Regulation: GNITC – R21

COURSE STRUCTURE
&
DETAILED SYLLABUS

for

II Year -I& II Sem
B.Tech. Degree Course

(Applicable for the batch admitted from 2021-22)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING(ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS
(An Autonomous Institution)
Campus: Ibrahimpatnam, R.R. District - 501506 (T.S).

GNITC is now NAAC A+ Accredited Institute

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS

(An UGC Autonomous Institution – Affiliated to JNTUH)
Ibrahimpatnam, Ranga Reddy (District) - 501 506.

College Code : WJ

EAMCET CODE : GURU

ACADEMIC REGULATIONS
GNITC – R21

for
UG- B.Tech. Programme
(Applicable for the batches admitted from 2021-2022)

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (AUTONOMOUS)
SCHOOL OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
(Applicable for the batch admitted from 2021-2022)

IV SEMESTER (SECOND YEAR)

S.	Subject Code	Grou	Subject	L	T	P	Credits
1	21BS0MA06	BSC	Mathematical and Statistical Foundations	3	0	0	3
2		PCC	Theory of Computation	3	0	0	3
3		PCC	Object Oriented Programming using Java	3	0	0	3
4		PCC	Relational Data Base Management Systems	3	0	0	3
5		PCC	Fundamentals of Operating Systems	3	0	0	3
6		PCC	Object Oriented Programming using Java Lab	0	0	3	1.5
7		PCC	Relational Data Base Management Systems Lab	0	0	3	1.5
8		PCC	Fundamentals of Operating Systems Lab	0	0	3	1.5
9		PCC	R Programming Lab	0	0	3	1.5
5 Theory + 4 Lab			**Total Credits**	**17**	**00**	**12**	**21**

L – Lecture **T** – Tutorial **P** – Practical

II Year B.Tech. Sem-II

L	T	P	C
3	0	0	3

THEORY OF COMPUTATION (21PC0AD07)

PRE-REQUISITE:

1. A course on "Discrete Mathematics"

COURSE OBJECTIVE:

This course is intended to comprehend the theoretical foundations of computer science from the perspective of formal languages and classify machines by their power to recognize languages

SYLLABUS:

UNIT - I

Introduction to Finite Automata:Structural Representations, the Central Concepts of Automata Theory – Alphabet, Strings, Languages, Problems, Applications of finite automata. Deterministic Finite Automata: Definition, How a DFA Process strings, The language of DFA - Designing DFAs, Non-deterministic Finite Automata:Formal Definition, How a NFA Process strings, The language of NFA - Designing NFAs,Conversion of NFA to DFA, Equivalence and Minimization of Automata.Finite Automata with Epsilon-Transitions, Conversion of NFA with €-transitions to NFA without €-transitions.

UNIT - II

Finite Automata with output - Moore and Mealy machines, inter-conversions, designing moore and mealy machines.

Regular Expressions and Regular Grammars: Recursive definition of Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions. Pumping Lemma for regular languages, Closure Properties, Decision Properties, Grammar Formalism: Regular Grammars, Inter-conversions: Finite automata and Regular Grammar.

UNIT – III

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

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

UNIT – IV

Minimization of CFG: Eliminating useless symbols and productions, Eliminating€-Productions, Eliminating Unit productions. Normal Forms for Context- Free Grammars: Chomsky Normal form, Griebach Normal form, Conversion of CFG to CNF, Conversion of CFG to GNF. Pumping Lemma for CFLs, Closure Properties of CFLs, Decision Properties of CFLs.

UNIT – V

Turing Machines: Introduction to Turing Machine,Formal Description, Instantaneous description, The language of a Turing machine,Types of Turing machine, universal turing machine, counter machine, church's hypothesis, halting problem, undecidability, Recursive Enumerable Languages, Recursive languages, Properties of recursive enumerable and recursive languages, Posts Correspondence Problem, Modified Post Correspondence problem.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

COURSE OUTCOMES:

By the end of the course, students will be able to:

CO 1: Design the finite automata for the recognized languages
CO 2: Write regular expressions for the languages
CO 3: Demonstrate context-free grammars/languages, derivations and parse trees
CO 4: Design push down automata and turing machines for the languages
CO 5: Comprehend language classes & grammars, relationship among them with the help of Chomsky hierarchy

II Year B.Tech. Sem-II

L	T	P	C
3	0	0	3

OBJECT ORIENTED PROGRAMMING USING JAVA (21PC0AD08)

PRE-REQUISITE:

1. A course on "Programming for problem solving"

COURSE OBJECTIVE:

The aim of this course is to comprehend objectoriented programming concepts and apply them in problem solving

SYLLABUS:

UNIT – I

OOP concepts: Classes and objects, data abstraction, encapsulation, inheritance, benefits of inheritance, polymorphism, procedural and object oriented programming paradigm; Java programming: History of java, comments data types, variables, constants, scope and life time of variables, operators, operator hierarchy, expressions, type conversion and casting, enumerated types, control flow statements, jump statements, simple java stand alone programs, arrays, console input and output, formatting output, constructors, methods, parameter passing, static fields and methods, access control, this reference, overloading methods and constructors, recursion, garbage collection, exploring string class.

UNIT – II

Inheritance: Inheritance hierarchies, super and subclasses, member access rules, super keyword, preventing inheritance: final classes and methods, the object class and its methods; Polymorphism: Dynamic binding, method overriding, abstract classes and methods; Interface: Interfaces vs Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface; Packages: Defining, creating and accessing a package, understanding CLASSPATH, importing packages.

UNIT – III

Exception Handling: Benefits of exception handling, the classification of exceptions, exception hierarchy, checked and unchecked exceptions, usage of try, catch, throw, throws and finally, re-throwing exceptions, exception specification, built in exceptions, creating own exception sub classes. Multithreading: Differences between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, inter thread communication.

UNIT – IV

Files: Streams, byte streams, character stream, text input/output, binary input/output, random access file operations, file management using file class; Connecting to Database: Connecting to a database, querying a database and processing the results, updating data with JDBC.

UNIT – V

GUI programming with Java: The AWT class hierarchy, introduction to swing, swing Vs AWT, hierarchy for swing components, containers, JFrame, JApplet, JDialog, JPanel; Overview of some swing components: JButton, JLabel, JTextField, JTextArea, simple applications; Layout management: Layout manager types: Border, grid and flow; Applets: Inheritance hierarchy for applets, differences between applets and applications, life cycle of an applet, passing parameters to applets.

TEXT BOOKS:

1. Herbert Schildt, Dale Skrien, "Java Fundamentals: A Comprehensive Introduction", McGraw Hill, 1st Edition, 2013.
2. Herbert Schildt, "Java the Complete Reference", McGraw Hill, Osborne, 8th Edition, 2011.
3. T. Budd, "Understanding Object Oriented Programming with Java", Pearson Education, Updated Edition (New Java 2 Coverage), 1999.

REFERENCE BOOKS:

1. P.J. Deitel, H. M. Deitel, "Java: How to Program", Prentice Hall, 6th Edition, 2005.
2. P. Radha Krishna, "Object Oriented Programming through Java", Universities Press, CRC Press, 2007.
3. Bruce Eckel, "Thinking in Java", Prentice Hall, 4th Edition, 2006.
4. Sachin Malhotra, Saurabh Chaudhary, "Programming in Java", Oxford University Press, 2nd Edition, 2014.

COURSE OUTCOMES:

By the end of the course, students will be able to

CO 1: describe the concepts of OOP and basics of java programming
CO 2: express the programming skills in problem solving
CO 3: solve the exceptions and handle the exceptions in programming
CO 4: outline the GUI based applications
CO 5: extend their knowledge of java programming further on their own

II Year B.Tech. Sem-II

L	T	PC
3	0	0 3

RELATIONAL DATA BASE MANAGEMENT SYSTEMS(21PC0AD09)

PRE-REQUISITE:

1. A course on "Data Structures"

COURSE OBJECTIVE:

This course is introduced to describe the basic concepts of SQL, build queries using SQL commands and generate applications of database systems

SYLLABUS:

UNITI:

Introduction, Database-System Applications, Advantages of a DBMS, File Systems versus a DBMS, View of Data, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval, Database Users and Administrators.

Database Design and ER Diagrams, Beyond ER Design, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT II:

Introduction to the Relational Model: integrity constraint over relations, enforcing integrity constraints, querying relational data, logical data base design, introduction to views, destroying/altering tables and views, form of basic SQL query, introduction to nested queries, correlated nested queries, set comparison operators, aggregate operators, NULL values, comparison using null values, logical connectivity's, AND, OR and NOT, outer joins, disallowing NULL values, complex integrity constraints in SQL, triggers and active data bases.

UNIT III:

Relational Algebra: Selection and projection, set operations, renaming, Joins, Division, Examples of Algebra overviews, Relational calculus.

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

UNIT IV:

Transaction Concept, A Simple Transaction Model, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels

Lock-Based Protocols, Deadlock Handling, Multiple Granularity, Timestamp-Based Protocols, The ACID Properties, Transactions and Schedules, Concurrent Execution

of Transactions, Lock-Based Concurrency Control,
Recovery and Atomicity, Log–Based Recovery, Buffer Management, Failure with loss of nonvolatile storage, ARIES, Remote Backup systems.

UNIT V:

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

TEXT BOOKS:

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill 3rdEdition
2. Database System Concepts, Silberschatz, Korth, Mc Graw hill, Vedition.

REFERENCE BOOKS:

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

COURSE OUTCOMES:

By the end of the course, students will be able to:

CO 1: Describe the basic fundamentals of DBMS, database design and normal forms
CO 2: Identify the appropriate SQL commands for retrieval and management of data
CO 3: Analyze the schema refinement and normal forms
CO 4: Identify data models for relevant problems
CO 5: Model database storage structures and access techniques

II Year B.Tech. Sem-II **L T PC**
3 0 0 3

FUNDAMENTALS OF OPERATING SYSTEMS(21PC0AD10)

PRE-REQUISITES:
1. A course on "Programming for problem solving"

2. A course on "Computer Organization & Architecture"

COURSE OBJECTIVE:
The purpose of this course is to realize the concepts of input- output, storage and file management in Unix/Linux

SYLLABUS:

UNIT – I
Introduction: What OperatingSystems do, Computer System Organization,Storage management, Single Processor Systems, Multiprocessor systems,ClusteredSystems,DistributedSystems,Special Purpose Systems, OS Services, System Calls, Structure of an OS - Simple,Layered Microkernel Operating Systems, Virtual Machines.

UNIT – II
Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR

UNIT – III
Inter-process Communication: Race Condition, The Critical Section problem, Peterson's Solution, Synchronization hardware, Semaphores, Monitors, Classical IPC Problems: Bounded Buffer Problem, Reader's& Writer Problem, Dinning Philosopher Problemetc.
Deadlocks: Definition, Necessary and sufficient conditions for Deadlock,Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection andRecovery.

UNIT – IV
Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memoryallocation– Fixed and variable partition– Internal and External fragmentation and Compaction;

Paging:BasicConcept,Hardware support for paging, Protection and sharing, Segmentation.

Virtual Memory Management: Basics of Virtual Memory, Demand paging, Page fault,Handling of Page Fault,Page Replacement Techniques: Optimal, First in First Out (FIFO),and Least Recently used(LRU).

UNIT – V

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table).

Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks

TEXT BOOKS:

1. Operating System Concepts Essentials, 9th Edition by AviSilberschatz, Peter Galvin, Greg Gagne, Wiley Asia StudentEdition.
2. Operating System Concepts Essentials, 8th Edition by AviSilberschatz, Peter Galvin, Greg Gagne, Wiley Asia StudentEdition.

REFERENCEBOOKS:

1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt,Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall ofIndia
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly andAssociates

COURSE OUTCOMES:

By the end of the course, students will be able to:

CO 1: Describe the synchronous and asynchronous communication mechanisms in their respective operating systems

CO 2: Discuss the inter process communication in Unix/Linux

CO 3: Apply optimization techniques for the improvement of system performance

CO 4: Analyze turnaround time, waiting time, response time and throughput

CO 5: Compare the different operating system functionalities

II Year B.Tech. Sem-II **L T PC**
0 0 4 2

OBJECT ORIENTED PROGRAMMING USING JAVA LAB(21PC0AD11)

CO-REQUISITE:
1. A course on "Object Oriented Programming"

COURSE OBJECTIVE:
This lab course is introduced to create the Graphical User Interface using Applets, AWT Components & Swing Components

SOFTWARE REQUIREMENTS:
Java / Eclipse / Netbeans

SYLLABUS:

LIST OF PROGRAMS

1. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box. [Use JOption Pane –Input dialog, Message dialog]
2. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
3. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
4. Write a Java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.
5. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown.

6. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.Handle any possible exceptions like divide by zero
7. a) Develop an applet in Java that displays a simple message.

b)Develop an applet in Java that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named "Compute" is clicked.

8. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a Java program to display the table using Labels in Grid Layout.
9. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
10. Implement the above program with the database instead of a text file.
11. Write a Java program that prints the meta-data of a given table.

TEXT BOOK:

5. Java Fundamentals – A comprehensive Introduction, Herbert Schildt and Dale Skrien, TMH.

REFERENCE BOOKS:

1. Java for Programmers, P.J.Deitel and H.M.Deitel, Pearson education (OR) Java: How to Program, P.J.Deitel and H.M.Deitel, PHI.
2. Object Oriented Programming through Java, P. Radha Krishna, University Press.

COURSE OUTCOMES:

Upon successful completion of this Lab, students will be able to:

CO 1: Work with java compiler and eclipse platform to write and execute java programs

CO 2: Apply object oriented features in java programming for problem solving

CO 3: Access data from database with java programs

CO 4: Describe exception handling mechanism

CO 5: Develop applications using Console I/O and File I/O,GUI applications

II Year B.Tech. Sem-II

L	T	P	C
0	0	4	2

Relational Data Base Management Systems LAB (21PC0AD12)

CO-REQUISITE:
1. A course on "Data Base Management Systems"

COURSE OBJECTIVE:
This lab course is intended to describe the SQL basics for data definition, data manipulation and introduce ER data model, database design and normalization

SOFTWARE REQUIREMENTS:
Oracle / MySql

SYLLABUS:
LIST OF TASKS
1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

TEXT BOOKS:
1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill, 3rd Edition
2. Database System Concepts, Silberschatz, Korth, Mc Graw hill, 5th Edition.

REFERENCE BOOKS:

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

COURSE OUTCOMES:

Upon successful completion of this Lab, students will be able to:

CO 1: design database schema for a given application
CO 2: apply normalization
CO 3: acquire skills in using SQL commands for data definition and data manipulation
CO 4: develop solutions for database applications using procedures
CO 5: employ cursors and triggers

II Year B.Tech. Sem-II **L T PC**
0 0 31.5

FUNDAMENTALS OF OPERATING SYSTEMSLAB (21PC0AD13)

CO-REQUISITE:
1. A course on "Operating Systems"

COURSE OBJECTIVE:
This lab course is intended to perform different functionalities in Operating system Unix/Linux using commands

SOFTWARE REQUIREMENTS:
Turbo C / Unix / Linux

SYLLABUS:

LIST OF PROGRAMS

1. Write a program to implement following process scheduling algorithms for a given
specification of CPU utilization, Throughput, Turnaround Time, Waiting Time,ResponseTime.
    i. First Come First Serve
    ii. Shortest Job First
    iii. Priority
    iv. Round Robin
2. Write a program to simulate the following memory management techniques:
    i. Variable Memory technique
    ii. Fixed Memory Technique
3. Write a program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
4. Write a program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
5. Write a program to simulate the following memory management techniques:
    i. Paging
    ii. Segmentation
6. Write a programs to simulate the following file organization Techniques:
    i. Single level
    ii. Two level

iii. Hierarchical
iv. DAG

7.Write a programs to simulate the following file allocation strategies:
i. Sequential
ii. Linked
iii. Indexed

8 .Write a programs to simulate the following Page Replacement Techniques:
i. FIFO
ii. LRU
iii. Optimal

9. Write a program to simulate disk scheduling algorithms.

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley
2. Advanced programming in the Unix environment, W.R.Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems – Internals and Design Principles Stallings, Fifth Edition–2005, Pearson Education/PHI
2. Operating System A Design Approach-Crowley,TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition, Pearson/PHI
4. Unix programming environment, Kernighan and Pike, PHI. / Pearson Education
5. Unix Internals The New Frontiers, U.Vahalia, Pearson Education

COURSE OUTCOMES:

Upon successful completion of this Lab, students will be able to:

CO 1: Describe the operating systems concepts in Unix/Linux

CO 2: Illustrate various concepts in operating systems through implementation

CO 3: Solve the real-time problems like deadlock by providing suitable solutions

CO 4: Analyze the different operating system functionalities

CO 5: Simulate the page replacement algorithms

II Year B.Tech. Sem-II

L	T	PC	
0	0	2	1

R PROGRAMMING LAB(21PC0AD14)

COURSE OBJECTIVE:
This lab course is intended to learn syntax and semantics and build web services and introduction to network and database programming in R programming.
SOFTWARE REQUIREMENTS:R Studio
SYLLABUS:
LIST OF PROGRAMS
Overview of R, R data types and objects, reading and writing data
1. Write an R-Program to print Hello World
2. Write an R-Program to take input from user.
3. Write an R-Program to demonstrate working with operators (Arithmetic, Relational, Logical, Assignment operators).

Control structures, functions, scoping rules, dates and times…
4. Write an R Program to Check if a Number is Odd or Even
5. Write an R Program to check if the given Number is a Prime Number
6. Write an R Program to Find the Factorial of a Number

Control structures, functions, scoping rules, dates and times
7. Write an R Program to Find the Fibonacci sequence Using Recursive Function
8. Write an R Program to Make a Simple Calculator
9. Write a R program to create the system's idea of the current date with and without time

Vectors and working on data sets (Reading and Writing different Types of Datasets (.txt, .csv))
10. Write an R Program to create a Vector and to access elements in a Vector
11. Write a R program, Reading data from files and working with datasets
(i) Reading data from csv files, inspection of data. (ii) Reading data from Excel files.

Matrices, Arrays and Lists
12. Write an R Program to create a Matrix and access rows and columns using functions colnames() and rownames() .
13. Write an R Program to create a Matrix using cbind() and rbind() functions.
14. Write an R Program to create a Matrix from a Vector using dim() function.

15. Write an R Program to create a List and modify its components.

Data Frames

16. Write an R Program to create a Data Frame.
17. Write an R Program to access a Data Frame like a List.
18. Write an R Program to access a Data Frame like a Matrix.
19. Write an R Program to create a Factor.
20. Write an R Program to Access and Modify Components of a Factor.

Data Visualization

21. Write a R program to implement Graphs
(i) Basic high-level plots
(ii) Modifications of scatter plots
(iii) Modifications of histograms, parallel box plots.

OOP- S3 Classes – S4 Classes and Networking, web services and data base application

22. Write an R Program to create an S3 Class and S3 Objects.
23. Write an R Program to write a own generic function in S3 Class.
24. Write an R Program to create an S4 Class and S4 Objects.
25. Write an R Program to write a own generic function in S4 Class.
26. Write an R Program to create Reference Class and modify its Methods.
27. Wrie an R program to create simple web application.

TEXT BOOKS:

1. Norman Matloff , "The Art of R Programming: A Tour of Statistical Software Design", No Starch Press, 2011
2 Jared P. Lander, "R for Everyone: Advanced Analytics and Graphics", Addison-Wesley Data & Analytics Series, 2013.

REFERENCE BOOKS:

1. Mark Gardener, " Beginning R – The Statistical Programming Language", Wiley, 2013
2. Robert Knell, "Introductory R: A Beginner's Guide to Data Visualisation, Statistical Analysis and Programming in R", Amazon Digital South Asia Services Inc, 2013.

COURSE OUTCOMES:

Upon successful completion of this Lab, students will be able to:

CO 1: Examine R syntax and semantics and be fluent in the use of R flow control and functions.

CO 2: Demonstrate proficiency in handling Strings and File Systems.

CO 3: Create, run and manipulate R Programs using core data structures like Lists, Data frames, matrices and use Regular Expressions.

CO 4: Interpret the concepts of Object-Oriented Programming as used in R.

CO 5: Implement exemplary applications related to Network Programming, Web Services and Databases in R.