performance Appraisal.

Unit – IV:
Leading and Motivation: Leadership, Power and Authority, Leadership Styles; Behavioral Leadership, Situational Leadership, Leadership Skills, Leader as Mentor and Coach, Leadership during adversity and Crisis.
Motivation - Types of Motivation; Motivational Theories - Needs Hierarchy Theory,Two

Factor Theory, Theory X and Theory Y.

Unit – V:

Controlling: Control, Types and Strategies for Control, Steps in Control Process, Budgetary and Non- Budgetary Controls. Characteristics of Effective Controls

Text Book:

1. Management Fundamentals, Robert N Lussier, 5e, Cengage Learning, 2013.
2. Fundamentals of Management, Stephen P. Robbins, Pearson Education, 2009.

Reference Books:

1. Essentials of Management, Koontz Kleihrich, Tata Mc - Graw Hill.
2. Management Essentials, Andrew DuBrin, 9e, Cengage Learning, 2012.

Harold Koontz and Heinz Weihrich, 2010, Essentials of Management, TMH

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

B.Tech.V Semester **L/T/P C**
3/0/0 3

Course Objectives:
Understand the OS role in the overall computer system and learn the operationsperformed by OS as a resource manager .

Course Outcomes:
After completion of the course the student will be able:

1. Identify the components and structure of Operating System and the services provided by an Operating System. Analyze the role of Operating System. L4
2. Identify and solve issues related to the critical section problem by using the knowledge, of how to schedule a process, and implement the existing algorithms to evaluate the performance. L5
3. Compare and contrast memory management strategies, Analyze the process scheduling to handle and prevent deadlocks.
4. Analyze how system calls should be implemented for a given file system, Evaluate the performance of demand paging and page replacement algorithms L5
5. Evaluate the role of mass storage in supporting file systems, protection mechanisms in preventing unauthorized access and analyse the architecture, memory & IO management , file system and security features implemented in windows vista. L5

Unit – I:
Overview:
Introduction-Operating system objectives, User view, System view, Operating system definition, Computer System Architecture, OS Structure, OS Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments. Operating System services, User and OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure.

Unit – II:

CPU Scheduling Process:

concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Operations on Processes, System calls-fork(),exec(),wait(),exit(), Inter-process communication-ordinary pipes and named pipes, message queues, shared memory, in Unix

Process Scheduling:

Basic concepts, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real- Time Scheduling, Thread scheduling, Linux scheduling and Windows scheduling. Process Synchronization, Background, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization in Linux and Windows.

Unit – III:

Deadlocks:

System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

Memory Management and Virtual Memory:

Memory ManagementStrategies- Background,Swapping, Contiguous MemoryAllocation, Segmentation, Paging, Structure of Page Table.

Unit – IV:

Virtual Memory Management Background, Demand Paging, Copy-on-Write, Page Replacement, Page Replacement Algorithms, Allocation of Frames, Thrashing, Virtual memory in Windows.

Storage Management File System- Concept of a File, System calls for file operations - open (), read (), write (), close (), seek (), unlink (), Access methods, Directory and Disk Structure, File System Mounting, File Sharing**.**

Unit – V:

File System Implementation: File System Structure, File System Implementation, Directory Implementation, Allocation methods, Free-space Management, Efficiency, and Performance. Overview of Mass Storage Structure.

Protection: System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

Case Study On Windows Vista:

History of windows vista, Programming windows vista, System structure, Process and threads in windows vista, memory management, Caching in windows vista, input/output in windows vista, the Windows NT file system, Security in windows vista.

Text Books:

1. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 8th Edition, Wiley, 2016 India Edition.
2. Operating Systems – Internals and Design Principles, W. Stallings, 7th Edition,Pearson.

Reference Books:

1. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI
2. Operating Systems: A concept-based Approach, 2nd Edition, D.M. Dhamdhere,

Tmh:

3. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
4. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
5. Principles of Operating systems, Naresh Chauhan, Oxford University Press.

T K R COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

COMPUTER NETWORKS -B55PC3

B.Tech V Semester **L/T/P C**
3/0/0 3

Course Objective:
Identify the components required to build different types of networks , choose the required functionality at each layer for given application.

Course Outcomes:
After completion of the course the student will be able:

1. Compare and contrast different network models and their significance in networking architecture. L4
2. Develop elementory protocols by applying the knowledge of principles of framing and error control to design robust and reliable data link layer protocols. L3
3. Make use of various routing algorithms and analyze them to design efficient and scalable network architectures. L4
4. Analyze and compare the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) in terms of their features and Evaluate congestion control algorithms. L4
5. Utilize the knowledge on how to address and resolve issues related to application layer protocols, ensuring seamless interaction and data exchange in online environment. L3

Unit – I:
Introduction: OSI, TCP/IP and other networks models, Examples of Networks: Novell Networks, Arpanet, Internet, Network Topologies WAN, LAN, MAN.
Physical Layer: Transmission media copper, twisted pair wireless, switching and encoding asynchronous communications; Narrow band, broad band ISDN and ATM.

Unit – II:
Data link layer: Design issues, framing, error detection and correction, CRC, Elementary Protocol- stop and wait, Sliding Window, Slip, Data link layer in HDLC,Internet, ATM.
Medium Access sub layer: ALOHA, MAC addresses, Carrier sense multiple access. IEEE 802.X Standard Ethernet, wireless LANS. Bridges

Unit – III:
Network Layer: Virtual circuit and Datagram subnets-Routing algorithm shortest path routing, Flooding, Hierarchical routing, Broad cast, Multi cast, distance vector routing. Dynamic routing – Broadcast routing. Rotary for mobility, The Network layer in the internet

and in the ATM Networks.

Unit – IV:

Transport Layer: Transport Services, Connection management, TCP and UDP protocols; Congestion, Control Algorithms – General Principles – of Congestion prevention policies ATM AAL Layer Protocol.

Unit – V:

Application Layer – Domain name system, SNMP, Electronic Mail (SMTP, POP3, IMAP, MIME) the WWW, HTTP.

Text Book

1.Computer Networks — Andrew S Tanenbaum, 4th Edition. Pearson Education/PHI
2.Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

Reference Books

1.An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education.
2.Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson.

T K R COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

THEORY OF COMPUTATION - B55PC4

B.Tech.V Semester **L/T/P C**

3/0/0 3

Course Objective:
To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.

Course Outcomes:
After completion of the course the student will be able to

1. Analyze and Differentiate between Deterministic and Nondeterministic Finite Automata, and analyze them to apply for solving text search problems. L4
2. Construct regular expressions, Apply the algebraic laws to convert between DFA and regular expressions to Sove the solutions for language recognition problems. L3
3. Analyze context-free grammars, derivations, Identify and resolve ambiguity in grammars and languages. L4
4. Utilize Chomsky Normal Form conversions and evaluate the computational intricacies in transforming grammars and pushdown automata. L3
5. Assess undesirability, apply concepts to recursively enumerable languages and Turing machines, and analyze the implications of computational limits in theoretical computer science. L5

Unit – I:
Introduction: Introduction to Finite Automata, Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems. Deterministic Finite Automata, Nondeterministic Finite Automata, an application: Text Search, Finite Automata with Epsilon-Transitions, Finite automata with output - Mealy and Moore machines, Equivalence of Mealy and Moore machines.

Unit – II:
Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Automata and Regular expressions, Converting DFA's to Regular Expressions, Converting Regular Expressions to DFA, Properties of Regular Languages-Pumping Lemma for Regular Languages, Applications of the Pumping Lemma, 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 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, Deterministic Pushdown Automata, non-deterministic pushdown automata, power of Deterministic Pushdown Automata and Non-DeterministicPushdown Automata.

Unit – IV:

Normal Forms for Context- Free Grammars: The Pumping Lemma for Context- Free Languages, Closure Properties of Context-Free Languages. Decision Properties of CFL's - Complexity of Converting among CFG's and PDA's, Running time of conversions to Chomsky Normal Form.

Introduction to Turing Machines: Problems That Computers Cannot Solve, The Turing Machine, Programming Techniques for Turing Machines, Extensions to the basic Turing machine, Restricted Turing Machines, Turing Machines, and Computers.

Unit – V:

Undesirability: A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Post's Correspondence Problem, Other Undecidable Problems,

Intractable Problems: Polynomial time and space, NP-complete problems, The Classes P and NP, An NP-Complete Problem.

Text Books:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, CengageLearning.

Reference Books:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, CambridgeUniversity Press.
4. Introduction to Formal languages Automata Theory and Computation KamalaKrithivasan, Rama R, Pearson.
5. Theory of Computer Science – Automata languages and computation,Mishra and Chandrashekaran, 2nd edition, PHI.

T K R COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

DESIGN AND ANALYSIS OF ALGORITHMS - B55PC5

B.Tech.V Semester **L/T/P C**
3/0/0 3

Course Objective:
Understand the design paradigms for developing an algorithm and analyzing it for a given problem.

Course Outcomes:
After completion of course the students will be able to:
1. Analyze mathematical analysis methods to explore algorithm performance and apply the divide-and-conquer technique for resolving computing problems. L4
2. Demonstrate disjoint set operations and implement the backtracking technique to address computing problems. L3
3. Implement the Greedy method to solve diverse computing problems. L3
4. Develop effective algorithms for typical engineering design scenarios through the application of the dynamic programming technique. L5
5. Solve intricate problems by applying the branch-and-bound technique, and analyze the characteristics of NP-hard and NP-complete problems. L4

Unit – I:
Introduction-Algorithm definition, Algorithm Specification, Performance Analysis- Space complexity, Time complexity, Randomized Algorithms. Divide and conquer- General method, applications – Binary search, Merge sort, Quick sort, Strassen's Matrix Multiplication.

Unit – II:
Disjoint set operations, union and find algorithms, AND/OR graphs, Connected Components and Spanning trees, Bi-connected components Backtracking-General method, applications the 8-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

Unit – III:
Greedy method- General method, applications- Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Single source shortest path problem.

Unit – IV:
Dynamic Programming- General Method, applications- Chained matrix multiplication,
All pairs shortest path problem, Optimal binary search trees, 0/1 knapsack problem,

Reliability design, Traveling sales person problem.

Unit – V:

Branch and Bound- General Method, applications-0/1 Knapsack problem, LC Branch and Bound solution, FIFO Branch and Bound solution, Traveling sales person problem.NP-Hard and NP- Complete problems- Basic concepts, Non-deterministic algorithms, NP – Hard and NP-Complete classes, Cook's theorem.

Text Books:

1. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, SartajSahni and S. Rajasekharan, Universities Press.
2. Design and Analysis of Algorithms, P. H. Dave, H. B. Dave, 2nd edition, Pearson Education.

Reference Books:

1. Algorithm Design: Foundations, Analysis and Internet examples, M. T. Goodrichand R. Tomassia, John Wiley and sons.
2. Design and Analysis of Algorithms, S. Sridhar, Oxford Univ. Press
3. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson Education.
4. Foundations of Algorithms, R. Neapolitan and K. Naimipour, 4th edition, Jones andBartlett Student edition.
5. Introduction to Algorithms, 3rd Edition, T. H. Cormen, C. E. Leiserson, R. L. Rivest,and C. Stein, PHI.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

WEB TECHNOLOGIES - B55PC6

B.Tech.V Semester **L/T/P C**
3/0/0 15

Course Objectives:
To introduce PHP language for server side scripting and introduce XML and processing of XML Data with Java , server side programming with JavaServlets and JSP, Client side scripting with Java script and AJAX.

Course Outcomes:
After completion of the course the student will be able to:

1. Develop a web application using PHP and MySQL to Recall and reproduce PHP syntax and Identify different data types in PHP and explain the purpose and use of control structures, role of functions in PHP to handle file permissions, error checking, and security considerations when working with file operations in PHP. L3
2. Make use of XML and their concepts to store, navigate, transport, manipulate, and parse the content and Implement XML parsing using DOM and SAX parsers in Java. L3
3. Identify the components of the Servlet API. Recall the basic concepts of servlets, such as the servlet lifecycle and deployment and Apply knowledge of servlet parameters to read and process client data, initialization parameters in servlets. Analyze HTTP request and response handling in servlets. L3
4. Develop JSP pages with appropriate declarations, directives, and expressions and implement database connectivity in JSP using JDBC. L3
5. Apply knowledge of JavaScript to create and use functions, create a simple AJAX application to dynamically update web page content Implement event handlers for various user interactions on web pages. L3

Unit – I:
Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads, Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

Unit – II:

XML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemas, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

Unit – III:

Introduction to Servlets: Common Gateway Interface (CGI), Lifecycle of a Servlet, deploying a servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

Unit – IV:

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

Unit – V:

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

Text Books:

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

Reference Books:

1. Web Programming, building internet applications, Chris Bates 2nd edition, Wiley Dreamtech.
2. Java Server Pages –Hans Bergsten, SPD O'Reilly Java Script, D. Flanagan,O'Reilly,SPD.
3. Beginning Web Programming-Jon Duckett WROX.
4. Programming World Wide Web, R. W. Sebesta, Fourth Edition, Pearson.
5. Internet and World Wide Web – How to program, Dietel and Nieto, Pearson.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERING-R18

OPERATING SYSTEMS LAB - B55PC7

B.Tech.V Semester **L/T/P C**
0/0/3 1.5

Course Objective:

To understand the theoretical approach of an Operating System

Course Outcomes:

After completion of the course the student will be able to:

1. Test commands for file manipulation, navigation, and system information on both Linux and Windows platforms. L4
2. Develop basic shell programs in Linux for tasks such as addition, digit sum, finding the greatest number, pattern searching, and number printing. L3
3. Implement and test C programs for addition, digit sum, finding the greatest number, Armstrong number check, number printing, deadlock prevention , page replacement algorithms, paging, segmentation and Inter Process Communication . L4

Use Linux operating system and GNU C compiler. List of Programs:

1.Practice operating system basic commands

a) DOS: DIR, MD, CD, CHDIR, CLS, COPYCON, TYPE, MOVE, COPY, DEL,

b) LINUX: ls, cat, rm, pwd, cal, wc, mkdir, rmdir, cd, grep, touch, truncate, cp, mv, echo, cut, locate, who, head, tail, sort, find, help, concat, chmod, umask, man, name, hostname, ping, clear, ln, date, prompt, ren.

c) Practice on windows

2.Practice simple shell programs:

a) Addition of two integer numbers
b) To find sum of individual digits of a given number
c) To find greatest number among three numbers
d) To search the pattern in a given file using grep
e) To print numbers from 1 to n.

3.Execution of C programs in Linux environment

a) Addition of two integer numbers
b) To find sum of individual digits of a given number
c) To find greatest number among three numbers
d) Test whether the given number is Armstrong or not
e) To print numbers from 1 to n.

4.Practice system calls: open, close, read, write, exit

5.Write a C program to copy the contents of one file to another using system calls

6.Write a C program to implement the ls | sort command. (Use unnamed Pipe)

7.Construct C programs for the following CPU scheduling algorithms:

a) Round Robin
b) SJF

8.Create C Programs to simulate the following CPU Scheduling algorithms:
a) FCFS
b) Priority

9. Solve the Dining- Philosopher problem using semaphores in C.Implement IPC between two unrelated processes.

10.Develop C programs to implement IPC between two processes using
a) Message queues
b) Shared memory.

11.Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

12.Design C programs to simulate the following page replacement algorithms:
a) FIFO
b) LRU
c) LFU

13.Develop C programs to simulate the following techniques of memory management:
a) Paging
b) Segmentation

Construct C Programs to simulate the following File Organization techniques:
c) Single level directory
d) Two level
e) Hierarchical

14.Develop C Programs to simulate the following File Allocation Methods:
a) Contiguous
b) Linked
c) Indexed

15.Design a C program to simulate Bankers Algorithm for Dead Lock Avoidance.

16.Design a C program to simulate Bankers Algorithm for Dead LockPrevention**.**

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

TECH. COMPUTER SCIENCE & ENGINEERING– R18

COMPUTER NETWORKS LAB -B55PC8

B.Tech.V Semester **L/T/P C**
0/0/3 1.5

Course Objectives:

To understand the functionalities of various layers of OSI model, and the operating System functionalities.

Course Outcomes:

After completion of the course the student will be able to:

1. Implement different DLL framing methods and CRC polynomials. L3
2. Apply appropriate algorithm for the finding of shortest route. L3
3. Construct different encryption and decryption techniques. L3

List Of Experiments:

1. Implement the data link layer framing methods such as character, characterstuffing, and bit stuffing.
2. Implement on a data set of characters the three CRC polynomials – CRC 12, CRC16 and CRC CCIP.
3. Implement Dijkstra‘s algorithm to compute the Shortest path thru a graph.
4. Take an example subnet graph with weights indicating delay between nodes. Nowobtain Routing table art each node using distance vector routing algorithm
5. Take an example subnet of hosts. Obtain broadcast tree for it.
6. Take a 64 bit playing text and encrypt the same using DES algorithm.
7. Write a program to break the above DES coding
8. Using RSA algorithm encrypts a text data and Decrypt the same

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)
B. TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

WEB TECHNOLOGIES LAB -B55PC9

B.Tech.V Semester **L/T/P C**
0/0/3 1.5

Course Objective:
Learn program web applications using HTML, Javascript , AJAX, PHP, Tomcat Server, Servlets, JSP.

Course Outcomes:
After completion of the course the student will be able to:
1. Develop and build interactive web pages using HTML and JavaScript and implement basic JavaScript functions for user interactions and data manipulation. L6
2. Develop and create XML documents and validate them against a given Document Type Definition (DTD) or XML Schema and Construct the structure of an XML document and represent it using a DOM tree and SAX. L6
3. Develop a registration form in PHP that adds user information to the XML file and the data stored in the XML file. L3
4. Develop and Implement a database connection in a web application using a server-side scripting language (e.g., PHP, Java). L3
5. Analyze the structure and content of cookies stored in the browser. L4

List Of Experiments:
1. Write an HTML page including JavaScript that takes a given set of integer numbers and shows them after sorting in descending order.
2. Write an HTML page including any required Javascript that takes a number from onetext field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show "out of range" and if it is not a number, it should show "not a number" message in the result box.
3. Write an HTML page that has one input, which can take multi-line text and a submit button. Once the user clicks the submit button, it should show the number of characters, words and lines in the text entered using an alert message. Words are separated with white space and lines are separated with new line character.

Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).

4. Create an XML document that contains 10 users information. Write a Java

program,which takes User Id as input and returns the user details by taking the user information from the XML document using (a) DOM Parser and (b) SAX parser Implement the following web applications using (a) PHP, (b) Servlets and (c) JSP:

5. A user validation web application, where the user submits the login name and password to the server. The name and password are checked against the data alreadyavailable in Database and if the data matches, a successful login page is returned.

Otherwise a failure message is shown to the user.

6. Modify the above program to use an xml file instead of database.
7. Modify the above program to use AJAX to show the result on the same page belowthe submit button.
8. A simple calculator web application that takes two numbers and an operator (+, -, /, * and %) from an HTML page and returns the result page with the operationperformed on the operands.
9. Modify the above program such that it stores each query in a database and checks the database first for the result. If the query is already available in the DB, it returns the value that was previously computed (from DB) or it computes the result and returns it after storing the new query and result in DB.
10. A web application takes a name as input and on submit it shows a hello page where is taken from the request. It shows the start time at the right top corner of the page and provides a logout button. On clicking this button, it should show a logout page with Thank You message with the duration of usage (hint: Use session to store nameand time).
11. A web application that takes name and age from an HTML page. If the age is less than it should send a page with "Hello , you are not authorized to visit this site" message, where should be replaced with the entered name. Otherwise it should send "Welcome to this site" message.
12. A web application for implementation: The user is first served a login page which takes user's name and password. After submitting the details the server checks thesevalues against the data from a database and takes the following decisions.
13. If name and password matches, serves a welcome page with user's full name. If name matches and password doesn't match, then serves "password mismatch" page If name is not found in the database, serves a registration page, where user's full nameis asked and on submitting the full name, it stores, the login name, password and fullname in the database (hint: use session for storing the submitted login name and password)
12. A web application that lists all cookies stored in the browser on clicking "List Cookies" button. Add cookies if necessary.
13. A web application for implementation: The user is first served a login page which takes user's name and password. After submitting the details the server checks thesevalues against the data from a database and takes the following decisions. If name and password matches, serves a welcome page with user's full name. If name matches and password doesn't match, then serves "password mismatch" page If name is not found in the database, serves a registration page, where user's full nameis asked and on submitting the full name, it stores, the login name, password and fullname in the database (hint: use session for storing the submitted login name and password)
14. A web application that lists all cookies stored in the browser on clicking "List Cookies" button. Add cookies if necessary.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

INFORMATION SECURITY - B56PE11

B.Tech VI Semester **L/T/P C**

3/0/0 3

Course Objectives:

Learn fundamental concepts of symmetric and asymmetric cipher models, fundamental concepts of authentication, network security andweb security protocols.

Course Outcomes:

Upon completion of the course the students will be able to:

1. Illustrate the concepts and techniques of cryptography. L3
2. Explore the distinctions and similarities between symmetric and asymmetric key ciphers. L4
3. Analyze the differences and similarities between symmetric and asymmetric key ciphers through a comparison and-contrast approach. L4
4. Develop IP security mechanisms and illustrate their implementation through case studies on cryptography and security. L5
5. Examine and differentiate various web security considerations through a comparative analysis. L4

UNIT – I:

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

UNIT – II:

Symmetric key Asymmetric Ciphers: Block Cipher principles, DES, AES, Blowfish, RC5, IDEA, Block cipher operation, Stream ciphers. Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Diffie-Hellman Key Exchange,Eliptic Curve Cryptography.

UNIT – III:

Cryptographic Hash Functions: Message Authentication, MD5,Secure Hash Algorithm (SHA-512), Message authentication codes: Authentication requirements, HMAC,Digital signatures, Elgamal Digital Signature Scheme. Key Management and Distribution: Symmetric Key Distribution Using Symmetric & Asymmetric Encryption, Distribution of Public Keys, Kerberos, X.509 Authentication Service, Public – Key Infrastructure.

UNIT – IV:

IP Security: IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, Internet KeyExchange Case Studies on Cryptography and security: Secure Multiparty Calculation, Virtual Elections, Single sign On, Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability.

UNIT – V:

Web Security: Web security considerations, Pretty Good Privacy, S/MIME, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH) Wireless Network Security: Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security.

Text Book:

1. Cryptography and Network Security – Principles and Practice: William Stallings, Pearson Education, 6th Edition.

Reference Books:

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

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

INTRODUCTION TO ANALYTICS - B56PE12

B.Tech.VI Semester **L/T/P C**
3/0/0 3

Course Objectives:

Introduce the terminology, technology and its applications and the concept of Analytics for business, tools, technologies & programming languages which is used in day to day analytics cycle.

Course Outcomes:

After completion of the course the student will be able to:

1. Analyze the effective project management skills, communication, and the ability to tackle challenges encountered during analytics projects. L4
2. Classify both technical statistical knowledge and essential soft skills for effective teamwork and collaboration. L4
3. Improve collaboration between R, SQL, and NoSQL for comprehensive data solutions. L6
4. Determine comprehensive understanding of statistical concepts and practical skills for data analysis. L5
5. Utilize data analysis techniques for fostering a holistic understanding of complex systems and business challenges. L3

UNIT – I:

Introduction to Analytics and R programming : Introduction to R, R Studio (GUI): R Windows Environment, introduction to various data types, Numeric, Character, date, data frame, array, matrix etc., Reading Datasets, Working with different file types .txt,. csv etc. Outliers, Combining Datasets, R Functions and loops. Manage your work to meet requirements (NOS 9001): Understanding Learning objectives, Introduction to work & meeting requirements, Time Management, Work management & prioritization, Quality & Standards Adherence.

UNIT – II:

Summarizing Data & Revisiting Probability: Summary Statistics – Summarizing data with R, Probability, Expected, Random, Bivariate Random variables, Probability distribution. Central Limit Theorem etc. Work effectively with Colleagues (NOS 9002): Introduction to work effectively, Team Work, Professionalism, Effective Communication skills, etc.

UNIT – III:

SQL using R Introduction to No SQL, Connecting R to No SQL databases. Excel and R

integration with R connector.

UNIT – IV:

Correlation and Regression Analysis : Regression Analysis, Assumptions of OLS Regression, Regression Modeling. Correlation, ANOVA, Forecasting, Heteroscedasticity, Autocorrelation, Introduction to Multiple Regression etc.

UNIT – V:

Understand the Verticals – Engineering, Financial and others (NOS 9002) Understanding systems viz. Engineering Design, Manufacturing, Smart Utilities, Production lines, Automotive, Technology etc. Understanding Business problems related to various businesses. Requirements Gathering: Gathering all the data related to Business objective

Text Book:

1. Student's Handbook for Associate Analytics.

Reference Books:

1. Introduction to Probability and Statistics Using R, ISBN: 978-0-557-24979-4, is a textbook written for an undergraduate course in probability and statistics.
2. An Introduction to R, by Venables and Smith and the R Development Core Team. This may be downloaded for free from the R Project website (http://www.rproject.org/see Manuals). There are plenty of other free references available from the R Project website.
3. Montgomery, Douglas C., and George C. Runger, Applied statistics and probability for engineers. John Wiley & Sons, 2010.
4. Time Series Analysis and Mining with R. Yanchang Zhao.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

DISTRIBUTED SYSTEMS (B56PE13)

B.Tech VI Semester	**L/T/P**	**C**
	3/0/0	**3**

Course Objectives:

Understand theoretical concepts, namely, virtual time, agreement and consensus protocols . Group Communication , RPC , DFS and DSM.

Course Outcomes:

Upon completion of course the student will be able be:

1. Acquire a comprehensive understanding of the fundamental concepts of distributed systems, including their key characteristics, resource sharing mechanisms, and the challenges they pose. L3
2. Demonstrate the ability to design and implement complex distributed system architectures, incorporating principles of fault tolerance, scalability, and security. L6
3. Analyze and design distributed algorithms, demonstrating proficiency in solving complex computational problems within distributed systems. L4
4. Be Proficient in evaluating, configuring, and troubleshooting network protocols to ensure seamless communication within distributed systems. L3
5. Achieve proficiency in assessing and implementing robust security measures to safeguard distributed systems against vulnerabilities and threats. L3

UNIT – I

Characterization of Distributed Systems: Introduction, Examples of Distributed Systems, Resource Sharing and the Web, Challenges. System Models: Introduction, Architectural Models, Fundamental Models.

UNIT – II

Time and Global States: Introduction, Clocks Events and Process States, Synchronizing Physical Clocks, Logical Time and Logical Clocks, Global States, Distributed Debugging. Coordination and Agreement: Introduction, Distributed MutualExclusion, Elections, Multicast Communication, Consensus and Related Problems.

UNIT – III

Inter Process Communication: Introduction, the API for the Internet Protocols, External Data Representation and Marshalling, Client-Server Communication, Group Communication, Case Study: IPC in UNIX.

Distributed Objects and Remote Invocation: Introduction, Communication between Distributed Objects, Remote Procedure Call, Events and Notifications, Case Study:

JAVA RMI.

UNIT – IV

Distributed File Systems: Introduction, File Service Architecture, Case Study 1: Sun Network File System, Case Study 2: The Andrew File System.

Name Services: Introduction, Name Services and the Domain Name System, Directory Services, Case Study of the Global Name Services.

Distributed Shared Memory: Introduction, Design and Implementation Issues, Sequential Consistency and IVY case study, Release Consistency, Munin Case Study, Other Consistency Models.

UNIT – V

Transactions and Concurrency Control: Introduction, Transactions, Nested Transactions, Locks, Optimistic Concurrency Control, Timestamp Ordering, Comparison of Methods for Concurrency Control. Distributed Transactions: Introduction, Flat and Nested Distributed Transactions, Atomic Commit Protocols, Concurrency Control in Distributed Transactions, Distributed Deadlocks, Transaction Recover.

Text Book:

1. Distributed Systems, Concepts and Design, George Coulouris, J Dollimore and Tim Kindberg, Pearson Education, 4th Edition, 2009.

Reference Books:

1. Distributed Systems, Principles and Paradigms, Andrew S. Tanenbaum, Maarten Van Steen, 2nd Edition, PHI.
2. Distributed Systems, An Algorithm Approach, Sukumar Ghosh, Chapman & Hall/CRC, Taylor & Fransis Group, 2007.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

COMPILER DESIGN (B56PC3)

B.Tech VI Semester	**L/T/P**	**C**
	3/0/0	**3**

Course Objectives:
Understand the various phases of a compiler.

Course Outcomes:
After completion of the course the student will be able to:

1. Analyze and comprehend the lexical analyzer generator specifying rules and patterns for token recognition. L4
2. Develop top-down and bottom-up parser for a programming language with a specific context-free grammar. L6
3. Assess the complexities involved in generation of various SDDs during intermediate code generation. L5
4. Frame and apply dynamic programming code generation techniques. L3
5. Evaluate the effectiveness of machine-independent optimization techniques in application scenarios. L5

UNIT – I
Introduction, Language Processors, structure of 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, Parser Generators.

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

Run-Time Environments: Storage organization, 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 Optimizations: The Principle Sources of Optimization, Introduction to Data – Flow Analysis, Foundations of Data-Flow Analysis, Constant Propagation, Partial Redundancy Elimination, Loops in Flow Graphs.

Text Books:

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

Reference Books:

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

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

PROGRAMMING (B56PC4)-R18

B.Tech VI Semester **L/T/P C**
2/0/0 2

Course Objective:

Gain knowledge on statistical data manipulation and analysis.

Course Outcomes:

After competition of this course the students will be able to:

1. Apply statistical techniques, basic functions and Vectors in R to analyze and interpret data. L4
2. Apply matrix, Lists and Array operations to solve problems and perform data analysis tasks. L6
3. Apply appropriate functions and operations to analyze and summarize data frames, factors, and tables. L4
4. Apply functions to solve specific problems and analyze data using a variety of mathematical and statistical methods L6
5. Analyze file operations, string manipulation functions and interpret data using graphs and visualizations. L5

UNIT – I

Introduction to R

Introduction, Functions, Preview of Some Important R Data Structures, Regression Analysis of Exam Grades, Startup and Shutdown, Getting Help, The help() Function, The example() Function. **Vectors** Scalars, Vectors, Arrays, and Matrices, Declarations, Common Vector Operations, Using all() and any(),Vectorized Operations, NA and NULL Values, Filtering, Vectorized if-then-else.

UNIT – II

Matrices And Arrays

Creating Matrices, General Matrix Operations, Applying Functions to Matrix Rows and Columns, More on the Vector/Matrix Distinction, Avoiding Unintended Dimension Reduction, Naming Matrix Rows and Columns, Higher-Dimensional Arrays.

Lists

Creating Lists, General List Operations, Accessing List Components and Values Applying Functions to Lists, Recursive Lists.

UNIT-III

Data Frames

Creating Data Frames, Other Matrix-Like Operations, Merging Data Frames, Applying Functionsto Data Frames.

Factors And Tables

Factors and Levels, Common Functions Used with Factors, Working with Tables,Other Factor-and Table-Related Functions.

UNIT – IV

R Programming Structures: Control Statements, Arithmetic and Boolean Operators and Values, Default Values for Arguments, Return values, Functions Are Objects, Environment and Scope Issues, No Pointers in R, Writing Upstairs, Recursion, Replacement Functions, Anonymous Functions.

Math And Simulations In R: Math Functions, Functions for Statistical, Sorting, Set Operations.

UNIT – V

FILES: Accessing the Keyboard and Monitor, Reading and Writing Files, Accessing the Internet.

String Manipulation: String-Manipulation Functions.

Graphics: Creating Graphs, Customizing Graphs

Text Book:

1. The Art of R Programming by Norman Matloff-No Starch Press.

Reference Books:

1. R Programming for Bioinformatics by Robert Gentleman—CRC Press.
2. Data Analytics using R by Seema Acharya-TMH.
3. Hands-On Programming with R by Grrett Grolemund-OREILLY.
4. Beginners guide for Data Analytics using R by Jeeva Jose-Khanna Publications.
5. R for Beginners by Sandip Bakshit-TMH.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

DATA WAREHOUSING AND DATA MINING (B56PC5)

B.Tech VI Sem **L/T/P C**
3/0/0 3

Course Objective:

To learn the approaches designed for handling large data with reference to data models.

Course Outcomes:

After Completion of Course the student will be able to:

1. Analyze fundamental concepts, key characteristics, architecture, logical data modeling, and various analysis techniques, encompassing OLAP concepts and server architectures. L4
2. Apply foundational knowledge in Data Mining, and use in Data preprocessing techniques. Demonstrate understanding of Data Transformation and apply proficiency in exploring measures of similarity and Dissimilarity for analyzing data. L3
3. Apply a comprehensive understanding to define problems, utilize APRIORI principles for generating frequent item sets, focusing in compact representations like maximal and closed frequent item sets. L3
4. Analyze problem definitions, employ general strategies for solving classification problems, and assess classifier performance and various classification techniques including their characteristics. L4
5. Analysis of clustering algorithms, addressing key issues and techniques and conducting a thorough analysis of strengths and weaknesses. L4

UNIT – I

Data warehouse: Introduction to Data warehouse, Difference between operational database systems and data warehouses. Data warehouse Characteristics, Data warehouse Architecture and its Components, Extraction – Transformation – Loading, Logical (Multi – Dimensional), Data Modeling, Schema Design, Star and Snow – Flake Schema, Fact Consultation, Fact Table, Fully Addictive, Semi – Addictive, Non Addictive Measures; Fact Consultation, Fact Table, Fully Addictive, Semi – Addictive, Non Addictive Measures; Fact – Less – Facts, Dimension Table Characteristics; OLAP Cube, OLAP Operations, OLAP Server Architecture – ROLAP, MOLAP and HOLAP.

UNIT – II

Introduction to Data Mining: Introduction, What is Data Mining, Definition, KDD, Challenges, Data Mining Tasks, Data Preprocessing, Data Cleaning, Missing data, Dimensionality Reduction, Feature Subset Selection, Discretization and Binaryzation, Data Transformation; Measures of Similarity and Dissimilarity- Basics.

UNIT – III

Association Rules: Problem Definition, Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation; APRIOIRI Algorithm, The Partition Algorithms, FP-Growth Algorithms, Compact Representation of Frequent Item Set- Maximal Frequent Item Set, Closed Frequent Item Set.

UNIT – IV

Classification: Problem Definition, General Approaches to solving a classification problem, Evaluation of Classifiers, Classification techniques, Decision Trees-Decision tree Construction, Methods for Expressing attribute test conditions, Measures for Selecting the Best Split, Algorithm for Decision tree Induction; Naïve – Bayes Classifier, Bayesian Belief Networks; K- Nearest neighbor classification-Algorithm and Characteristics.

UNIT – V

Clustering: Problem Definition, Clustering Overview, Evaluation of Clustering Algorithms, Partitioning Clustering-K-Means Algorithm, K-Means Additional issues, PAM Algorithm; Hierarchical Clustering-Agglomerative Methods and divisive methods, Basic Agglomerative Hierarchical Clustering Algorithm, Specific techniques, Key Issues in Hierarchical Clustering, Strengths and Weakness; Outlier Detection.

Text Books:

1.Data Mining- Concepts and Techniques- Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers, Elsevier, 2 Edition, 2006.
2.Introduction to Data Mining, Pang-Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education.

Reference Books:

1.Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities Press.
2.Data Warehousing Fundamentals, Paulraj Ponnaiah, Wiley Student Edition.
3.The Data Warehousinh Life Cycle Toolkit – Ralph Kimbal. Wiley Student Edition.
4.Data Mining, Vikaram Pudi, P Radha Krishna, Oxford University Press.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

OBJECT ORIENTED ANALYSIS AND DESIGN (B56PC6)

B.Tech VI Sem **L/T/P C**
3/0/0 3

Course Objectives:
Learn object-oriented development life cycle with the E-R models. Recognize when to use generalization, aggregation, and composition relationships. Specify different types static and behavioral diagram.

Course Outcome:
After completion of the course the student will be able to:

1. List the importance and use of basic principles in object-oriented modelling. L4
2. Develop pictorial representation by making use of basic structural modeling concepts for any application development. L6
3. Distinguish different system behavioral modeling techniques. L4
4. Categorize advanced behavioral modelling for visualizing flow control of objects and activities. L4
5. Develop and design a document using UML for simple and complex scenarios of the unified Library System. L6

UNIT – I
Introduction to UML: Importance of modeling, principles of modelling, object oriented modelling, conceptual model of the UML, Architecture, Software Development Life Cycle.

UNIT – II
Basic Structural Modelling: Classes Relationships, common Mechanisms, and diagrams. Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

UNIT-III
Basic Behavioral Modelling-I: Interactions, Interaction diagrams.
Basic Behavioral Modeling – II: Use cases, Use case Diagrams, Activity Diagrams.

UNIT IV
Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

UNIT – V

Patterns and Frameworks, Artificer Diagrams. Case Study: The Unified libraryapplication.

Text Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling LanguageUser Guide, Pearson Education 2nd Edition
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit,WILEY-Dreamtech India Pvt. Ltd.

Reference Books:

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML PearsonEducation.
2. Pascal Rogues: Modeling Software Systems Using UML2, WILEY- DreamtechIndia Pvt. Ltd.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERING

COURSE STRUCTURE -R18

PROGRAMMING LAB (B56PC7)

B.Tech VI Semester **L/T/P C**
0/0/2 1

Course Objective:
The basics of statistical computing and data analysis, use R for analytical programming. implement data structure in R. R loop functions and debugging tools. Object-oriented programming concepts in R. Data visualization in R.

Course Outcomes:
At the end of this course, you have all the computational tools to:
1. Solve programing using R and other programming languages. L3
2. Apply empirical economic analysis. L3
3. Analyze online data science challenges and generate maps, plots based on the data. L4

List of programs:
1.R and Rstudio set-up.
2.R types, vectors and writing functions.
3.More R types, vectorization and efficient R code.
4.Importing data, time series data and Google Trends.
5.A data science project – a first look.
6.Data manipulation with tidyr.
7.Plotting with ggplot2.
8.R style guide, git integration, projects in R Studio.
9.Maps, plots and Amazon's choice for a next office building.
10.A data science project – more advanced.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

MINING LAB (B56PC8)

B.Tech VI Semester **L/T/P C**
0/0/2 1

Course Objectives:

Learn to perform data mining tasks using a data mining toolkit (such as open source WEKA), Understand the data sets and data preprocessing, demonstrate the working of algorithms for data mining tasks .

Course Outcome:

Ability to understand the various kinds of tools.

1. Analyze data from files and other sources. L4
2. Apply various data manipulation tasks on various datasets. L3
3. Apply data mining techniques on real time data sets. L3

TASK – 1

Build Data Warehouse and Explore WEKA

1. Build a Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration tool, Pentoaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.).

1) Identify source tables and populate sample data

2) Design multi-dimensional data models namely Star, snowflake and Fact constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, Manufacturing, Automobile, etc.).

3) Write ETL scripts and implement using data warehouse tools

4) Perform various OLAP operations such slice, dice, roll up, drill up and pivot

5) Explore visualization features of the tool for analysis like identifying trends etc.

2. Explore WEKA Data Mining/Machine Learning Toolkit

1) Downloading and/or installation of WEKA data mining toolkit,

2) Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.

3) Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel) (iv). Study the arff file format

4) Explore the available data sets in WEKA.

5) Load a data set (ex. Weather dataset, Iris dataset, etc.)

6) Load each dataset and observe the following:

List the attribute names and they types

1. Number of records in each dataset
2. Identify the class attribute (if any)
3. Plot Histogram
4. Determine the number of records for each class.

5. Visualize the data in various dimensions

TASK – 2

Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets

1. Explore various options available in Weka for preprocessing data and apply (like Discretization Filters, Resample filter, etc.) on each dataset
2. Load each dataset into Weka and run Aprori algorithm with different support and confidence values. Study the rules generated.
3. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated. Derive interesting insights andobserve the effect of discretization in the rule generation process.

TASK – 3

Demonstrate performing classification on data sets

1. Load each dataset into Weka and run Id3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.
2. Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix and derive Accuracy, F-measure, TPrate, FPrate, Precision and Recall values. Apply cross-validation strategy with various fold levels and compare the accuracy results.
3. Load each dataset into Weka and perform Naïve-bayes classification and k- Nearest Neighbour classification. Interpret the results obtained.
4. Plot RoC Curves
5. Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for eachdataset, and deduce which classifier is performing best and poor for each dataset and justify.
6. Load each dataset into Weka and perform Naïve-bayes classification and k- Nearest Neighbour classification. Interpret the results obtained.
7. Plot RoC Curves
8. Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for eachdataset, and deduce which classifier is performing best and poor for each dataset and justify.

TASK – 4

Demonstrate performing clustering on data sets

1. Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters). Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
2. Explore other clustering techniques available in Weka.
3. Explore visualization features of Weka to visualize the clusters. Derive interesting insights and explain.

Task – 5

Demonstrate performing Regression on data sets

1. Load each dataset into Weka and build Linear Regression model. Study the clusters formed. Use Training set option. Interpret the regression model and derive patterns and conclusions from the regression results.
2. Use options cross-validation and percentage split and repeat running the Linear Regression Model. Observe the results and derive meaningful results.
3. Explore Simple linear regression technique that only looks at one variable.

Text Books:

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

Resource Sites:

1.http://www.pentaho.com/
2.http://www.cs.waikato.ac.nz/ml/weka/

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

Data Mining Lab

Objectives:

Learn practical experience using data mining techniques on real world data sets. Emphasize hands-on experience working with all real data sets.

Course Outcomes:

1. Understand and Analyze data from files and other sources. L4
2. Apply various data manipulation tasks on various datasets. L3
3. Apply data mining techniques on real time data sets. L3

List of Sample Problems:

Task 1: Credit Risk Assessment Description:

The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the banks profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans couldlead to the collapse of the bank. The bank's loan policy must involve a compromise:not too strict, and not too lenient. To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

1. Knowledge Engineering. Find a loan officer who is willing to talk. Interview herand try to represent her knowledge in the form of production rules.
2. Books. Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text form to production rule form.
3. Common sense. Imagine yourself as a loan officer and make up reasonable ruleswhich can be used to judge the credit worthiness of a loan applicant.
4. Case histories. Find records of actual cases where competent loan officers correctlyjudged when, and when not to, approve a loan application.

The German Credit Data:

Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. Credit dataset (original) Excel spreadsheet version of the German credit data. In spite of the fact that the data is German, you should probably make use of it for this assignment. (Unless you really can consult a real loan officer!)

A few notes on the German dataset:

1. DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian (but

looks and acts like a quarter).

2. owns_telephone. German phone rates are much higher than in Canada so fewer people own telephones.
3. foreign_worker. There are millions of these in Germany (many from Turrkey). It is very hard to get German citizenship if you were not born of German parents. • There are 20 attributes used in judging a loan applicant. The goal is the classify the applicant into one of two categories, good or bad.

Subtasks: (Turn in your answers to the following tasks)

1. List all the categorical (or nominal) attributes and the real-valuedattributesseperately. (5 marks)
2. What attributes do you think might be crucial in making the credit assessment? Come up with some simple rules in plain English using your selected attributes.(5 marks)
3. One type of model that you can create is a Decision Tree - train a Decision Treeusing the complete dataset as the training data. Report the model obtained after training. (10 marks)
4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly? (This is also called testing on the training set) Why doyou think you cannot get 100
   % training accuracy? (10 marks)
5. Is testing on the training set as you did above a good idea? Why or Why not ? (10 marks)
6. One approach for solving the problem encountered in the previous question is using cross-validation? Describe what is cross-validation briefly. Train a Decision Tree again using cross- validation and report your results. Does your accuracy increase/decrease? Why? (10 marks)
7. Check to see if the data shows a bias against “foreign workers” (attribute 20), or “personal-status” (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka’s GUI Explorer. Did removing these attributes have any significant effect?Discuss. (10 marks)
8. Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Rememberto reload the arff data file to get all the attributes initially before you start selecting the ones you want.) (10 marks)
9. Sometimes, the cost of rejecting an applicant who actually has a good credit(case 1) might be higher than accepting an applicant who has bad credit (case 2).

   Instead of counting the misclassifications equally in both cases, give a highercost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)? (10 marks)
10. Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees? How does the complexity of a Decision Tree relate to the bias of

the model? (10 marks)

11. You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning - Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain? Also, report your accuracy using the pruned model. Does your accuracy increase? (10 marks)

12. (Extra Credit): How can you convert a Decision Trees into "if-thenelse rules". Make up your own small Decision Tree consisting of 2-3 levels and convert it into a set of rules. There also exist different classifiers that output the model in the form of rules - one such classifier in Weka is rules. PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one Can you predict what attribute that might be in this dataset ? OneR classifier uses a single attribute to make decisions (it chooses the attribute based on minimum error). Report the rule obtained by training a one R classifier. Rank the performance of j48, PART and one R. (10 marks)

Task Resources:

13. Mentor lecture on Decision Trees
14. Andrew Moore's Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation) Decision Trees (Source: Tan, MSU)
15. Tom Mitchell's book slides (See slides on Concept Learning and Decision Trees)

Weka resources:

16. Introduction to Weka (html version) (download ppt version) Download Weka Weka Tutorial ARFF format
17. Using Weka from command line

Task 2: Hospital Management System

18. Data Warehouse consists Dimension Table and Fact Table. REMEMBER Thefollowing Dimension
19. The dimension object (Dimension):
20. _ Name
21. _Attributes (Levels) , with one primary key
22. _Hierarchies
23. One time dimension is must. About Levels and Hierarchies
24. Dimension objects (dimension) consist of a set of levels and a set of hierarchies defined over those levels. The levels represent levels of aggregation. Hierarchies describe.
25. parent-child relationships among a set of levels. For example, a typical calendardimension could contain five levels. Two hierarchies can be defined on these levels: H1: YearL > QuarterL > MonthL > WeekL > DayL H2: YearL > WeekL > DayL
26. The hierarchies are described from parent to child, so that Year is the parent of Quarter, Quarter the parent of Month, and so forth. About Unique Key Constraints When you create a definition for a hierarchy, Warehouse Builder creates an identifier key for eachlevel of the hierarchy and a unique key constraint on the lowest level (Base Level) Design a Hospital Management system data warehouse (TARGET) consists of Dimensions Patient, Medicine, Supplier, Time. Where measures are 'NO UNITS', UNIT PRICE. Assume the Relational database (SOURCE) table schemas as follows

TIME (day, month, year),

27. PATIENT (patient_name, Age, Address, etc.,)
28. MEDICINE (Medicine_Brand_name, Drug_name, Supplier, no_units, Uinit_Price,etc.,) SUPPLIER
29. :(Supplier_name, Medicine_Brand_name, Address, etc.,) If each Dimension has 6 levels, decide the levels and hierarchies, Assume the level names suitably. Design the Hospital Management system data warehouse using all schemas. Give the example 4-Dcube with assumption names.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

OBJECT ORIENTED ANALYSIS AND DESIGN LAB(B56PC9)

B.Tech VI Semester **L/T/P C**
0/0/2 1

Course Objective:

Write programs for solving real world problems using java language.

Course Outcomes:

After learning the contents of this course, the student must be able to:

1. Understand the functioning of the Rational Rose or Umbrello software. L4
2. Design the UML diagrams for the E- Ticketing System. L6
3. Develop the various UML diagrams for any real time applications. L6

List Of Experiments:

Consider the following three case studies:
Online course reservation system E-ticketing Library Management System

Week 1

Familiarization with Rational Rose or Umbrello For each case studey develop a problem statement.

Week 2, 3 & 4:

For each case study:
1) Identify and analyze events
2) Identify Use cases
3) Develop event table
4) Identify & analyze domain classes
5) Represent use cases and a domain class diagram using Rational Rose
6) Develop CRUD matrix to represent relationships between use cases and problem domain classes

Week 5 & 6:

For each case study:
1) Develop Use case diagrams
2) Develop elaborate Use case descriptions & scenarios
3) Develop prototypes (without functionality)
4) Develop system sequence diagrams

Week 7, 8, 9 & 10:

For each case study:
1) Develop high-level sequence diagrams for each use case
2) Identify MVC classes / objects for each use case

3) Develop Detailed Sequence Diagrams / Communication diagrams for each use case

showing interactions among all the three-layer objects
4) Develop detailed design class model (use GRASP patterns for responsibilityassignment)
5) Develop three-layer package diagrams for each case study

Week 11 & 12:
For each case study:
1. Develop Use case Packages
2. Develop component diagrams
3. Identify relationships between use cases and represent them
4. Refine domain class model by showing all the associations among classes

Week 13 onwards:
For each case study:
1) Develop sample diagrams for other UML diagrams – state chart diagrams,activity diagrams and deployment diagrams.

TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(Autonomous)

B.TECH. COMPUTER SCIENCE & ENGINEERINGCOURSE STRUCTURE -R18

PROCESSORS LAB (B56PC10)

B.Tech VI Semester **L/T/P C**
0/0/2 1

Course Objective:
To provide an understanding of the language translation peculiarities by designing a complete translator for a mini language.

Course Outcome:

1. Analyze the practical approach of how a compiler works. L4
2. Develop lex analyzer and parsers for the given language. L3
3. Construct LEX programs to solve various problems. L3

Recommended System / Software Requirements:
1. Intel based desktop PC with minimum of 166 MHZ or faster processor with atleast 64 MBRAM and 100 MB free disk space
2. C++ comiler and JDK kit

Consider the following mini Language, a simple procedural high-level language, only operating on integer data, with a syntax looking vaguely like a simple C crossed with Pascal. The syntax of the language is defined by the following BNFgrammar:

<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> ::= <bexpression> then <slist> else <slist> endif | if <bexpression> then
<slist> endif
<whilestatement> ::= while <bexpression> do <slist> enddo
<printstatement> ::= print (<expression>)
<expression> ::= <expression> <additionop> <term> | <term> | addingop> <term>
<bexpression> ::= <expression> <relop> <expression>
<relop> ::= < | <= | == | >= | > | !=
<addingop> ::= + | -
<term> ::= <term> <mulitop> <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] Notealso 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
}
```

1. Design a Lexical analyzer for the above language. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value.
2. Implement the lexical analyzer using JLex, flex or lex or other lexical analyzer generating tools.
3. Design Predictive parser for the given language.
4. Design LALR bottom up parser for the above language.
5. Convert the BNF rules into Yacc from and write code to generate abstract syntaxtree.
6. Write program to generate machine code from the abstract syntax tree generated by the parser. The following instruction set may be consideredas target code.

The following is a simple register-based machine, supporting a total of 17 instructions. It has three distinct internal storage areas. The first is the set of 8 registers, used by the individual instructions as detailed below, the second is an area used for the storage of variables and the third is an area used for the storage of program. The instructions can be precede by a label. This consists of an integer in the range 1 to 9999 and the label is

followed by a colon to seperate it from the rest of the instruction. The numerical labelcan be used as the argument to a jump instruction, as detailed below.

In the description of the individual instructions below, instruction argument types arespecified as follows:
R specifies a register in the form R0, R1, R2, R3, R4, R5, R6 or R7 (orr0, r1, etc). L specifies a numerical label (in the rabge 1 tp 9999).
V specifies a "variable location" (a variable number, or a variable location pointed to by a register - see below).
A specifies a constant value, a variable location, a register or a variable location pointed to by a register (an indirect address). Constant values are specified as an integer value, optionally preceded by a minus sign, preceded by a # symbol. An indirect address is specified by an @ followed by a register.
So, for example an A-type argument could have the form 4 (variable number 4), #4 (the constant value 4), r4 (register 4) or @r4 (the contents of register 4 identifies the variable location to be accessed).

The instruction set is defined as follows:
LOAD A, R
loads the integer value secified by A into register
R. STORE R, V
stores the value in register R tovariable V. OUT R
outputs the value in register R. NEG R negates the value in register R. ADD A, R
adds the value specified by A to register R, leaving the result inregister R. SUB A, R
subtracts the value specified by A from register R, leaving the result inregister R. MUL A, R
multiplies the value specified by A by register R, leaving the result inregister R. DIV A, R
divides register R by the value specified by A, leaving the result inregister R. JMP L
causes an unconditional jump to the instruction with thelabel L. JEQ R, L
The declaration

int a[3] declares an array of three elements, referenced as a[0], a[1] and a[2] Notealso 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
}
```

7. Design a Lexical analyzer for the above language. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value.
8. Implement the lexical analyzer using JLex, flex or lex or other lexical analyzer generating tools.
9. Design Predictive parser for the given language.
10. Design LALR bottom up parser for the above language.

11. Convert the BNF rules into Yacc from and write code to generate abstract syntaxtree.
12. Write program to generate machine code from the abstract syntax tree generated by the parser. The following instruction set may be consideredas target code.

The following is a simple register-based machine, supporting a total of 17 instructions. It has three distinct internal storage areas. The first is the set of 8 registers, used by the individual instructions as detailed below, the second is an area used for the storage of variables and the third is an area used for the storage of program. The instructions can be precede by a label. This consists of an integer in the range 1 to 9999 and the label is
followed by a colon to seperate it from the rest of the instruction. The numerical label can be used as the argument to a jump instruction, as detailed below. jumps to the instruction with the label L if the value in register Ris zero. JNE R, L

jumps to the instruction with the label L if the value in register R isnot zero. JGE R, L
jumps to the instruction with the label L if the value in register R is greater than or equal to zero. JGT R, L
jumps to the instruction with the label L if the value in register R is greaterthanzero. JLE R, L
jumps to the instruction with the label L if the value in register R is less than orequal to zero. JLT R, L

jumos to the instruction with the label L if the value in register R is lessthan zero. NOP
is an instruction with no effect. It can be tagged bya label. STOP
stops execution of the machine. All programs should terminate by executing a STOP instruction.

1. Write a LEX program to count the number of lines, words, blank spaces and characters in a given input
2. Write a LEX program to find the number of vowels, consonants in the given input.
3. Write a LEX program to check if a given number is positive or negative.
4. Write a LEX program to check if a given statement is simple or compound.
5. Write a LEX program to count the type of numbers.
6. Write a LEX program to check the validity of a given arithmetic statement.
7. Write a lex program to count the number of “printf” and “scanf” statements in avalid c program and replace them with write and read statements respectively.
8. Write a lex program to count the number of characters, words, blanks andlines in a given strings.

   jumos to the instruction with the label L if the value in register R is lessthan zero. NOP is an instruction with no effect. It can be tagged bya label. STOP

   stops execution of the machine. All programs should terminate by executing a STOP instructio

9.Write a LEX program to count the number of lines, words, blank spaces and characters in a given input

9. Write a LEX program to find the number of vowels, consonants in the given input.
   Write a LEX program to check if a given number is positive or negative.
10. Write a LEX program to check if a given statement is simple or compound.
11. Write a LEX program to count the type of numbers.
12. Write a LEX program to check the validity of a given arithmetic statement.

13. Write a lex program to count the number of “printf” and “scanf” statements in avalid c program and replace them with write and read statements respectively.
14. Write a lex program to count the number of characters, words, blanks and lines in a given strings.
15. jumos to the instruction with the label L if the value in register R is lessthan zero.
    NOP
    is an instruction with no effect. It can be tagged by a
    label. STOP
    stops execution of the machine. All programs should terminate by executing a
    STOPinstruction.
16. Write a LEX program to count the number of lines, words, blank spaces and characters in a given input. Write a LEX program to find the number of vowels, consonants in the given input.
17. Write a LEX program to check if a given number is positive or negative.
18. Write a LEX program to check if a given statement is simple or compound.
19. Write a LEX program to count the type of numbers.
20. Write a LEX program to check the validity of a given arithmetic statement.
21. Write a lex program to count the number of “printf” and “scanf” statements in a valid c program and replace them with write and read statements respectively.
22. Write a lex program to count the number of characters, words, blanks and lines in a given strings.