COURSE STRUCTURE(R19)
AND
DETAILED SYLLABUS
(II YEAR)

COMPUTER SCIENCE & ENGINEERING

For
B.Tech., Four Year Degree Course
(Applicable for the batches admitted from 2019-20)

LENDI INSTITUTE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institution
Approved by AICTE & Permanently Affiliated to JNTUK, Kakinada
Accredited by NAAC with “A” Grade and NBA (CSE,ECE, EEE & ME)
Jonnada (Village), Denkada (Mandal), Vizianagaram Dist – 535 005
Phone No. 08922-241111, 241112
E-Mail: lendi_2008@yahoo.com
Website: www.lendi.org

II Year – I Semester							
S. No.	**Course code**	**Course Title**	**Category**	**L**	**T**	**P**	**Credits**
1	R19BSH-MA2101	Discrete Mathematical Structures	BS	3	0	0	3
2	R19CSE-PC2101	Principles of Programming Languages	PC	3	0	0	3
3	R19CSS-PC2102	Python Programming	PC	3	0	0	3
4	R19CSE-PC2103	Data Structures through C++	PC	3	0	0	3
5	R19CSE-PC2104	Computer Organization & Architecture	PC	3	0	0	3
6	R19CSS-PC2103	Python Programming Lab	PC	0	0	3	1.5
7	R19CSE-PC2105	Data Structures through C++ Lab	PC	0	0	3	1.5
8	R19BSH-MC2101	Essence of Indian Traditional Knowledge	MC	0	0	2	0
9	R19BSH-MC2102	Employability Skills - I	MC	3	0	0	0
10	R19CSE-MC2101	MOOC S-1	MC	0	0	0	0
			Total	**18**	**0**	**8**	**18**

II Year – II Semester							
S. No.	**Course code**	**Course Title**	**Category**	**L**	**T**	**P**	**Credits**
1	R19BSH-MA2201	Probability & Statistics	BS	3	0	0	3
2	R19CSE-PC2201	Java Programming	PC	3	1	0	4
3	R19CSE-PC2202	Operating Systems	PC	3	0	0	3
4	R19CSE-PC2203	Database Management Systems	PC	3	0	0	3
5	R19CSE-PC2204	Formal Languages and Automata Theory	PC	3	0	0	3
6	R19CSE-PC2205	Java Programming Lab	PC	0	0	3	1.5
7	R19CSE-PC2206	Operating System Lab	PC	0	0	3	1.5
8	R19CSE-PC2207	Database Managements Systems Lab	PC	0	0	3	1.5
9	R19BSH-MC2201	Professional Ethics & Human Values	MC	3	0	0	0
10	R19HS-MC2210	Constitution of India	MC	2	0	0	0
11	R19CSE-SI2201	Summer Internships	SI	0	0	0	0
12	R19CSE-MC2201	MOOC S-2	MC	0	0	0	0
			Total	**21**	**1**	**9**	**20.5**

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19BSH-MA2101	**Discrete Mathematical Structures** (Mathematical Foundations of Computer Science) **(Common to CSE, CSSE & CSIT)**	3	0	0	3

Course Objectives:

- To develop logical thinking in the field of Computer Science and Engineering.
- To introduce basic concepts and various algorithms of graphs.
- To introduce basics of group theory.
- To familiarise the concepts of various trees (i.e) spanning trees, shortest spanning trees.
- To Familiarise closed form solution of linear recurrence relations by various methods.
- To design the logical circuits using Boolean expressions.

Course Outcomes:

At the end of the course, the student will be able to

1. Test the validity of an argument through enhanced logical capabilities. (L3)
2. Implement Shortest path algorithm for different graphs.(L3)
3. Construct minimal spanning tree using algorithms for the graphs.(L3)
4. Understand the algebraic structures and their properties. (L3)
5. Find a general solution of recurrence equation using suitable method and apply the different properties of lattice to simplify Boolean expressions.(L3)

Unit I: Mathematical logic

Propositional Calculus: Statements, negation, conjunction, disjunction, conditional and bi-conditional, well formed formulae, tautologies, equivalence of formulae, duality, tautological implications, principal disjunctive and conjunctive normal forms, inference calculus, and rules of inference.

Predicate Calculus: Predicative logic, free & bound variables, quantifiers, rules of inference.

Learning Outcomes:

At the end of this unit, the student will be able to

- evaluate the basic logic statements (including compound statements, implications, inverses, converses and contrapositives) using truth tables and the properties of logic. (L3)
- reduce the given proposition into normal forms using properties of equivalence and implications.(L3)
- express a logic statement in terms of predicates using quantifiers and logical connectives. (L2)
- apply rules of inferences to verify the consistence of data. (L3)

Unit II: Graph theory

Introduction to relations, Graph, finite and infinite graphs, incidence and degree, types of Graphs, isomorphism, Adjacency Matrix, sub graphs, walk, path and circuit, connected and disconnected graphs, components, Euler graphs, Euler’s theorem, Hamiltonian paths and circuits, Shortest-path algorithm (Dijkstra’s Algorithm), planar graphs.

Learning Outcomes:

At the end of this unit, the student will be able to

- identify different types graphs. (L2)
- represent a graph using an adjacency matrix.(L2)
- construct Euler and Hamiltonian paths.(L3)
- implement Dijkstra’s Algorithm for the given graph(L4)

Unit III: Trees and Directed graphs (Theorems without Proof)

Properties of trees, binary trees, spanning trees, minimal spanning trees, Kruskal's algorithm, Prim's algorithm, tree traversals, DFS, BFS, Binary search tree, directed graph, graph coloring, chromatic number for various graphs and trees.

Learning Outcomes:

At the end of this unit, the student will be able to

- construct the spanning tree and binary tree from the given graphs. (L3)
- build minimal spanning tree by using different algorithms. (L3)
- implement DFS, BFS algorithm for the given graph(L3)
- determine the chromatic number of a given graph/ tree. (L3)

Unit IV: Algebraic Structures:

Algebraic Structures (semi groups, monoids, group and abelian groups), Sub groups, homomorphism, isomorphism, coset, Lagrange's theorem, normal subgroups, permutation groups and cyclic groups.

Applications: Group Codes

Learning Outcomes:

At the end of this unit, the student will be able to

- identity the given algebraic structure is a group or not.(L2)
- understand the concepts of sub groups, normal subgroups, cosets, cyclic groups, permutation groups, homomorphism and isomorphism of groups . (L2)
- apply error detection and correction techniques in the design of computer systems using group codes(L3)

Unit V: Lattice Theory & Recurrence relations (10 hours)

Lattice Theory: POSET, Hasse diagram, Lattice, Distributive lattice, Modular Lattice, complement, De-morgans laws, Boolean algebra (definition only)

Recurrence relations: Recurrence relations, substitution method, solving homogeneous linear recurrence relations by characteristic roots method, non – homogeneous linear recurrence relations.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand the concept of Poset and Lattice(L2)
- apply the different properties of lattice to simplifying Boolean expressions(L3)
- formulate recurrence relations of the sequences.(L3)
- apply substitution method to solve recurrence relations.(L3)
- solve non-homogeneous linear recurrence relations by characteristic roots method.(L3)

Text Books:

1. J.P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill, 1997. (for Unit 1,4, 5 (Lattice theory))
2. Joe L. Mott, Abraham Kandel and T. P. Baker, Discrete Mathematics for computer scientists & Mathematicians, 2/e, Prentice Hall of India Ltd, 2012.(for Unit 2,3,5 (recurrence relations))

References:

1. Keneth. H. Rosen, Discrete Mathematics and its Applications, 6/e, Tata McGraw-Hill,2009.
2. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall of India, 2006.
3. Susanna S. Epp, Discrete Mathematics with Application, Brooks/Colt Cengage learning, fourth edition.
4. Richard Johnsonburg, Discrete mathematics, 7/e, Pearson Education, 2008.

5. T. Koshy, Discrete Mathematics with Applications, Elsevier, New York, (2004).
6. J. A. Bondy, U. S. R. Murty, Graph Theory, Springer-Verlag, New York, (2008).
7. K. D. Joshi, Foundations of Discrete Mathematics, New Age International, (1989).

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2101	Principles of Programming Languages	3	0	0	3

Course Objectives:

- To understand and describe syntax and semantics of programming languages
- Understand the significance and implementation of programming languages in a compiler or interpreter
- To implement programs in an Imperative, functional, logical and object-oriented programming languages
- Learning principles to design new programming languages
- Increase capacity to express programming concepts alternative ways

Course Outcomes:

1. Design and implement the concepts of data types, arrays, pointers and control structures in various programming languages (L6)
2. Design and implement basic concepts of subprograms in various programming languages (L6)
3. Design and implement basic concepts of OOPs, Multithreading and Exception handling in various programming languages. (L6)
4. Understand the basic knowledge of lambda calculus, functional programming languages, Programming with Scheme, Programming with ML (L2)
5. Understand the basic knowledge of Logic programming, Prolog and Multi-paradigm languages. (L2)

Unit I: Evolution of programming languages, Names, variables, binding, Type checking, scope Rules, primitive data types, strings, array types, associative arrays, Record types, union types, pointers and references, Arithmetic expressions, overloaded operators, relational & Boolean expressions Statements & mixed mode assignments, control structures

Learning Outcomes: Student should be able to
1. Apply the data types, declarations and expressions in the languages (L3)
2. Understand various control statements (L2)

Unit II:Subprograms& Design issues, Local reference, Parameter passing, Overloaded& Generic Methods, Design issues for functions, Semantics of call & Return, Implementing programs Stack & Dynamic local variables, Nested subprograms, Blocks, Dynamic scoping

Learning outcomes: Student should be able to
1. Understand various parameter passing methods (L2)
2. apply scoping in various languages (L3)

Unit III : Object-Orientation, Design issues for oops languages, Implementation of object oriented constructs, Concurrency: Semaphores, monitors, Message passing, Threads, Statement level concurrency, Exception handling, Event handling.

Learning outcomes: Student should be able to
1. Understand various concurrency techniques (L2)
2. apply object oriented concepts in the real time applications (L3)

Unit IV: Introduction to lambda calculus, fundamentals of functional programming languages, Programming with Scheme, Programming with ML.

Learning outcomes: Student should be able to
1. Understand the fundamentals concepts of functional programming languages (L2)

Unit V: Introduction, Logic programming overview, Basics of prolog, Applications of Logic programming, Programming with prolog, Multi-paradigm languages.

Learning outcomes: Student should be able to
1. Learn how to implement expert systems with prolog (L2)
2. Understanding the usage of predicate calculus in the logic programming languages (L2)

Text Books:

1) Robert W. Sebesta, “Concepts of Programming Languages”, Tenth Edition, Addison Wesley2012
2) Programming Languages, Principles & Paradigms, 2ed, Allen B Tucker, Robert E Noonan, TMH

References:

1. R. Kent Dybvig, “The Scheme programming language”, Fourth Edition, MIT Press, 2009
2. Jeffrey D. Ullman, “Elements of ML programming”, Second Edition, Prentice Hall, 1998
3. W. F. Clocksin& C. S. Mellish, “Programming in Prolog: Using the ISO Standard”, Fifth Edition, Springer, 2003

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19CSS-PC2102	Python Programming	3	0	0	3

COURSE OBJECTIVES:

- Understand structure and data types of python script.
- Implement iterations and functions in python.
- Implement modules and data structures using mutable & immutable objects.
- Understand object-oriented concepts on real world scenarios.
- Understand packages for statistics and gaming.

COURSE OUTCOMES:

1. Understand Basic Python Programming Environment and various data types.
2. Implement iterators and functions for data processing.
3. Implement different modules and objects to organize data.
4. Understand Object oriented concepts and handle different errors through exceptions
5. Analyze datasets using different mathematical packages and draw shapes using turtle.

Unit I:

Introduction: History of Python, Need of Python Programming, Applications, Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation.

Operators and Type Conversion: Types-Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison(Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations, type conversion.

Learning Outcomes:

After completing this chapter, student will be able to

1. Understand the environment of python. (L2)
2. Create and run simple scripts in python. (L2)
3. Understand data types and their conversions. (L2)
4. Understand operators for doing operations on different expressions. (L2)

Unit II:

Control Flow: Control Flow- if, if-elif-else, for, while, break, continue, pass.

Functions: Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Fruitful Functions (Function Returning Values), Scope of the Variables in a Function - Global and Local Variables, Anonymous Functions, Lambdas, map, reduce and filter.

Learning Outcomes:

After completing this chapter, student will be able to

1. Understand the iterations using looping structures. (L2)
2. Make decisions through conditional statements. (L2)
3. Understand functions to define call and pass as arguments. (L2)
4. Write anonymous functions for resolving complex problems. (L2)

Unit III:

Modules: Creating modules, import statement, from. Import statement, name spacing, builtin modules- os, random, math, cmath, pprint, json, request, date, RegEx.

Strings & Data Structures: String**,** String Formatting**,** List, Tuples, Sets, Dictionaries, Sequences, List Comprehension, itertools, built-in functions of all Objects.

After completing this chapter, student will be able to

1. Create and implement modules using import. (L3)
2. Understand different built-in modules. (L2)
3. Understand text processing using String Object. (L2)
4. Implement Data structures on different real time data. (L2)

Unit IV:

Object Oriented Programming OOP in Python: Classes, 'self variable', Methods, Constructor Method, Inheritance, Overriding Methods, Data hiding, Duck Typing and Decorators.

Error and Exceptions: Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions.

Learning Outcomes:

After completing this chapter, student will be able to

1. Implement Object oriented concepts with real world scenarios. (L2)
2. Understand Methods and decorators for annotating objects. (L2)
3. Understand error handling and handle exceptions. (L2)

Unit V:

Python Turtle Module: Directions, Positions, Colors, Drawing States and Shapes, Filling, Visibility.

Packages: Introduction to PIP**,** Installing packages using PIP.

Mathematical Libraries: NumPy, SciPy, Sympy, Pandas, StatsModels, Matplotlib and Gnuplot.

Learning Outcomes:

After completing this chapter, student will be able to

1. Understand Turtle for drawing shapes and applying colors. (L2)
2. Understand PIP to install new packages in python. (L2)
3. Apply mathematical libraries for analysing data sets. (L2)

TEXT BOOKS:

1. Python Programming: A Modern Approach, VamsiKurama, Pearson.
2. Learning Python, Mark Lutz, Orielly.

REFERENCE BOOKS:

1. Think Python, Allen Downey, Green Tea Press.
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage.

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2103	Data Structures through C++	3	0	0	3

Course Objectives:

1. Describe to searching and sorting techniques.
2. Applying stack and queue techniques for logical operations
3. Describe to list representation models in various types of applications
4. Design and Implementation of trees in various forms
5. Describe of orientation on graphs, graph traversals, spanning trees

Course Outcomes:

1. Analyze different searching and sorting Techniques.
2. Apply concepts of stacks and queues in different mathematical evaluations
3. Apply concepts of linked lists and implementation of different Linked Lists
4. Understanding ofnon linear data structures like trees and binary search trees their operations
5. Evaluating concepts of graphs and their applications.

Unit 1: Data structure-Definition, types of data structures, Concepts of Object Oriented programming language.
The Array as an Abstract Data Type, The Polynomial Abstract Data type- Polynomial Representation- Polynomial Addition. Spares matrices,Introduction- Sparse Matrix Representation.
Searching : List Searches using Linear Search, Binary Search.
Sorting Techniques: Basic concepts, Sorting by : insertion (Insertion sort), exchange (bubble sort, quick sort), distribution (radix sort) ,merging (merge sort) Algorithms and selection heap sort.

Learning Outcomes:
After completion of this unit, student will be able to

- Understand and remember algorithms and its analysis procedure(L2).
- Analysis procedure of search (L4).
- Analyze sorting techniques (L4).

UNIT-II: STACKS AND QUEUES
Templates in C++, Template Functions,template classes, The Stack Abstract Data Type,Implementation of stack operations , Expression-Infix,prefix, Postfix Notations, Conversions- Infix to Postfix,Infix to Prefix, Evaluation of postfix Expressions.
The Queue Abstract Data Type,Representation of Queue, Implementation of Queue operations,Circularqueues,Application.
Learning Outcomes:
After completion of this unit, student will be able to

- Understand working process of stack and Queue(L2)..
- Evaluating Arithmetic Expressions(L5).
- Apply Transformation of infix to postfix conversion(L3).

UNIT-III: LINKED LISTS

Introduction to Linked List, difference between Arrays and Linked List, singly linked lists –creation,insertion, deletion, search and Traversal operations, Representation of stack, queue, polynomial and Sparse Matrix using single linked list, circular lists ,doubly linked lists-insertion, deletion and traversal operations.

Learning Outcomes:

After completion of this unit student will be able to

- Understand the linked list process(L2).
- Apply linked list into polynomial expressions(L3).
- Analyze operation on different Linked lists(L4).

Unit 4: Trees: Basic tree concepts, Binary Trees: Properties, Representation of Binary Trees using arrays and linked lists, operations on a Binary tree , Binary Tree Traversals (Inorder,Preorder,Postorder), Binary search tree: Basic concepts, Binary search tree operations: insertion, deletion, Threaded Binary Trees.Heap: Basic concept,operations of max Heap.

Learning Outcomes:

After completion of this unit student will be able to

- Create Binary Tree using linked list and Arrays(L6)..
- Understanding different types of Binary tree Traversals(L2).
- Analyze different Operation of Binary Search Tree operations(L4)..

UNIT-V: GRAPHS

Introduction, Definition, The Graph Abstract Data Type, Graph Representation, Depth First Search and Traversal, Breadth First Search and traversal, Connected , Biconnected Components , Spanning Trees, Minimum Cost Spanning Trees: Prim' s Algorithm ,Kruskal's Algorithm. Single Source shortest Path: Dijkstra's Algorithm

Learning Outcomes:

After completion of this unit student will be able to

- Create the spanning tree from graphs(L6).
- Create minimal spanning tree by using different algorithms(L6).

TEXT BOOKS:

1. Data Structure with C, Seymour Lipschutz, TMH
2. Fundamentals of Data Structures in C++, Ellis Horowitz, Sartaj Sahni and Dinesh Mehta, 2nd Edition, Universities Press (India) Pvt. Ltd.
3. Data Structures using C,ReemaThareja, Oxford
4. Data Structures, 2/e, Richard F, Gilberg ,Forouzan, Cengage
5. Data structures and algorithm analysis in C, 2nd ed, mark allenweiss

REFERENCE BOOKS:

1. Data Structures and Algorithms, 2008,G.A.V.Pai, TMH
2. Classic Data Structures, 2/e, Debasis ,Samanta,PHI,2009
3. Fundamentals of Data Structure in C, 2/e, Horowitz,Sahni, Anderson Freed,UniversityPrees

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2104	Computer Organization & Architecture	3	0	0	3

COURSE OBJECTIVES:

- Understand the Architecture of a Modern Computer with its Various Processing Units.
- Also, the Performance Measurement of the Computer System.
- Understand and Analyze Instruction Formats and Addressing modes
- In addition to this the Memory Management System of Computer
- Understand Control unit and Pipelining concepts

COURSE OUTCOMES:

1. Understand the basic components of a computer, including CPU, memories, and input/output, and their organization, number representations
2. Explore the representation of data, addressing modes, and instructions sets and logical, shift and rotate operations that are performed by computers.
3. Understand the basic knowledge on arithmetic, logic and branch instructions.
4. Understand the input output organization like handling multiple devices, buses and interface circuits
5. Illustrates the organization of memory and memory management system.

UNIT – I:

Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures,System Software, Performance of a Computer, Number Representation, IEEE 754 Floating Point Representation.

Learning Outcomes:

At the end of this unit, Students will be able to,

1. Acquire the Knowledge on Basic Structure of a Computer.

2. Analyze the Performance of a Computer System.

UNIT – II:

Machine Instructions and Addressing Modes: Register Transfer Notation (RTN), Assembly Language Notation (ALN), Basic Program Execution, Instruction Cycle, Instruction Formats:Zero, One, Two, and Three Address Instructions, Addressing Modes.

Types of Instructions:Data Transfer Instructions, Data Manipulation Instructions, Control Flow Instructions, Basic I/O Operations.

Learning Outcomes:

At the end of this unit, Students will be able to,

1. Apply Logical and Shift Instructions towards Arithmetic Operations.

2. Analyze the Addressing Modes.

UNIT – III:

Input / Output Organization: Accessing I/O Devices, Interrupts: Interrupt Hardware, Enabling& Disabling Interrupts, Handling Multiple Devices.

I/O Techniques: Programmed I/O, Interrupt – Driven I/O, Direct Memory Access (DMA).

Buses: Synchronous Bus, Asynchronous Bus, Interface Circuits: Serial & Parallel Ports.

Standard I/O Interface: Peripheral Component Interconnect Bus (PCI), Small Computer System Interface (SCSI), Universal Serial Bus (USB).

Learning Outcomes:

At the end of this unit, Students will be able to,

1. Analyze the Interrupts and DMA.

2. Understand the Peripheral Components and Bus Connections.

.

UNIT – IV:

The Memory Systems: Memory Hierarchy, Memory System Considerations, RAM, ROM.
Cache Memory: Introduction to Cache, Mapping Techniques, Memory Interleaving, Updation Techniques, Replacement Policies.
Secondary Storage: Magnetic Hard Disks, Optical Disks, Flash Memory.
Learning Outcomes:
At the end of this unit, Students will be able to,

1. Analyze the Difference Between the Functionality of RAM and ROM.
2. Apply Cache Organization and Different Mapping Techniques.
3. Understand the Secondary Memory Organizations.

.

UNIT – V:

Processing Unit: Fundamental Concepts: Register Transfers, Performing an Arithmetic or Logic Operation, Fetching a Word from Memory.
Control Unit: Hardwired Control Unit (HCU), Micro Programmed ControlUnit (MCU).
Pipelining: Basic Concepts of Pipelining, Hazards in Implementing Pipelining, Resolving Techniques.
Learning Outcomes:
At the end of this unit, students will be able to,

1. Analyze the Micro Instructions and Operations on Registers.
2. Analyze the Functionality of Control Unit.
3. Understand the Pipelining Concepts.

TEXT BOOKS:

1. Computer Organization, Carl Hamacher, ZvonksVranesic, SafeaZaky, 5th Edition,McGraw Hill.
2. Computer Architecture and Organization, John P. Hayes, 3rd Edition, McGrawHill.

REFERENCE BOOKS:

1. Computer Organization and Architecture – William Stallings Sixth Edition,Pearson/PHI.
2. Computer Organization and Architecture by Smruti RSarangi.
3. Computer Organization and Design: The Hardware/Software Interface by David A.Patterson and John L.Hennessy.
4. J.P. Hayes, "Computer Architecture and Organization", McGraw – Hill,1998.
5. Computer Organization and Design by P Pal Chaudari.

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19CSS-PC2103	Python Programming Lab	0	0	3	1.5

COURSE OBJECTIVES:

- Introduction to Scripting Language Experience with an interpreted Language.
- Exposure to various problems solving approaches of computer science
- Making Software easily right out of the box

COURSE OUTCOMES:

CO1: Understand the working environment of Python and its program structure.
CO2: Implement conditional and iterative statements.
CO3: Create custom modules and functions to handle different operations.
CO4: Implement Object oriented concepts through real time scenarios and handle errors.
CO5: Design different shapes and objects using turtle graphics.

Experiment 1:

a. Demonstrate the python script by running in Interactive and Script Mode.
b. Write a python script to read using input() and display using print() functions.

Experiment 2:

a. Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
b. Write a python script to make use of all conversion functions.

Experiment 3:

a. Write a program to take input as integer N and check whether N is Pronic Number or not. (Product of two consecutive numbers is pronic N(N+1): Eg 110 = 10*11)
b. Write a program to take input as integer N and check whether N is Niven number or not. (Niven number is sum of digits of given number should be 9)
c. Write a program to check whether given number N is N-Series number or not**.** (Eg. 135 is N-Series Number because $1^1+3^2+5^3 = 135$ and some others are 89, 175, 518 etc)

Experiment 4:

a. Write a python script to print Prime pairs within a given range of numbers. (Hint N=20, then (3,5) (5,7) (11,13) (17,19) are prime pairs)
b. By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

Experiment 5:

a. Write a program to take input as String S and print frequency of each character in S using List data structure.
b. Write a program to take input as String S contains combination of uppercase, lowercase, numbers and special symbols, then print Uppercase followed by Lowercase followed by numbers followed by special symbols.

Experiment 6:

a. Write a python script to take input as String sentence S and print each word count using dictionary.
b. Using Slice operator, write a python script to take input as String S and check whether string is palindrome or not.

Experiment 7:

a. Write a python script to implement different arguments in a function.
b. Write a python script to implement Anonymous function.
c. Write a python script to implement map(), reduce() and filter() functions.

Experiment 8:

a. Find mean, median, mode for the given set of numbers in a list.
b. Write a python script to understand List Comprehensions.

Experiment 9:

a. Using Reg Ex object check whether given phone number, email address and password is valid or not.
b. Using command Line arguments of python, read two string S1 and S2 and check whether S1 and S2 are anagrams or not.
c. Using date module, write a python script to take input as Date of birth (DOB) and current date(CD) and print age of the person.

Experiment 10:

a. Write a python script to implement itertools and izip modules.

Experiment 11:

a. Create a module named "Lendi" and create functions addStudent, removeStudent, searchStudent. Access the above module using import statement.

Experiment 12:

a. Install packages using PIP.
b. Write a script that imports requests and json to fetch content from the page.
c. Write a script that uses the random module for generating random numbers.

Experiment 13:

a. Using Python OOPS, create a class, constructor, method, __str__ and __repr__ for:
   i. Employee
   ii. Student

Experiment 14:

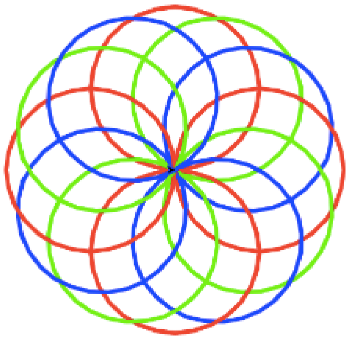
a. Write a python script to implement Decorators for methods in a class.
b. Write a program to implement Python duck Typing.

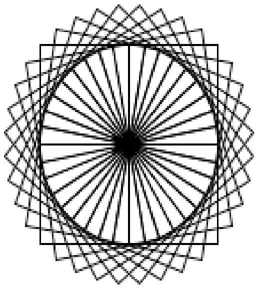
Experiment 15:

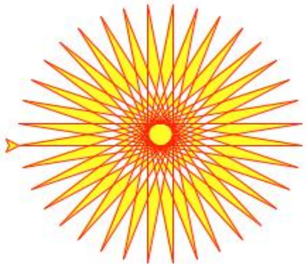
a. Write a python program to implement Exceptions hierarchy.
b. Create a user defined Exception named "FundsLessException" and raise the exception when there are no enough funds in the bank account.

Experiment 16:

a. Write a program to implement the following figures using turtle:







Experiment 17:

a. Using NumPy, implement different matrix operations in python.
b. Using SciPy, implement polynomials and linear algebra expressions.
c. Using pandas, read the data from CSV and JSON files.

Experiment 18:

a. Using MatPlotlib library, plot the graph with all different plot types.(Pie Chart, Area Plot, Scatter Plot, Histogram and Bar Graph)

APPLICATIONS:

- Web Application Development and Scraping
- Designing Games
- Machine Learning and AI based applications
- Data Science and Visualization
- Embedded and CAD Applications

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2105	Data Structures through C++ Lab	0	0	3	1.5

OBJECTIVES:

- To develop skills to design and analyze simple linear and non linear data structures
- To Strengthen the ability to identify and apply the suitable data structure for the given real world problem
- To Gain knowledge in practical applications of data structures

COURSE OUTCOMES:

1. Analyze different searching and sorting Techniques.
2. Apply logical ability to solve the problems related to Linked lists
3. Apply logical ability to solve the problems of Stacks and Queues Applications
4. Implement trees and their operations
5. Evaluate Binary search tree Operations

List of Programs:

1. Write a C++ programs to implement the following search algorithms:
   i) Linear Search ii) Binary Search

2.Write a C++ programs to implement the following sorting algorithms:
   i) Bubble Sort ii) Insertion Sort iii) Selection Sort.

3.Write a C++ programs to implement the following sorting algorithms
   i) Merge Sort ii) Quick Sort.

4.Write a C++ programs that implement the following data structures using arrays:
   i) Stack ii) Queue.

5. Write a C++ programs to Implement of Circular Queue using Arrays

6. Write a C++ programs to Evaluate of postfix expression.

7.Write a C++ programs to implement the following types of Lists
   i) Singly linked list iii) Doubly linked list.

8.. Write a C++ program to perform the following operations

- i)insert an element into a binary search tree.
- ii)Delete an element from a binary search tree.
- iii) Search for a key element in a binary search tree.
- iv) Tree Traversals

9. Write a C++ programs for the implementation of BFS for a given graph.

10. Write a C++ programs for the implementation of DFS for a given graph.

11. Write a C++ program for the implementation of Prim's algorithm to obtain the minimum cost spanning tree from a connected undirected graph.

12. Write a C++ program to implement Dijkstra's algorithm for the single source shortest path problem.

REFERENCES:

1. G A V PAI, "Data Structures and Algorithms, Concepts,Techniques and Applications", Volume-1, 1st Edition, TataMcGraw-Hill, 2008.
2. Richard F. Gilberg& Behrouz A. Forouzan, "Data Structures,A Pseudo code Approach with C", 2nd Edition, CengageLearning India Edition, 2007.

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19BSH-MC2101	Essence of Indian Traditional Knowledge	3	0		0

COURSE OBJECTIVES:

- Facilitate the students with the concepts of Indian traditional knowledge and to make them understand the importance of roots of knowledge system.(L2)
- Importing basic principle of thought process reasoning and inference sustainability of Indian traditional knowledge system(L2)
- Comprehend the legal framework, traditional knowledge, biological diversity act 2002 and geographical indication act 2003.(L3)
- Focus on traditional knowledge and intellectual property mechanism
- Analyze traditional knowledge in various sectors(L3)

COURSE OUTCOMES:

After completion of the course, students will be able to:

1. Knowledge about the concept of traditional knowledge and analyze social context
2. Apply significance of traditional knowledge protection
3. Analyze various enactments related to the protection of plant varieties
4. Evaluate desired concepts of Intellectual property to protect the traditional knowledge.
5. Compare the traditional knowledge in various sectors.

Unit-I:

Introduction to Traditional Knowledge: Define traditional knowledge, nature and characteristics, scope and importance, kinds of traditional knowledge, the physical and social contexts in which traditional knowledge develop, the historical impact of social change on traditional knowledge systems. Indigenous Knowledge (IK), characteristics, traditional knowledge vis-à-vis indigenous knowledge, traditional knowledge Vs western knowledge traditional knowledge vis-à-vis formal knowledge

Learning Outcomes:

At the end of the unit the student will able to:

- Recognize the social change in traditional knowledge(L4)
- Contrast and compare characteristics importance kinds of traditional knowledge. (L2)
- Analyze physical and social contexts of traditional knowledge. (L4)

Unit-II:

Protection of traditional knowledge: The need for protecting traditional knowledge Significance of TK Protection, value of TK in global economy, Role of Government to harness TK.

Learning Outcomes:

At the end of the unit the student will able to:

- Identify the need of protecting traditional knowledge. (L2)
- Apply significance of TK protection. (L3)
- Analyze the value of TK in global economy. (L3)
- Evaluate the role of government in harnessing Traditional Knowledge. (L4)

Unit-III:

Legal framework and Traditional knowledge in Food: A: The Scheduled Tribes and Other Traditional Forest Dwellers (Recognition of Forest Rights) Act, 2006, Plant Varieties Protection and Farmers Rights Act, 2001 (PVPFR Act);B:The Biological Diversity Act 2002 and Rules 2004, the protection of traditional knowledge bill, 2016. Importance of food – Styles of food-traditional food- Modern Food- Factors influencing food choice- Economic and Physical Determinants- Uniqueness of Culture in Food.

Learning Outcomes:

At the end of the unit the student will able to:

- Contrast and compare the Scheduled Tribes and other traditional forest dwellers(L2)
- Analyze plant variant protections and evaluate farmers right act(L4)
- Evaluate food security and protection of TK in the country(L5)

Unit-IV:

Traditional knowledge and intellectual property: Systems of traditional knowledge protection- Legal concepts for the protection of traditional knowledge- Certain non IPR mechanisms of traditional knowledge protection- Patents and traditional knowledge- Strategies to increase protection of traditional knowledge- global legal FORA for increasing protection of Indian Traditional Knowledge.

Learning Outcomes:

At the end of the unit the student will able to:

- Evaluate strategies to increase the protection of traditional knowledge and Intellectual Property Rights (L4)
- Apply systems of traditional knowledge protection. (L3)
- Analyze legal concepts for the protection of Traditional Knowledge. (L4)

Unit-V:

Traditional Knowledge in Different Sectors: Traditional knowledge and engineering, Traditional medicine system, TK and biotechnology, TK in agriculture,

Learning Outcomes:

At the end of the unit the student will able to:

- Compare traditional knowledge in different sectors. (L2)
- Apply traditional knowledge in engineering. (L3)

Reference Books:

1. Traditional Knowledge System in India, by AmitJha, 2009.
2. Traditional Knowledge System and Technology in India by Basanta Kumar Mohanta and Vipin Kumar Singh, PratibhaPrakashan 2012.
3. Traditional Knowledge System in India by AmitJha Atlantic publishers, 2002
4. "Knowledge Traditions and Practices of India" KapilKapoor, Michel Danino

E-Resources:

1 https://www.utrechtjournal.org/articles/10.5334/ujiel.283/

2 https://en.wikipedia.org/wiki/Traditional_knowledge

3 https://www.scconline.com/blog/post/2018/04/23/protecting-traditional-knowledge-the-india-story-till-date/

4 https://sciencebusiness.net/news/72773/India-leads-the-way-in-protecting-traditional-knowledge

II Year –I Semester

Subject Code	Subject Name	L	T	P	C
R19BSH-MC2102	Employability Skills - I	0	0	2	0

Course Objectives

- Aims to help learners develop their English language skills, particularly those planning to appear for Competitive Exams that test their English Language abilities.
- Gains the power of expression through rich Vocabulary.
- Imparts critical reading strategies for comprehension of complex texts
- Provides training and opportunities to develop fluency in English through participation in formal group discussions and Self Introductions.
- Demonstrates good writing skills for effective Paragraph Writing, Essay Writing and formal correspondence through Emails.
- Encourages use of a wide range of grammatical structures, Phrases, Clauses and Idioms in speech and writing.

Course Outcomes

1. Enable students to identify Parts of Speech and use them flawlessly, write Emails in formal correspondence effectively, participate confidently by introducing oneself in any formal discussion.
2. Attain Language Proficiency & Accuracy through Contextualized Vocabulary, Verb forms, Tense and subject verb agreement, produce coherent expressions for professional writing, introduce themselves unhesitatingly with Task-Based Activities.
3. Develop the fluency and accuracy to write Technical Reports and Emails for professional communication by using appropriate vocabulary and participate confidently in any formal discussion.
4. Assimilate lifelong reading habit to comprehend a passage for its gist. Avoid the errors in both Speech & Writing and write Letters and Emails for official communication.
5. Realise the technical communicative competence and attainment of grammaticallycorrect structures for formal communication.

Unit 1(10 Hours (4L+6P))

Vocabulary: How to talk about actions. Grammar:Using and Identifying Parts of Speech accurately . Writing: Paragraph Writing and formal correspondence through Emails.Speaking:Background to Group discussions & Self-introductions.

Learning Outcomes
At the end of the module, the learner will be able to

➢ Acquire vocabulary and use it contextually(L2)
➢ identify parts of speech and use them flawlessly in both Speech and Writing (L3)
➢ write paragraphs and Emails in formal correspondence effectively (L3)
➢ participate confidently in any formal discussion and introduce themselves unhesitatingly (L3)

Unit 2(10 Hours (4L+6P))

Vocabulary: How to talk about various speech habits.Grammar: Learning Verb forms, Tenses and Subject-verb agreement and using them accurately in both Speaking and Writing contexts. Writing :Essay Writing and formal correspondence through Emails. Speaking:Four major areas
-Subject Knowledge, Oral Communication Skills, Leadership Skills and Team Management-of GD;Real time GDs for Evaluation.

Learning Outcomes
At the end of the module, the learner will be able to

- ➢ Acquire vocabulary and use it contextually (L2)
- ➢ use Verb forms,Tense and subject verb agreement for effective speaking and writing (L3)
- ➢ produce coherent expressions for professional writing (L4)
- ➢ participate confidently in any formal discussion and introduce themselves unhesitatingly (L3)

Unit 3(8 Hours (2L+6P))

Vocabulary: How to insult your enemies. Grammar: Sentence Analysis & Synthesis - Voice, Degrees of Comparison, Reported Speech and Types & Forms of sentences. Writing: Report writing and Emails for formal correspondence. Speaking: Roles in structured GDs; real time GDs for practicing the above roles.

Learning Outcomes
At the end of the module, the learner will be able to

- ➢ Acquire vocabulary and use it contextually(L2)
- ➢ identify the complexity in the structure of a sentence (L2)
- ➢ write technical reports and emails for professional communication (L3)
- ➢ participate confidently in any formal discussion and introduce themselves unhesitatingly (L3)

Unit 4 (8 Hours (2L+6P))

Vocabulary: How to flatter your friends. Grammar: Common errors and Correction of Sentences Reading:Reading Comprehension passages through Skimming and Scanning and understanding the gist or the specific purpose of them.. Writing: Letter writing and Emails. Speaking:Advantages of GDs for hiring process ; real time GDs for evaluating.

Learning Outcomes
At the end of the module, the learner will be able to

- ➢ Acquire vocabulary and use it contextually(L2)
- ➢ comprehend a passage and know its gist(L3)
- ➢ avoid the errors in both Speech and Writing (L2)
- ➢ write letters and emails for official communication(L3)
- ➢ participate confidently in any formal discussion and introduce themselves unhesitatingly (L3)

Unit 5(10 Hours (4L+6P))

Vocabulary & Grammar: High-frequency words for all competitive exams, Clause ,Phrase& Idioms. Reading: Reading for Comprehending Writing:Business Letters and Emails Speaking:Group Discussions for Evaluation

Learning Outcomes
At the end of the module, the learner will be able to

- ➢ Acquire vocabulary and use it contextually(L2)
- ➢ use grammatically correct structures for formal communication (L3)
- ➢ write Business Letters effectively (L3)
- ➢ participate confidently in any formal discussion and introduce themselves unhesitatingly

(L3)

Reference Books

1. Bailey, Stephen. Academic writing: A handbook for international students. Routledge, 2014.
2. Chase, Becky Tarver. Pathways: Listening, Speaking and Critical Thinking. Heinley ELT; 2nd Edition, 2018.
3. Skilful Level 2 Reading & Writing Student's Book Pack (B1) Macmillan Educational.
4. Hewings, Martin. Cambridge Academic English (B2). CUP, 2012. (Student Book, Teacher Resource Book, CD & DVD)
5. Word Power Made Easy by Norman Lewis

ASSESSMENT

The learners will demonstrate their knowledge and abilities through completion of the following required assessments while or at the end of this course. —2 Quizzes, 1 Professional Certificate, 3 Activities on LSRW skills.

Quiz:
Quiz is conducted on Grammar& Vocabulary. Each Quiz consists of 50 questions and will be scaled down to 5 Marks. Two quizzes are conducted. One after the 3rd unit, the other, after the last unit. Duration of any quiz is 1hr 30 Min only.These Quizzes are Computer Based Tests (CBT)

Professional Certificate:
An International Language Assessment Certificate secured on B1 of Common European Framework for Reference (CEFR) scale.

Activities on LSRW skills:

Interviews:
The candidate has to interview one celebrity of his/her own choice.
The recorded 5-7 min video of the candidate should be uploaded on the ELCS LABLendiYoutube Channel with the help of concerned English Teacher

The Evaluation Parameters:
Quality of the Questionnaire(3M)
Body Language & Confidence of the candidate(5M) Youtube likes & Comments(2M)

E-mails:
Each student is required to submit 5 independently written Emails during the course. Specific requirements for each one are accessed on the following Link:
https://docs.google.com/document/d/1IXuzjjmfiOLI23t8xlbLwNefRzIIXi9aOi3XkSHIK_Q/edit?usp=sharing

Listen to Speak:
Students are expected to watch and listen to any one of the 10 given educational video and audio clips to express their point of view. After watching, they will have the opportunity to share their points of view about some of the everyday issues that they can relate to. They

have to explain and justify their reasoning to a team of three peers to explore their verbal expressions and their points of view before an External Examiner.

The following is the link to access those clippings:
https://docs.google.com/document/d/1tFuQ_43AVAHKJGVs9AeOODHJTnQMoydqcodSgENaZ3o/edit?usp=drivesdk
Details of Peer Evaluation & Assessment Parameters are available on the following Link:
https://docs.google.com/document/d/16l_PUzaOONnjpvMYVzE3XAYUBNhqMK9PbdDOPGIef_8/edit?usp=sharing

Grading:

Assessment Model	**Points**
Quiz-1	**10**
Quiz-2	**10**
Professional Certificate with B1 or above or Activity of Interview	**10**
E-Mails	**10**
Listen to Speak Activity	**10**
Total	**50**

Pass Criterion:

1. Student has to Secure 30 Marks to pass this examination
2. Student who is having a certificate of any International standard of English he/she has to secure a Minimum 20 Marks in this examination (Certificate+20 Marks) to pass the summative exam.
3. Student who is not having a certificate has to clear the exam with 30 marks mandatorily.
4. Clearing all categories is mandatory. Need to get 60% each category.
5. 20M +Certificate=Successful or 30M+No certificate=Successful

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19BSH-MA2201	Probability & Statistics **(Common to CSE, CSSE & CSIT)**	3	0	0	3

Course Objectives:

- To familiarize the data Science concepts.
- To impart knowledge in basic concepts and few techniques in probability and statistics in relation to the engineering applications.

Course Outcomes:

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

1. Evaluate correlation and regression for the given data.(L2)
2. Apply Baye's theorem to probabilistic experiments.(L3)
3. Apply discrete and continuous probability distributions to the real time problems .(L3)
4. Design the components of a classical hypothesis test.(L3)
5. Infer the statistical inferential methods based on small sampling tests. (L3)

Unit I: Descriptive statistics and methods for data science

Introduction to Descriptive statistics, measures of central tendency, measures of variability (spread or variance), skewness, Kurtosis.

Correlation and Regression: Correlation, correlation coefficient, rank correlation. Linear regression coefficients, principle of least squares, method of least squares, regression lines, multiple regression, Logistic regression.

Learning Outcomes:

At the end of this unit, the student will be able to

- Understand the various methods of central tendency and variability.(L2)
- evaluate correlation, correlation coefficient, rank correlation. (L2)
- apply method of least squares to find regression coefficients. (L3)

UNIT II: Probability

Probability, probability axioms, addition law and multiplicative law of probability, conditional probability, Baye's theorem, random variables (discrete and continuous), probability density functions, properties, mathematical expectation.

Learning Outcomes:

At the end of this unit, the student will be able to

- make use of probabilities of events in finite sample spaces from experiments. (L3)
- apply Baye's theorem to real time problems. (L3)

UNIT III: Probability Distributions

Probability distribution - Binomial, Poisson distribution and normal distribution-their properties.

Learning Outcomes:

At the end of this unit, the student will be able to

- find the binomial, poisson and normal distribution of the given data data. (L2)

Unit IV: Estimation and Testing of hypothesis, large sample tests

Introduction to Sampling, parameters, statistics, sampling distribution, point estimation, interval estimations, null hypothesis, alternative hypothesis, the critical & acceptance regions, level of significance, type-1 and type-2 of errors.

Large Sample Tests: Test for single proportion, difference of proportions, test for single mean and difference of means, confidence interval for parameters in one sample and two sample problems.

Learning Outcomes:

At the end of this unit, the student will be able to

- explain the concept of estimation, interval estimation and confidence intervals. (L2)

- apply the concept of hypothesis testing for large samples. (L4)

Unit V: Small sample tests

Student t-distribution (test for single mean, two means and paired t-test), testing of equality of variances (F-test), $\chi 2$ - test for Single varience, $\chi 2$ - test for goodness of fit, ANOVA.

Learning Outcomes:

At the end of this unit, the student will be able to

- apply the concept of testing hypothesis for small samples to draw the inferences. (L3)
- estimate the goodness of fit. (L3)
- apply ANOVA test to compare the data (L3)

Text Books:

1. Veerarajan T., Probability, Statistics and Random Processes, 3rd edition, Tata McGraw-Hill, New Delhi, 2008.
2. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 44/e, 2017.
3. Miller and John E. Freund, Probability & Statistics for Engineers, Prentice Hall of India.

References:

1. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, 2008.
2. T. K. V. Iyengar, B. Krishna Gandhi and Others, Probability & Statistics, S. Chand & Company.
3. S. C. Guptha and V. K. Kapoor, Fundamentals of Mathematical Statistics, 11/e, Sultan Chand and Sons Publications, 2012.
4. Sheldon M. Ross ,Introduction to Probability and Statistics for Engineers and Scientists, Academic Press,2009.
5. Ronald E.Walpole, Sharon L.Myers, Keying Ye, Probability and Statistics for Engineers and Scientists, Pearson, 2007.
6. Ravichandran, Probability and Statistics for Engineers, Wiley, 2019.

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2201	Java Programming	3	1	0	4

Course Objectives:

- To understand the structure and environment of Java.
- To implement the relationship between objects.
- To apply data hiding strategy in objects.
- To implement text processing and error handling.
- To organize data using different data structures.
- To create multi threaded graphical user interface applications.

Course Outcomes:

1. Understand the environment of JRE and Control Statements.
2. Implement real world objects using class Hierarchy.
3. Implement generic data structures for iterating distinct objects.
4. Implement error handling through exceptions and file handling through streams.
5. Design thread-safe GUI applications for data communication between objects.

Unit I: Java Environment and Program Structure

History of Java, Features, Applications, Java Installation - JDK and JRE, JVM Architecture, OOPS Principles, Class and Object, Naming Convention, Data Types, Type Casting, Type Conversion, Wrapper classes, Operators, instance of operator, Command Line Arguments, Decision making, Arrays, and Looping statements.

Learning Outcomes: Student will be able to

1. Understand architecture of Java Virtual Machine.(L2)
2. Understand the structure of java program and its environment. (L2)

Unit II: Class Hierarchy & Data Hiding

Property, Method, Constructor, Inheritance (IS-A) , Aggregation and Composition (HAS-A), this and super, static and initialize blocks, Method overloading and overriding, static and final keywords, Types of Inheritance, Compile time and Runtime Polymorphism, Access Specifiers and scope, packages and access modifiers, Abstract class, Interface, Interface Inheritance, Achieving Multiple Inheritance, Class casting, Object Cloning, Inner Classes.

Learning Outcomes: Student will be able to

1. Understand the class hierarchy and their scope. (L2)
2. Implement relationship between objects. (L3)
3. Understand data hiding and nested classes. (L2)
4. Implement data type casting and cloning of objects. (L3)

Unit III: Strings and Collections

String: Methods,StringBuffer and StringBuilder, StringTokenizer,

Collections: Exploring java.util.*, Scanner, Iterable, Collection Hierarchy, Set, List, Queue and Map, Comparable and Comparator, Iterators: foreach, Enumeration, Iterator and ListIterator.

Learning Outcomes: Student will be able to

1. Understand the usage of String and its properties and methods.(L2)
2. Understand data structures and Iterators. (L2)

3. Create the data structures and implement different utility classes. (L3)

Unit IV: IO and Error Handling

IO Streams: Exploring java.io.*, Character and Byte Streams, Reading and Writing, Serialization and De-serialization,

Error Handling: Error vs Exception, Exception hierarchy, Types of Exception, Exception handlers, User defined exception, Exception propagation.

Learning Outcomes: Student will be able to

1. Understand character and byte streams. (L2)
2. Understand the hierarchy of errors and exceptions. (L2)
3. Implement data streams and exception handlers. (L3)

Unit V: Threads and GUI

Multi Threading: Process vs Thread, Thread Life Cycle, Thread class and Runnable Interface, Thread synchronization and communication.

GUI: Component, Container, Applet, Applet Life Cycle, Event delegation model, Layouts, Menu, MenuBar, MenuItem.

Learning Outcomes: Student will be able to

1. Understand the Thread Life Cycle and its scheduling.(L2)
2. Implement the synchronization of threads. (L2)
3. Create graphical components using Abstract window toolkit. (L3)

TEXT BOOKS:

1. The complete Reference Java, 8th edition, Herbert Schildt, TMH.
2. Programming in JAVA, Sachin Malhotra, SaurabhChoudary, Oxford.
3. Introduction to java programming, 7th edition by Y Daniel Liang, Pearson.
4. Java: How to Program, 9th Edition (Deitel) 9th Edition.
5. Core Java: An Integrated Approach, Java 8 by R. Nageswara Rao.

REFERENCE BOOKS:

1. Swing: Introduction, JFrame, JApplet, JPanel, Componets in Swings, Layout Managers in
2. Swings, JList and JScrollPane, Split Pane, JTabbedPane, JTree, JTable, Dialog Box.

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2202	Operating Systems	3	0	0	3

Course objectives:

- Provide knowledge about the services rendered by operating systems.
- Present detail discussion on processes, threads and scheduling algorithms.
- Expose the student with different techniques of handling deadlocks.
- Discuss various file-system implementation issues and memory management techniques..
- Learn the basics of Linux system and Android Software Platform**.**

Course Outcomes:

1. Understand the importance of operating systems and different types of system calls(L2).
2. Analyze process scheduling algorithms and various IPC mechanisms.(L4).
3. Understand the synchronization process, different ways for deadlocks handling. (L2).
4. Analyze different page replacement methods, various File management techniques. (L4).
5. Understand storage, mass storage structures and basics of Android operatingsystems. (L2).

UNIT-I: **Operating Systems Overview:**Introduction: what is an operating system, Types of operating systems, operating systems concepts, operating systems services, Introduction to System call, System call types. Operating System Generation

Learning outcomes: Student should be able to

1. Understand operating system structure and functions. (L2).
2. Understand operating system services and system calls (L2).

UNIT–II: Process Management:Process concept: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication.

Multithreaded Programming: Overview, Multithreading models, Threading Issues.

Process scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

Learning outcomes: Student should be able to

1. Identify various message sharing mechanisms used in IPC. (L2).
2. Understand how to handling multiple threads. (L2).
3. Differentiate between preemptive, non-preemptive and real time CPU scheduling (L2).

UNIT-III: Synchronization: Process Synchronization, The Critical-Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization examples

Principles of deadlock – System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery form Deadlock

Learning outcomes: Student should be able to

1. Analyze various solutions for process synchronization. (L4).
2. Analyze the reasons for deadlocks and proposed solutions to detect, avoid, recovery from deadlocks. (L4).

UNIT-IV: Memory Management: Memory Managementstrategies: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.

Virtual Memory Management: Virtual Memory, Demand Paging, Page-Replacement Algorithms, Thrashing..

Learning outcomes: Student should be able to

1. Demonstrate the ability to implement various memory management techniques (L2)
2. Illustrate various demand paging techniques. (L2).

UNIT-V: File system Interface- the concept of a file, Access Methods, Directory and Disk structure, File system mounting.

File System implementation: File system structure, allocation methods, free-space management

Mass-storage structure: overview of Mass-storage structure, Disk scheduling, Device drivers

Android Software Platform: Android Architecture, Operating System Services, Android Runtime Application Development, Application Structure, Application Process management.

Learning outcomes: Student should be able to

1. Identify various file management and optimization techniques. (L2).
2. Understand how data streams are exchanged between I\O subsystems. (L2).
3. Analyze various storage structures to store the data in secondary memory. (L4).
4. Analyze different disk scheduling algorithms. (L4).
5. Understand Android operating system environment. (L2).

Text Books:

1. Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9th edition, Wiley, 2013.
2. Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008. (for Interprocess Communication and File systems).

References:

1. Tanenbaum A S, Woodhull A S, Operating Systems Design and Implementation, 3rd edition, PHI, 2006.
2. Dhamdhere D M, Operating Systems A Concept Based Approach, 3rd edition, Tata McGraw-Hill, 2012.
3. Stallings W, Operating Systems -Internals and Design Principles, 6th edition, Pearson Education, 2009.
4. Nutt G, Operating Systems, 3rd edition, Pearson Education, 2004.

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2203	Database Management Systems	3	0	0	3

Course objectives:

- Learn the fundamental concepts of database systems.
- Enable students to design ER diagram for any customized applications
- Learn simple and Complex queries using SQL.
- Learn schema refinement techniques (Normalization).
- Knowledge about transaction and recovery techniques.

Course Outcomes:

1. Understand the fundamental elements of database management systems.
2. Design ER-models to represent simple database application scenarios.
3. Design principles for logical design of database including the ER model and normalization approach.
4. Understand the basics optimization techniques of transaction processing and concurrency control.
5. Understand the basics of storage structures, indexing and page organization methods including B-Tree and Hashing.

UNIT-I

INTRODUCTION-Database system, Characteristics (Database Vs File System),Database Users (Actors on Scene, Workers behind the scene), and Advantages of Data base systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Learning outcomes: Student will be able to

1. Distinguish between Database System and File System (L2)
2. Categorize different kinds of data models (L2)

UNIT-II

RELATIONAL MODEL: Introduction to relational model, concepts of domain,attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance , Relational algebra, Relational Calculus.

ENTITY RELATIONSHIP MODEL: Introduction, Representation of entities,attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.

Learning Outcomes: Student will be able to

1. Develop E-R model for the given problem (L6)
2. Knowledge about integrity constraints in relational model (L1)

UNIT-III

SCHEMA REFINEMENT (NORMALIZATION): Purpose of Normalization orschema refinement, concept of functional dependency, normal forms based on functional dependency(1NF, 2NF and 3NF), concept of surrogate key, Boyce-codd normal form(BCNF), Lossless join and dependency preserving decomposition, Fourth normal form(4NF).

Learning Outcomes: Student will be able to

1. Differentiate between various normal forms based on functional dependency (L2)
2. Apply Normalization techniques to eliminate redundancy (L3)

UNIT-IV

TRANSACTION AND RECOVERY: Transaction State,Implementation of Atomicity and Durability, Concurrent Executions, Serializability,Deadlocks in transactions, Recoverability, Implementation of Isolation, Testing for Serializability, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm.

Learning Outcomes: Student will be able to

1. Summarize transaction properties and recoverability (L2)

UNIT-V

File Organization and Indexing, File Types, File Operations ,Cluster Indexes, Primary and Secondary Indexes , Index data Structures, Hash Based Indexing: Tree based Indexing, Indexes and Performance Tuning

Learning Outcomes: Student will be able to

1. Understand basic concepts ofFile Organization and Indexing (L2)

Text Books:

1. Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke,TMH
2. Database System Concepts,5/e, Silberschatz, Korth, TMH
3. Introduction to Database Systems, 8/e C J Date, PEA..

Reference Books:

1. Database Management System, 6/e RamezElmasri, Shamkant B. Navathe, PEA
2. Database Principles Fundamentals of Design Implementation and Management, Carlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2204	Formal Languages and Automata Theory	3	0	0	3

COURSE OBJECTIVES:

This course is designed to:

- Introduce the student to the concepts of Theory of computation in computer science and The students should acquire insights into the relationship among formal languages,
- Formal Grammars and automat.
- Analysis of Finite State Machine, its representation and automation. Description of finite automata, variants in it and their equivalence.
- Minimization, optimization of finite automata, regular expressions and equivalence of finite automata and regular expressions
- Illustration about grammars, classification and simplification of grammars Transducers and PDA
- Understand the designing techniques of Turing machines and between Decidable and Un-decidable Problems and group them into NP-Complete and NP-Hard..

COURSE OUTCOMES:

Students will be able to:

1. Understand the concept of finite automata for the design of finite state machine for some subset of languages.
2. Analyze the given problem and use the regular expression properties to form a regular expression.
3. Describe the relations between formal languages and grammar and simplify the grammars for the application of designing various parsing techniques.
4. Design of push down automata for some set of languages .
5. Design of Turing Machine for the language and analyze the differences between decidable and un-decidable Problems and group them into NP-Complete and NP-Hard.

UNIT – I: Finite Automata

Why Study Automata Theory? The Central Concepts of Automata Theory, Automation, Finite Automation, Transition Systems, Acceptance of a String by a Finite Automation, DFA, Design of DFAs, NFA, Design of NFA, Equivalence of DFA and NFA, Conversion of NFA into DFA, Finite Automata with E-Transition, Minimization of Finite Automata, Mealy and Moore Machines, Applications and Limitation of Finite Automata.

Learning Outcomes:

Students will be able to:

1. Distinguish DFA and NFA (L4)
2. Construct DFA for an input string (L6)
3. Perform minimization of Automata (L3)
4. Compare Moore and Mealy Machines (L4)

UNIT – II: Regular Expressions

Regular Expressions, Regular Sets, Identity Rules, Equivalence of two Regular Expressions, Manipulations of Regular Expressions, Finite Automata, and Regular Expressions, Inter Conversion, Equivalence between Finite Automata and Regular Expressions, Pumping Lemma, Closers Properties, Applications of Regular Expressions, Finite Automata and Regular Grammars, Regular Expressions and Regular Grammars.

Learning Outcomes:

Students will be able to:

1. Build regular expression for the given Finite Automata (L3)
2. Construct finite automata for the given regular expression (L6)
3. Apply closure properties on regular expressions (L3)

UNIT – III: Context Free Grammars

Formal Languages, Grammars, Classification of Grammars, Chomsky Hierarchy Theorem, Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars-Elimination of Useless Symbols, E-Productions and Unit Productions, Normal Forms for Context Free Grammars-Chomsky Normal Form and Greibach Normal Form, Pumping Lemma, Closure Properties, Applications of Context Free Grammars.

Learning Outcomes:

Students will be able to:

1. Define Context Free Grammar (L1)
2. Differentiate between Chomsky Normal Form and Greibach Normal form (L4)
3. Apply Pumping Lemma theorem on Context Free Grammar (L3)

UNIT – IV: Pushdown Automata

Pushdown Automata, Definition, Model, Graphical Notation, Instantaneous Description Language Acceptance of pushdown Automata, Design of Pushdown Automata,
Deterministic and Non – Deterministic Pushdown Automata, Equivalence of Pushdown Automata and Context Free Grammars Conversion, Two Stack Pushdown Automata, and Application of Pushdown Automata.

Learning Outcomes:

Students will be able to:

1. List the applications of Pushdown Automata (L1)
2. Build Pushdown Automata for context free grammar (L6)

UNIT – V: Turing Machine

Turing Machine, Definition, Model, Representation of Turing Machines-Instantaneous Descriptions, Transition Tables and Transition Diagrams, Language of a Turing Machine, Design of Turing Machines, Techniques for Turing Machine Construction, Types of Turing Machines, Universal Turing Machine, Decidable and Un-decidable Problems, Halting Problem of Turing Machines, Post’s Correspondence Problem, Classes of P and NP, NPHard and NP-Complete Problems.

Learning Outcomes:

Students will be able to:

1. List the applications of Turing machine (L1)
2. Design Turing machine for context free grammar (L6)

TEXT BOOKS:

1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwani and J. D.Ullman, 3rd Edition, Pearson, 2008.
2. Theory of Computer Science-Automata, Languages and Computation, K.L.P.Mishra and N.Chandrasekaran, 3rd Edition, PHI, 2007.

REFERENCE BOOKS:

1. Formal Language and Automata Theory, K.V.N.Sunitha and N.Kalyani, Pearson, 2015.
2. Introduction to Automata Theory, Formal Languages and Computation, ShyamalenduKandar, Pearson, 2013.
3. Theory of Computation, V.Kulkarni, Oxford University Press, 2013.
4. Theory of Automata, Languages and Computation, Rajendra Kumar, McGraw Hill, 2014.
5. https://en.wikipedia.org/wiki/Automata_theory
6. https://nptel.ac.in/courses/111103016/

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2205	Java Programming Lab	0	0	3	1.5

Course Objectives:

- To understand the structure and environment of Java.
- To implement the relationship between objects.
- To apply data hiding strategy in objects.
- To implement text processing and error handling.
- To organize data using different data structures.
- To create multi threaded graphical user interface applications.

Course Outcomes:

1. Create classes and objects for real world entities.
2. Implement polymorphic and abstract behaviour in objects.
3. Implement the parent-child relationships between objects with access protection.
4. Create exceptions for handling runtime errors during text processing.
5. Implement generic data structures for iterating distinct objects.

Exercise-1:

a. Write a Java program to create Class as Registration with properties as Full Name(String) , Gender(char), Age(int), Height(double), Phone Number(long), and is Married(Boolean) and print their values.
b. Write a Java program to implement Type Casting and Conversion.
c. Write a Java program to implement Wrapper Classes.

Exercise-2:

a. Write a Java program to take input as Regd.No and print the branch depending upon the department code in that Regd.No using else-if and switch statements. (EgRegNo: 19KD1A0505, 8th character is department Code, 5-CSE, 4-ECE, 3-MECH, 2-EEE etc.
b. Write a Java program to read input integers from Command Line Arguments and print first and second largest numbers.
c. Write a Java program to take input as Integer array and print even indexed even numbers and odd indexed odd numbers.

Exercise-3:

a. Write a Java program to take input as Decimal number and convert into Roman Number.
b. Write a Java program to check whether given number is Extension number. The extension number is the number which is present in the last digit(s) of its square.(Eg. N=25, 625 is Extension number since it contains 25).
c. Write a Java program to take input as Amount in rupees and print their denominations and total number notes.

Exercise-4:

a. Create a Class named Student with properties as Student Id, Student Name, gender, department, Age, Aggregate and methods as insertStudent() for inserting student details and displayStudent() for printing student details.
b. Create a class Student with same properties as above and create a constructor to insert student details and return the data using toString() method.

Exercise-5:

a. Design a Class named Transaction to transfer amount (double) in different ways using Account Number(int) , Phone Number(Long) and qr Code (String) as parameter into a method transferAmount() to achieve Method or Constructor OverLoading.

b. Design a super Class Account and sub Classes as LoanAccount, SavingsAccount and CurrentAccount and implement relationship between parent and child classes. (Implement Packages for the above classes)

Exercise-6:

a. Write a Java program to implement this and super keywords.
b. Write a Java program to implement Static property, method, block and package.
c. Write a Java program to implement final property, method and class.

Exercise-7:

a. Write a Java program to implement Data Abstraction using Abstract class and Interface.
b. Write a Java program to implement Multiple Inheritance through Interfaces.

Exercise-8:

a. Write a Java program to take input as String Sentence S and print largest and shortest word in S.
b. Write a Java program to take input as String S and remove the consecutive repeated characters from S. (Eg. S = Raaaamaaa then, Rama)
c. Write a Java program to take input as String Sentence S and print sentence again with all the words with the first letter as capital letter using StringBuffer.

Exercise-9:

a. Write a Java program to implement Map interface.
b. Write a Java program to implement Set Interface.
c. Write a Java program to implement List Interface.
d. Write a Java program to implement Comparator Interface.

Exercise-10:

a. Write a Java program to read data from Employee file and print Highest salary employee information. (Employee File Contains: ID, name, Dept, Salary).
b. Write a Java program to implements Serializable Interface to read and write Objects to/from the file.

Exercise-11:

a. Write a Java program to implement try, catch, finally blocks.
b. Write a Java program to create user defined Exception and implement throw and throws handlers.

Exercise-12:

a. Write a Java program to create Thread using Thread Class and Runnable Interface.
b. Write a Java program to implement multi threading and synchronization.
c. Write a Java program to implement Inter Thread Communication.

Exercise-13:

a. Create an Applet that changes the Font and background Color depending upon the user selection from the input.
b. Write a Java program to implement Event Delegation model through AWT Components.
c. Write a Java program to implement Layout Manager with AWT components.

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2206	Operating System Lab	0	0	3	1.5

Course objectives:

- To gain knowledge about the Operating Systems concepts such as process, main memory management, secondary memory management, CPU and disk scheduling etc
- Simulation of CPU scheduling algorithms, file allocation strategies, file organization techniques
- Simulation of Deadlock Avoidance and prevention. Algorithms Simulation of Page replacement algorithms and paging techniques
- Introduced the student to Unix/Linux kernel programming techniques.
- Review basic concepts coveredin the core operating Systems course prerequisite as they are realized in the Linux platform.

COURSE OUTCOMES:

1. implement various process scheduling programs
2. implement various memory management algorithms.
3. Identify various solutions for critical section problems and also implement different algorithms that are applied in virtual memory .
4. implement various file allocation algorithms
5. Describe and write shell scripts in order to perform basic shell programming.
6. Analyze various program editors and implement small program in linux environment.

OPERATING SYSTEMS

1. Simulate the following CPU scheduling algorithms

a) Round Robin b) SJF c) FCFS d) Priority

2. Multiprogramming-Memory management- Implementation of fork (), wait (), exec() and exit (), System calls

3. Simulate the following

a) Multiprogramming with a fixed number of tasks (MFT)

b) Multiprogramming with a variable number of tasks (MVT)

4. Simulate Bankers Algorithm for Dead Lock Avoidance

5. Simulate Bankers Algorithm for Dead Lock Prevention.

6. Simulate the following page replacement algorithms.

a) FIFO b) LRU c) LFU

7. Simulate the following File allocation strategies

a) Sequenced b) Indexed c) Linked

UNIX PROGRAMMING

1. a) Study of Unix/Linux general purpose utility command list
man,who,cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown.

b) Study of vi editor.

c) Study of Bash shell, Bourne shell and C shell in Unix/Linux operating system.

d) Study of Unix/Linux file system (tree structure).

e) Study of .bashrc, /etc/bashrc and Environment variables.

2.Write a C program that makes a copy of a file using standard I/O, and system calls

3. Write a C program to emulate the UNIX ls –l command.

4. Write a C program that illustrates how to execute two commands concurrently
with a command pipe.Ex: - ls –l | sort

5. Write a C program that illustrates two processes communicating using sharedmemory

6. Write a C program to simulate producer and consumer problem usingsemaphores

7. Write C program to create a thread using pthreads library and let it run its function.

8. Write a C program to illustrate concurrent execution of threads using pthreads library.

Text Books:

1. Operating System -Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley
2. The ultimate guide unix, Sumitan das.

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19CSE-PC2207	Database Managements Systems Lab	0	0	3	1.5

Course Objectives:

1.To provide a sound introduction to the discipline of database management as a subject in its own right, rather than as a compendium of techniques and product- specifictools.

2.To familiarize the participant with the nuances of database environments towards an information-oriented data-processing oriented frame work

3.To give a good formal foundation on the relational model of data

4.To present SQL and procedural interfaces to SQLcomprehensively

5.To give an introduction to systematic database design approaches covering conceptual design, logical design and an overview of physicaldesign

Course Outcomes:

1. Understand the procedure for creating the database.
2. Apply querying techniques to create Database tables by properly specifying Integrity constraints.
3. Apply SQL commands such as DDL, DML, DCL, TCL to access data from database objects
4. Understand the procedure to write Nested queries.
5. Develop PL/SQL stored procedures, stored functions ,cursors , packages and database connectivity

List of Experiments:

SQL

1. Queries to facilitate acquaintance of Built-In Functions, String Functions, Numeric Functions, Date Functions and Conversion Functions.
2. Queries using operators in SQL
3. Queries to Retrieve and Change Data: Select, Insert, Delete, and Update
4. Queries using Group By, Order By, and Having Clauses
5. Queries on Controlling Data: Commit, Rollback, and Save point
6. Queries to Build Report in SQL*PLUS
7. Queries for Creating, Dropping, and Altering Tables, Views, and Constraints
8. Queries on Joins and Correlated Sub-Queries
9. Queries on Working with Index, Sequence, Synonym, Controlling Access,and Locking Rows for Update, Creating Password and Security features

PL/SQL

1. Write a PL/SQL Code using Basic Variable, Anchored Declarations, and Usage of Assignment Operation
2. Write a PL/SQL Code Bind and Substitution Variables. Printing in PL/SQL
3. Write a PL/SQL block using SQL and Control Structures in PL/SQL
4. Write a PL/SQL Code using Cursors, Exceptions and Composite DataTypes
5. Write a PL/SQL Code using Procedures, Functions, and Packages FORMS
6. Write a PL/SQL Code Creation of forms for any Information System such as Student Information System, Employee Information System etc.18
7. Demonstration of data base connectivity

Text Books/Suggested Reading:

1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle”, PHI,2007.
3. Rick F Vander Lans, “Introduction to SQL”, Fourth Edition, Pearson Education,2007

II Year –II Semester

Subject Code	Subject Name	L	T	P	C
R19BSH-MC2201	Professional Ethics & Human Values	3	0	0	0

Course Objectives:

- Create an awareness on Engineering Ethics and Human Values.
- Instill Moral and Social Values and Loyalty
- Appreciate the rights of others.
- Create awareness on assessment of safety and risk

Course Outcomes:

- Identify and analyze an ethical issue in the subject matter under investigation or in a relevant field
- Identify the multiple ethical interests at stake in a real-world situation or practice
- Assess their own ethical values and the social context of problems
- Identify ethical concerns in research and intellectual contexts, including academic integrity.
- Develop knowledge about global ethical issues.

Unit I: Human Values:

Morals, Values and Ethics-Integrity-Work Ethic-Service learning – Civic Virtue – Respect for others –Living Peacefully –Caring –Sharing –Honesty -Courage-Cooperation– Commitment – Empathy –Self Confidence Character –Spirituality.

Learning Outcomes:

1. Learn about morals, values & work ethics.
2. Learn to respect others and develop civic virtue.
3. develop commitment
4. learn how to live peacefully

Unit II: Engineering Ethics:

Senses of 'Engineering Ethics-Variety of moral issued –Types of inquiry –Moral dilemmas –Moral autonomy –Kohlberg's theory-Gilligan's theory-Consensus and controversy –Models of professional roles-Theories about right action-Self interest -Customs and religion –Uses of Ethical theories –Valuing time –Co operation –Commitment.

Learning Outcomes:

1. Learn about the ethical responsibilities of the engineers.
2. Create awareness about the customs and religions.
3. Learn time management
4. Learn about the different professional roles.

Unit III: Engineering as Social Experimentation

Engineering As Social Experimentation –Framing the problem –Determining the facts –Codes of Ethics –Clarifying Concepts –Application issues –Common Ground -General Principles –Utilitarian thinking respect for persons

Learning Outcomes:

1. demonstrate knowledge to become a social experimenter.
2. provide depth knowledge on framing of the problem and determining the facts.
3. provide depth knowledge on codes of ethics.
4. develop utilitarian thinking

UNIT IV: Engineers Responsibility for Safety and Risk:

Safety and risk –Assessment of safety and risk –Risk benefit analysis and reducing risk-Safety and the Engineer-Designing for the safety-Intellectual Property rights(IPR).

Learning Outcomes:

1. Create awareness about safety, risk & risk benefit analysis.
2. Engineer's design practices for providing safety.

3. Provide knowledge on Intellectual Property Rights.

UINIT V: Global Issues

Globalization –Cross culture issues-Environmental Ethics –Computer Ethics –Computers as the instrument of Unethical behavior –Computers as the object of Unethical acts – Autonomous Computers-Computer codes of Ethics –Weapons Development -Ethics and Research –Analyzing Ethical Problems in research.

Learning Outcomes:

1. Develop knowledge about global issues.
2. Create awareness on computer and environmental ethics
3. Analyze ethical problems in research.
4. Give a picture on weapons development.

Text Books:

1. “Engineering Ethics includes Human Values” byM.Govindarajan, S.Natarajananad, V.S.SenthilKumar-PHI Learning Pvt. Ltd-2009
2. “Engineering Ethics” by Harris, Pritchard and Rabins, CENGAGE Learning, India Edition, 2009.
3. “Ethics in Engineering” by Mike W. Martin and Roland Schinzinger –Tata McGraw-Hill– 2003.
4. “Professional Ethics and Morals” by Prof.A.R.Aryasri, DharanikotaSuyodhana-Maruthi Publications.
5. “Professional Ethics and Human Values” by A.Alavudeen, R.Kalil Rahman and M.Jayakumaran-LaxmiPublications.
6. “Professional Ethics and Human Values” by Prof.D.R.Kiran-
7. “Indian Culture, Values and Professional Ethics” by PSR Murthy-BS Publication